

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25

2019

№ 3

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ
И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И
НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ
СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС

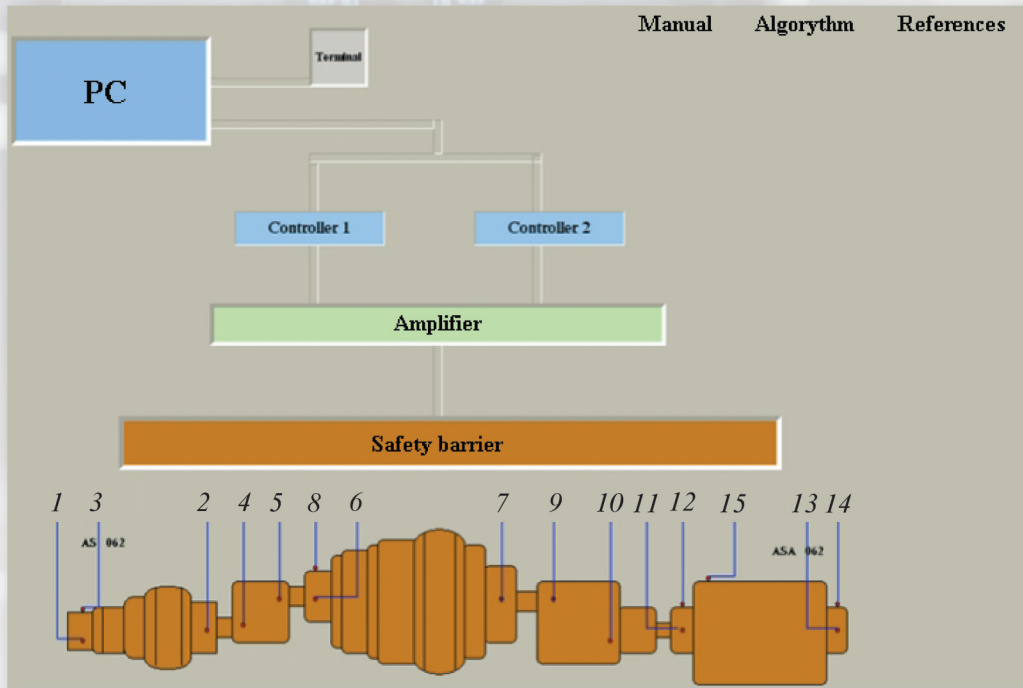


Fig. 2. Information display of Intelligent Monitoring System for Compressor Devices

Рисунок к статье Е. А. Ююгина

«КОРРЕКТНОЕ И БЫСТРОЕ ИСПОЛНЕНИЕ ОТДЕЛЬНЫХ
 ИНСТРУКЦИЙ АРХИТЕКТУРЫ INTEL® 64 В ВИРТУАЛЬНОМ ОКРУЖЕНИИ»

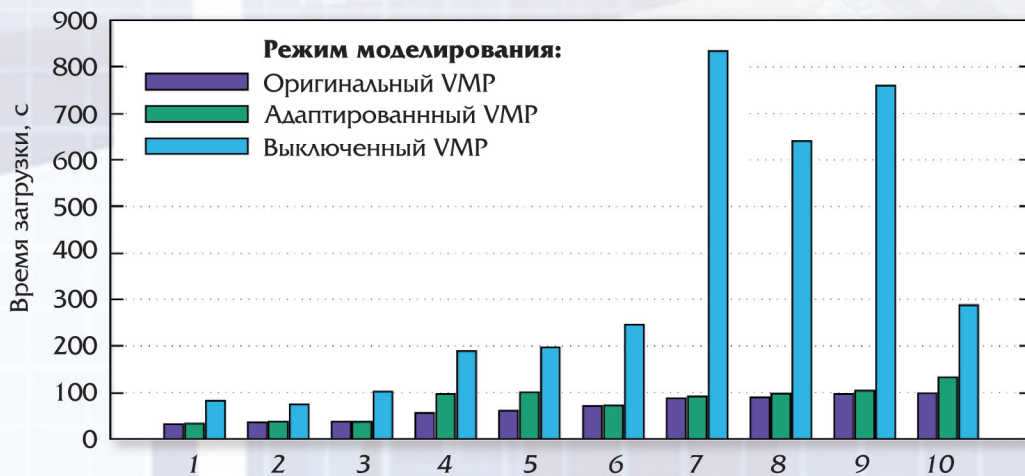


Рис. 4. Время загрузки ключевых сценариев

с использованием различных режимов моделирования:

- 1 – Free DOS; 2 – Busybox (3.2.4); 3 – FreeBSD 10.3; 4 – Fedora 23 (4.2.3);
- 5 – OS Linux (3.10.7); 6 – Fedora 18 (3.6.10); 7 – Windows 8.1;
- 8 – Windows server 2016; 9 – Windows 10; 10 – Fedora 22 KVM (4.2.8)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25
2019
№ 3

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Петров И. С.** Системы обнаружения скомпрометированных коммутаторов в программно-конфигурируемых сетях 131
- Ефанов Д. В.** Условия построения полностью самопроверяемых дискретных устройств на основе метода логического дополнения до равновесного кода "2 из 4" 143

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Pashayev F. N., Pashayeva S. E., Najafov N. T., Suleymanli B. A.** Operating Algorithms for Folders and Files 152
- Олюгин Е. А.** Корректное и быстрое исполнение отдельных инструкций архитектуры Intel® 64 в виртуальном окружении 157

БАЗЫ ДАННЫХ

- Гвоздев В. Е., Блинова Д. В., Кириллов К. В.** Информационная поддержка принятия решений по устранению дефектов, выявленных при эксплуатации систем обработки данных 165

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Абдуллаев Н. Т., Исмаилова К. Ш.** Выбор шага итерации в процессе обучения нейронной сети при использовании релаксационных алгоритмов 174

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В БИОМЕДИЦИНСКИХ СИСТЕМАХ

- Артюшина Е. В., Ершов Д. Е., Прохода Ю. А.** Роль интернет-технологий в коммуникационной деятельности медицинской организации 179

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Гвоздкова Ю. Д., Гвоздкова И. А., Курочкин А. В., Черняев А. В.** Информационная система оценки экологической безопасности авиационных материалов и технологий методом анализа иерархий 185

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.
ВАСИЛЬЕВ В. и., д. т. 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.
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.

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. 25
2019
No. 3

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

COMPUTING SYSTEMS AND NETWORKS

- Petrov I. S.** Systems for Compromised Switch Detection in Software-Defined Networking 131
- Efanov D. V.** The Conditions of the Formation of the Totally Self-Checking Discrete Devices Based on the Boolean Complement Method up to 2-out-of-4 Constant-Weight Code 143

SOFTWARE ENGINEERING

- Pashayev F. H., Pashayeva S. E., Najafov H. T., Suleymanli B. A.** Operating Algorithms for Folders and Files 152
- Yulyugin E. A.** Fast and Accurate Direct Execution of Particular Intel® 64 Instructions in Virtual Environment 157

DATABASE

- Gvozdev V. E., Blinova D. V., Kirillov K. V.** Information Decision-Making Support on Elimination of the Defects Detected at Exploitation of Data Processing Systems 165

NEUROTECHNOLOGIES

- Abdullayev N. T., Ismayilova K. Sh.** The Choice of the Iteration Step in the Process of Learning a Neural Network Using Relaxation Algorithms 174

INFORMATION SYSTEMS IN BIOMEDICAL SYSTEMS

- Artyushina E. V., Erschov D. E., Prokhoda Yu. A.** The Role of Internet Technologies in the Communication Activity of Medical Organization 179

APPLIED INFORMATION TECHNOLOGIES

- Gvozdkova J. D., Gvozdkova I. A., Kurochkin A. V., Chernyaev A. V.** Information System of Environmental Safety Assessment of Aviation Materials and Technologies by the Analytic Hierarchy Process 185

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.
Vasiliev V. I., Dr. Sci. (Tech.), 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.
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.

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"

И. С. Петров, аспирант, e-mail: ipetrov@cs.msu.ru,
Московский государственный университет им. М. В. Ломоносова

Системы обнаружения скомпрометированных коммутаторов в программно-конфигурируемых сетях

Скомпрометированные коммутаторы являются серьезной угрозой безопасности сети, так как они могут использоваться злоумышленником для проведения широкого спектра атак на сеть. В статье представлен сравнительный анализ существующих систем обнаружения скомпрометированных коммутаторов в программно-конфигурируемых сетях, отмечены достоинства и недостатки каждой из них.

Ключевые слова: программно-конфигурируемые сети, системы обнаружения вторжений, скомпрометированные коммутаторы

Введение

Программно-конфигурируемые сети¹ (ПКС), по мнению ведущих производителей сетевого оборудования, являются одним из самых перспективных направлений сетевой индустрии на данный момент [1].

В ПКС уровни управления и передачи данных разделяются за счет переноса функций управления на отдельное устройство — контроллер. На сетевых устройствах — коммутаторах — используются специальные таблицы потоков, в которых контроллер задает правила для маршрутизации и/или модификации пакетов.

Одним из важных элементов концепции ПКС является протокол OpenFlow [2] для программирования и управления сетевыми устройствами. Протокол OpenFlow предполагает использование на коммутаторах специальных таблиц маршрутизации, в которые контроллер загружает правила для маршрутизации и/или модификации пакетов, проходящих через коммутатор. Такие таблицы называются таблицами потоков.

С появлением нового вида компьютерных сетей появляются и новые виды уязвимостей, которые могут привести не только к материальным потерям, но и к утрате репутации и имиджа компаний, использующих подобные

сети. Следовательно, во избежание негативных последствий компьютерных атак необходимо проводить анализ защищенности протоколов ПКС и разрабатывать механизмы защиты таких сетей от атак различных типов.

Одной из возможных угроз безопасности сети является наличие в ПКС скомпрометированного коммутатора. Под скомпрометированным коммутатором понимается некоторый коммутатор ПКС, управляемый контроллером, который в действительности находится под контролем злоумышленника. Возможность компрометации коммутатора обусловлена тем, что сетевое оборудование может иметь уязвимости [3, 4], которые могут быть использованы злоумышленником для получения контроля над коммутатором.

Несмотря на то что все функции управления сетью вынесены на контроллер, компрометация коммутатора является серьезной угрозой безопасности всей сети, так как атакующий может использовать подконтрольные ему коммутаторы для проведения широкого спектра атак как на контур данных, так и на контур управления [4, 5].

Сложность задачи обнаружения скомпрометированного коммутатора заключается в том, что подобный коммутатор невозможно обнаружить средствами аутентификации устройства. Это обусловлено тем, что, получая контроль над коммутатором, злоумышленник получает доступ к криптографическим ключам, находящимся

¹ Англ. аналог — Software Defined Network.

в памяти коммутатора. Далее, используя эти ключи, злоумышленник может провести процедуру аутентификации скомпрометированного коммутатора, а любой коммутатор, прошедший процедуру аутентификации, будет считаться легитимным с точки зрения контроллера.

На данный момент существует множество систем обнаружения скомпрометированных коммутаторов [6—15]. Такие системы используют различные механизмы для проверки корректности работы контура данных. К таким механизмам относятся анализ сетевой статистики и тестирование сети с помощью специально созданных пакетов. Также системы обнаружения используют тот факт, что у контроллера ПКС имеется полная информация о топологии сети, свойствах потоков, которые проходят через эту сеть, и правилах обработки сетевого трафика, установленных на коммутаторы.

В статье представлен сравнительный анализ существующих систем обнаружения скомпрометированных коммутаторов в ПКС, отмечены достоинства и недостатки каждой из них. Необходимо отметить, что в обзоре рассматриваются только системы обнаружения коммутаторов, которые уже были скомпрометированы и использованы для различных атак на контур данных. Системы обнаружения процесса компрометации не рассмотрены, так как эта область информационной безопасности уже является хорошо исследованной [16].

1. Критерии сравнения

Ниже сформулированы требования к системе обнаружения скомпрометированных коммутаторов в ПКС, основанные на функциональности OpenFlow-коммутатора, которые мы будем использовать в качестве критериев сравнения систем в этом обзоре.

К1. Обнаружение двух и более скомпрометированных коммутаторов

Система обнаружения должна иметь возможность распознать ситуацию, когда злоумышленник захватил несколько коммутаторов в сети и организовал их согласованную работу так, чтобы избежать обнаружения.

К2. Верификация данных, поступающих от каждого коммутатора

Система обнаружения должна распознавать скоординированные действия скомпрометированных коммутаторов, при которых контроллер получает статистику, искаженная таким образом, чтобы скрыть факт наличия атаки, либо заставить систему обнаружения подозревать легитимный коммутатор.

К3. Разграничение вредоносного и легитимного сброса пакетов

Важным критерием сравнения является возможность системы отличать вредоносный сброс пакета на коммутаторе вследствие атаки от легитимного сброса пакета в силу либо наличия в памяти коммутатора соответствующего правила обработки пакетов, либо из-за перегрузки легитимного коммутатора. Если система не способна различать эти случаи сброса пакетов, то обнаружение скомпрометированных коммутаторов будет сопровождаться большим числом ошибок второго рода, когда сброс пакетов из-за перегрузки в сети будет восприниматься как атака.

К4. Отсутствие требования модификаций контура данных

Применение системы обнаружения скомпрометированных коммутаторов не должно требовать изменений в существующих протоколах и логике работы коммутаторов в контуре данных. Изменения в логике работы коммутаторов могут быть дорогостоящими и в некоторых ситуациях являются неприемлемым решением. Поэтому для того чтобы система могла быть применима в реальной сети, необходимо, чтобы она не требовала внесения изменений в существующие протоколы и логику работы коммутаторов.

К5. Обнаружение атаки вне зависимости от ее длительности

Этот критерий описывает возможность системы обнаруживать атаки, время которых незначительно по сравнению со временем жизни потока данных в сети. Например, к таким атакам могут относиться кратковременный сброс пакетов, кратковременная DoS-атака на некоторого пользователя или атака на определенные пакеты некоторого потока в сети.

К6. Независимость от используемых в сети алгоритмов и политик маршрутизации

Этот критерий предполагает возможность системы работать при использовании в сети разнообразных алгоритмов и политик маршрутизации. Поскольку архитектура ПКС дает большую свободу в управлении сетевым трафиком, то заранее предвидеть сложность алгоритмов и политик маршрутизации невозможно. Также система не должна зависеть от применяемых в сети механизмов маршрутизации и оптимизации потоков, таких как агрегация потоков, балансировка нагрузки, перенаправление трафика в случае изменений в топологии из-за ошибок в сети [17].

К7. Отсутствие влияния на атаки

Процедура обнаружения атаки может влиять на состояние сети, например, устанавливать новые правила маршрутизации, генериро-

вать новые служебные пакеты. Поэтому могут возникнуть ситуации, когда процедура обнаружения может повлиять на саму атаку, проводимую в сети. Из-за такого влияния атака может прекратиться, и, таким образом, она не будет обнаружена системой.

2. Системы обнаружения скомпрометированных коммутаторов

2.1. ATPG

Система ATPG² [6] — это система, предназначенная для обнаружения ошибок в работе контура данных. Под ошибками понимается некорректное исполнение коммутатором установленных на него правил маршрутизации. Система ATPG обнаруживает ошибки с помощью независимого и полного тестирования всех правил маршрутизации, установленных в сети.

Тестирование правил маршрутизации проводится с помощью так называемых тестовых пакетов — специальных пакетов, посылаемых в сеть системой, для которых заранее известен маршрут. Система устанавливает в сеть дополнительные правила маршрутизации, которые должны отправлять тестовые пакеты обратно на контроллер в конце их маршрута. Таким образом, система ATPG может проверить, что пакеты действительно прошли предполагаемым маршрутом и не были изменены в процессе.

Так как большое число тестовых пакетов может снизить производительность сети, то система ATPG формирует минимальное множество пакетов, которое будет обработано всеми правилами маршрутизации в сети. Минимальность формируемого множества доказана авторами в работе [6].

Для этого система формирует таблицу достижимости между всеми портами в сети. Эта таблица описывает всевозможные маршруты, которые пакеты могут пройти в сети. Также для каждого маршрута известен набор заголовков пакетов, которые могут пройти по этому маршруту.

Для каждого маршрута система случайно выбирает один пакет из множества, соответствующего этому маршруту. В результате формируется множество пакетов \mathcal{P} такое, что для каждого правила маршрутизации найдется пакет $p \in \mathcal{P}$, который обрабатывается этим правилом. Если разные пакеты из этого множества обрабатываются одним и тем же правилом

маршрутизации, то алгоритм выбирает минимальное подмножество пакетов, которое также обрабатывается всеми правилами маршрутизации. Иными словами, решается задача покрытия множества подмножествами. Так как известно, что эта задача NP-сложна [18], то для ее решения используется аппроксимационный алгоритм со сложностью $O(n^2)$.

Одним из недостатков этой системы является тот факт, что обнаружение ошибок работы сети с помощью тестовых пакетов может приводить к большим ошибкам второго рода, которые могут возникать вследствие сброса пакетов, происходящего из-за перегрузок в сети. Поскольку в системе не предусмотрено никаких механизмов анализа перегрузок, возникающих в сети, система не сможет отличить вредоносный сброс пакетов атакующим от легитимного сброса из-за перегрузки.

Подход на основе использования тестовых пакетов также может не обнаружить атаки, время жизни которых меньше интервала между тестированием сети. Также система может не обнаружить сложные атаки, которые осуществляются с помощью нескольких скомпрометированных коммутаторов.

Таким образом, система ATPG не удовлетворяет критериям **K3** и **K5**.

2.2. FADE

Система FADE³ [7] анализирует сетевую статистику, предоставляемую счетчиками правил маршрутизации, и проверяет ее согласованность. Под согласованностью сетевой статистики понимается следующее: разность значений счетчиков правил маршрутизации, обрабатывающих один и тот же поток, должна быть в пределах заранее определенного порогового значения.

Для анализа сетевой статистики система FADE выбирает минимальный набор потоков, которые описывают поведение счетчиков всех правил маршрутизации, установленных в сети. Для нахождения таких потоков система FADE строит граф зависимостей правил маршрутизации в соответствии с топологией и установленными в сети правилами маршрутизации. Граф зависимостей правил маршрутизации — это ориентированный граф, вершинами которого являются правила маршрутизации, а ребрами — возможные переходы пакетов между правилами. Таким образом, каждый поток в сети представлен некоторым путем в графе зависимостей правил маршрутизации.

² Automatic Test Packet Generation.

³ Forwarding Anomaly Detect Environment.

Авторы статьи [7] предполагают наличие в сети агрегирующих правил маршрутизации — правил, которые обрабатывают сразу несколько потоков. Такие правила описываются вершинами графа с несколькими входящими ребрами и одним исходящим.

Из-за наличия таких вершин следует, что потоки, описываемые путями в графе зависимостей правил, объединяются в деревья, и этот граф представляет собой лес, где потоки направлены от листьев к корням деревьев. Везде далее, если не оговорено противное, под термином граф мы будем понимать граф зависимостей правил маршрутизации.

Система FADE выбирает минимальный набор путей, которые покрывают все вершины в графе. Такие пути соответствуют потокам в сети, обработка которых затрагивает все правила маршрутизации. Для каждого таким образом выбранного потока система FADE генерирует набор дополнительных правил маршрутизации и устанавливает его на коммутаторы, обрабатывающие этот поток. Дополнительные правила маршрутизации, установленные системой FADE, используются для сбора сетевой статистики с коммутаторов.

Для того чтобы устанавливаемые системой FADE правила маршрутизации обрабатывали только пакеты соответствующих им потоков, система FADE помечает пакеты выделенными VLAN-метками. На каждом таком правиле маршрутизации установлено временное ограничение — *hard-timeout*, после истечения которого правило удаляется, и на контроллер отправляется статистика о числе пакетов, обработанных этим правилом.

После сбора статистики система FADE сравнивает значения счетчиков правил маршрутизации и делает вывод о наличии аномалии маршрутизации в сети, если эти значения отличаются более чем на заранее определенную пороговую величину.

Одним из недостатков системы FADE является тот факт, что она не учитывает возможность наличия в сети легитимных сбросов пакетов из-за перегрузок. Также система применима только в том случае, когда на контроллере реализована простая логика маршрутизации, не использующая агрегацию потоков, групповую адресацию, балансировку нагрузки и т. д. Учитывается возможность наличия агрегирующих правил, обрабатывающих сразу несколько потоков, но затем система эти потоки не различает. Не рассматривается групповая маршрутизация, изменение заголовков пакетов и деагрегация потоков, используемая для балансировки нагрузки на коммутаторы.

В статье [7] предполагается, что одна атака нацелена на одно определенное правило маршрутизации, т. е. не учитывается возможность наличия в сети нескольких скомпрометированных коммутаторов.

Также существует возможность, что во время анализа сетевой статистики система FADE своими действиями может повлиять на атаку и, таким образом, ее не обнаружить. Например, если дополнительные правила маршрутизации устанавливаются на скомпрометированный коммутатор, то эти правила могут начать обрабатывать пакеты вместо вредоносных правил, и атака не будет обнаружена. При этом после анализа сетевой статистики, когда дополнительные правила будут удалены с коммутаторов, атака может продолжиться.

Кроме того, в статье не описано, каким образом система реагирует на изменения в сети.

Таким образом, система FADE не удовлетворяет критериям *K1*, *K3*, *K6* и *K7*.

2.3. FlowMon

Система FlowMon [8] предназначена для обнаружения скомпрометированных коммутаторов с помощью анализа сетевой статистики, получаемой с портов коммутаторов, и сообщений о новых потоках в сети. Эта система позволяет обнаружить два типа вредоносного поведения коммутатора: нелегитимный сброс пакетов и нарушение политики маршрутизации. Под сбросом пакетов подразумевается, что на скомпрометированный коммутатор устанавливаются вредоносное правило, которое сбрасывает пакеты всех или части потоков, обрабатываемых скомпрометированным коммутатором. Под нарушением политики маршрутизации понимается отправка пакетов на порты, не предусмотренные правилами, которые установил контроллер.

Для обнаружения вредоносного сброса пакетов система FlowMon анализирует значения различных счетчиков на портах коммутаторов, которые соответствуют числу переданных, полученных и сброшенных пакетов. Система выбирает некоторый коммутатор S_k для проверки, запрашивает статистику с его портов.

Коммутатор S_k считается скомпрометированным, если коэффициент сброса пакетов превышает заранее определенное пороговое значение θ , т. е. выполняется следующая формула:

$$\left| \frac{\sum_{\forall i \in P_k} T_k^i - \sum_{\forall i \in P_k} R_k^i}{\sum_{\forall i \in P_k} T_k^i + \sum_{\forall i \in P_k} R_k^i} \right| > \theta,$$

где T_k^i — число пакетов, переданных с порта i коммутатора S_k , R_k^i — число пакетов, полученных на порту i коммутатора S_k , P_k — множество портов коммутатора S_k .

Также система FlowMon проверяет наличие вредоносного сброса трафика на физических линиях. Система проверяет соответствие числа пакетов, отправленных и полученных на портах, соединенных одной и той же физической линией. Если некоторые пакеты были сброшены на физической линии, то статистика на портах будет отличаться и нижеприведенное отношение не будет выполняться:

$$|T_{ij} + T_{ji} - R_{ij} - R_{ji}| > \alpha |T_{ij} + T_{ji}|,$$

где T_{ij} — число пакетов, переданных с коммутатора S_i на коммутатор S_j , R_{ij} — число пакетов, полученных на коммутаторе S_i с коммутатора S_j , а α — это заранее определенное пороговое значение.

Для обнаружения некорректной коммутации пакетов система FlowMon проверяет сообщения обо всех новых потоках, появившихся в сети. Проверка проводится путем анализа таблицы маршрутизации предыдущего коммутатора, т. е. коммутатора, с которого был получен пакет нового потока. Контроллер проверяет, на какой порт должен был быть отправлен этот пакет. Если такой порт отличается от порта, на котором был получен пакет, то делается вывод, что коммутатор некорректно коммутирует пакеты.

Серьезным недостатком системы FlowMon является то, что при ее построении используется слишком простая модель атакующего, которая не позволяет проконтролировать подделку статистики скомпрометированным коммутатором и которая не учитывает возможность наличия в сети нескольких скомпрометированных коммутаторов.

Из-за того, что система поочередно проверяет различные коммутаторы, она не способна обнаружить кратковременные атаки, которые выполняются на коммутаторе, не участвующем в проверке в данный момент.

Не учитываются также различные способы сокрытия атак. Например, атакующий может целенаправленно коммутировать пакеты на коммутаторы, на которых уже присутствуют вредоносные правила обработки подобных пакетов. Таким образом, статистика об этом пакете никогда не попадет на контроллер. Подобная ситуация может появиться, когда злоумышленник изменяет заголовки новых потоков на заголовки существующих.

Также система не учитывает возможность наличия на коммутаторе легитимных правил, которые сбрасывают трафик. Такими правилами, например, могут быть правила, установленные межсетевым экраном.

Таким образом, система FlowMon не удовлетворяет критериям **K1**, **K2**, **K3** и **K5**.

2.4. FDWD

Система FDWD⁴ [9] — это система для обнаружения скомпрометированных коммутаторов в ПКС, которая использует тестовые пакеты для проверки того факта, что коммутатор не выполняет установленные на него правила маршрутизации. Применяются два алгоритма обнаружения: Forwarding Detection и Weighting Detection.

Алгоритм Forwarding Detection использует тестовые пакеты для проверки корректности выполнения правил маршрутизации на коммутаторах. Алгоритм работает следующим образом. Случайно выбирается коммутатор S , и на нем случайно выбирается правило маршрутизации f с полем match M . Далее алгоритм генерирует новые правила маршрутизации f' с полем match $M' = M \cup M_i$, где M_i — поле match, не пересекающееся с полем M , такое что в сети нет пакетов с заголовками, удовлетворяющими этому полю. Эти правила маршрутизации устанавливаются на коммутаторы, непосредственно соединенные с коммутатором S .

После этого алгоритм генерирует тестовый пакет p , заголовок которого удовлетворяет полю M' , и отправляет его на коммутатор S . Теперь контроллер может проверить, на какой коммутатор будет отправлен пакет p . Если правило f должно отправлять пакеты на коммутатор S' , то при корректном поведении коммутатора S пакет p должен быть обработан правилом f' на коммутаторе S' . В противном случае коммутатор S будет считаться скомпрометированным. Также некорректным поведением будет считаться изменение пакета p .

Алгоритм Weighting Detection проверяет специальные групповые правила g протокола OpenFlow, которые отправляют пакеты на разные порты в заранее заданном соотношении. Проверка заключается в том, что контроллер отправляет на тестируемый коммутатор несколько специальных пакетов для того, чтобы затем проверить, что пакеты были отправлены на порты в правильном соотношении, заданном правилом g . Проверка происходит так же, как и

⁴ Forwarding Detection and Weighting Detection.

в алгоритме Forwarding Detection — с помощью специальных правил, заранее установленных на соседние с тестируемым коммутаторы.

Недостатком описанной выше системы является тот факт, что она не учитывает возможность наличия в сети нескольких скомпрометированных коммутаторов, которые могут кооперироваться для того, чтобы избежать обнаружения. Например, если несколько соседних коммутаторов скомпрометированы, то они могут передавать контроллеру неверные сведения о прохождении тестовых пакетов в сети, таким образом скрывая наличие атаки.

Так как система проверяет одно правило за раз, то она может не обнаружить кратковременные атаки.

Из-за того, что система устанавливает дополнительные правила, она может повлиять на атаку, проводимую в сети. Например, система может установить правило, имеющее больший приоритет, чем у вредоносного правила.

Также система не учитывает наличие в сети сбросов пакетов из-за перегрузок.

Таким образом, система FDWD не удовлетворяет критериям **K1**, **K3**, **K5** и **K7**.

2.5. MLPC

Система MLPC⁵ [10] также использует тестовые пакеты и анализ сетевой статистики для обнаружения скомпрометированных коммутаторов. В статье [10] предполагается, что скомпрометированные коммутаторы могут выборочно сбрасывать, генерировать, изменять или некорректно коммутировать пакеты. Также предполагается, что в сети могут быть несколько скомпрометированных коммутаторов, которые могут кооперироваться между собой для того, чтобы избежать обнаружения.

Для проверки корректной работы сети система вычисляет минимальный набор тестовых пакетов, обработка которых охватывает все правила на коммутаторах в сети. Для нахождения такого набора пакетов система использует информацию об установленных в сети правилах маршрутизации и строит граф зависимостей правил маршрутизации.

Далее система использует топологическую сортировку [19] и модифицированную версию алгоритма Хопкрофта — Карпа [20] для нахождения минимального покрытия графа путями, причем учитываются только пути, соответствующие реальным маршрутам пакетов в сети. Назовем такой путь реализуемым. Для

каждого реализуемого пути генерируется тестовый пакет, который должен пройти по всем правилам маршрутизации всех коммутаторов, соответствующим этому пути. В конце каждого пути устанавливается специальное правило маршрутизации, которое должно вернуть тестовые пакеты на контроллер для анализа. Для того чтобы процедура анализа сети не изменяла логику обработки пакетов, тестовые пакеты помечаются уникальной меткой (например, VLAN-меткой, не используемой в сети).

Если тестовый пакет был сброшен или модифицирован, контроллер уменьшает степень доверия к конкретному пути, соответствующему этому пакету. Когда степень доверия уменьшается ниже допустимого порога, путь помечается как подозрительный. Для определения местонахождения скомпрометированного коммутатора контроллер разбивает подозрительный путь на две части и повторяет процедуру тестирования рекурсивно, пока длина подозрительного пути не будет равна 1.

Для того чтобы обнаружить скомпрометированные коммутаторы S , которые отправляют некорректную информацию, контроллер использует коммутаторы, соединенные с S физическими линиями. Система проверяет правило сохранения потока на входе и выходе коммутатора: число пакетов, отправленных на коммутатор, должно быть равно числу пакетов, полученных от этого коммутатора.

Для обнаружения несоответствий в сетевой статистике контроллер устанавливает на соседние коммутаторы дополнительные правила, которые считают число пакетов, отправленных на тестируемый коммутатор, и число пакетов, полученных от тестируемого коммутатора.

Далее проверяется правило сохранения потока. Если найдено несоответствие, то уровень доверия к коммутатору снижается. Если уровень доверия снизится ниже допустимого порога, то коммутатор считается скомпрометированным.

Важно отметить, что подобная проверка статистики может быть неспособна обнаружить точное местоположение скомпрометированного коммутатора. Например, скомпрометированный коммутатор может предоставлять контроллеру специально сформированную сетевую статистику так, чтобы снижать уровень доверия системы к некоторому легитимному коммутатору. Также в ситуации, когда легитимный коммутатор окружен скомпрометированными, его уровень доверия может снижаться быстрее, чем у граничащих с ним коммутаторов.

Система не учитывает случай, когда скомпрометированные коммутаторы могут быть со-

⁵ Minimal Legal Path Cover.

седними, т. е. соединены физической линией. В таком случае один из них может скрывать от контроллера информацию о потоке, выходящем с другого коммутатора. Также система не отличает легитимный сброс пакетов от вредоносного и не рассматривает агрегацию потоков, балансировку нагрузки и другие сложные механизмы маршрутизации.

Из-за того, что система проводит выборочные проверки правил маршрутизации с помощью тестовых пакетов и анализа сетевой статистики, она может пропустить кратковременные атаки.

Также из-за установки правил система влияет на состояние сети и, как следствие, влияет на проводимые в сети атаки.

Таким образом, система MLPC не удовлетворяет критериям **K1**, **K2**, **K3**, **K5**, **K6** и **K7**.

2.6. PDMD

Система PDMD⁶ [11] — это система обнаружения скомпрометированных коммутаторов путем проверки маршрутов, пройденных пакетами в сети.

Для выявления реальных маршрутов, пройденных пакетами, система устанавливает в сети правила маршрутизации, которые дублируют часть пакетов на контроллер. Выбор пакетов для отправки на контроллер проводится с помощью алгоритма Trajectory Sampling [21]. Этот алгоритм выбирает пакеты на основе решающей функции от метки пакета. Под меткой понимается набор полей заголовка пакета, которые уникально идентифицируют пакет в сети. Авторы статьи предполагают, что в сети не установлены правила маршрутизации, которые изменяют заголовки пакетов. Иными словами, значение решающей функции от одного и того же пакета одинаково на любом участке сети. Следовательно, выбранный для анализа пакет будет отправляться на контроллер на всех коммутаторах, на которых он был обработан.

Таким образом, система может получать из сети информацию о реальных маршрутах, пройденных пакетами. Далее, на основе информации об установленных в сети правилах маршрутизации система вычисляет теоретический маршрут, который каждый такой пакет должен был пройти в сети. Теоретический маршрут вычисляют по алгоритму достижения из модели Header Space Analysis [22].

Далее система может сравнить реальный маршрут, пройденный выбранным пакетом,

с теоретическим маршрутом. Если маршруты не совпадают, то делается вывод о наличии в сети скомпрометированного коммутатора.

Недостатком системы является тот факт, что при анализе маршрутов не учитывается, что в сети может находиться несколько скомпрометированных коммутаторов, и они могут отправлять на контроллер некорректную информацию о маршрутах пакетов, влияя на процедуру обнаружения. Например, такие коммутаторы могут не отправлять пакеты на контроллер, тем самым скрывая факт атаки. Также они могут изменять заголовки тестовых пакетов для того, чтобы в дальнейшем эти пакеты не были отправлены на контроллер другими легитимными коммутаторами.

В статье [11] предполагается, что в сети не может быть перегрузок, и все сброшенные пакеты были сброшены из-за атаки, а не из-за перегрузки. Также предполагается, что задержка обработки пакетов на коммутаторах и контроллере незначительна.

Из-за того, что алгоритм Trajectory Sampling [21] может не выбрать для анализа пакеты, относящиеся к некоторой проводимой в сети атаке, то эта атака не будет обнаружена. Такое может произойти, например, в случае наличия в сети кратковременной атаки.

Также необходимо отметить, что система рассматривает только простую логику маршрутизации — отдельные правила для различных потоков.

Из сказанного следует, что система PDMD не удовлетворяет критериям **K1**, **K2**, **K3**, **K5** и **K6**.

2.7. SPHINX

Система SPHINX [12] — это система обнаружения вторжений для ПКС. Она обнаруживает различные атаки, направленные на контур управления сетью, и проверяет соблюдение политики безопасности. Также эта система позволяет обнаруживать скомпрометированные коммутаторы, которые проводят различные атаки на контур данных.

Для обнаружения скомпрометированных коммутаторов система SPHINX анализирует устанавливаемые в сеть правила и строит модель сети — потоковый граф. Этот граф описывает маршруты различных потоков в сети. Потоковый граф используется для анализа сетевой статистики и обнаружения аномалий, таких как нелегитимный сброс пакетов.

Анализ сетевой статистики происходит следующим образом: для каждого потока в сети система запрашивает значения счетчиков правил, входящих в маршрут этого потока. Если

⁶ Packet Drops and Misroutings Detector.

счетчики правил одного и того же потока отличаются больше, чем на заранее определенное пороговое значение, то система SPHINX сообщает о наличии в сети скомпрометированного коммутатора.

Недостатком системы SPHINX является то, что она не учитывает возможность использования в сети сложных механизмов маршрутизации, таких как групповая маршрутизация, агрегация потоков и балансировка нагрузки. Все потоки представляются маршрутами в сети, для которых установлены уникальные правила маршрутизации. Также не учитывается информация о пакетах, сброшенных из-за перегрузок в сети.

Таким образом, система SPHINX не удовлетворяет критериям **К3** и **К6**.

2.8. RDDF

Система RDDF⁷ [13] — это система обнаружения скомпрометированных коммутаторов, использующая тестовые пакеты для проверки работы правил маршрутизации в сети. В этой системе использован алгоритм оптимизации процедуры обнаружения, в котором один тестовый пакет может проверить более одного коммутатора. Оптимизация достигается за счет агрегации правил маршрутизации с совпадающими или непересекающимися полями *match*.

Как и система ATPG, эта система генерирует тестовые пакеты на основе информации о правилах маршрутизации. Для каждого тестового пакета эта система вычисляет маршрут, который этот пакет должен пройти, и сравнивает его с реальным маршрутом, пройденным пакетом.

Процедура обнаружения оптимизируется за счет объединения нескольких тестовых пакетов в один. Маршруты для тестовых пакетов представляются так называемыми деревьями агрегации (*aggregation trees*). Дерево агрегации — это дерево, в котором вершины соответствуют коммутаторам, а каждый путь от корня к листу описывает путь для тестового пакета в сети.

Дерево агрегации содержит:

- начальный коммутатор — коммутатор, который является стартовой точкой движения тестовых пакетов;
- промежуточные коммутаторы — коммутаторы, которые дублируют тестовые пакеты на различные порты;
- листовые коммутаторы — коммутаторы, отправляющие тестовые пакеты на контроллер.

Для каждого дерева агрегации в сети устанавливаются правила маршрутизации из так называемой группы агрегации. Группой агрегации называется множество правил маршрутизации, которые удовлетворяют условиям:

- правила маршрутизации из одной и той же группы на различных коммутаторах имеют либо совпадающие, либо непересекающиеся поля *match*;
- тестовый пакет не должен проходить через один и тот же коммутатор дважды;
- правило маршрутизации *A* может принадлежать нескольким группам агрегации только в случае, если существует правило маршрутизации *B* на другом коммутаторе такое, что правила *A* и *B* имеют совпадающие поля *match*.

Для построения деревьев агрегации необходимо найти минимальное число групп агрегации при условии, что все правила маршрутизации в сети принадлежат хотя бы одной группе агрегации. Эта задача эквивалентна задаче нахождения минимального вершинного покрытия для ориентированного графа, которая, как доказано в работе [13], является NP-трудной.

Каждый тестовый пакет создается так, чтобы он мог быть обработан всеми правилами маршрутизации только из одной и той же группы агрегации. Такой тестовый пакет движется по дереву маршрутов, дублируется в промежуточных вершинах и отправляется на контроллер в листовых вершинах дерева. Если в сети нет скрытых правил, то все тестовые пакеты должны быть отправлены на контроллер из листовых вершин. Таким образом, контроллер может проверить, что каждый коммутатор выполняет все правила маршрутизации.

Эта система обладает тем же недостатком, что и любая система, использующая тестовые пакеты — система не способна отличить вредоносный сброс пакетов, появившийся вследствие атаки на сеть, от легитимного сброса из-за перегрузок в сети и не способна обнаружить кратковременные атаки.

Таким образом, система RDDF не удовлетворяет критериям **К3** и **К5**.

2.9. WedgeTail

Система WedgeTail [14] анализирует маршруты, пройденные пакетами в сети, для обнаружения скрытых правил маршрутизации. Эта система перехватывает OpenFlow-сообщения, передаваемые между контроллером и коммутаторами, и строит у себя виртуальную копию сети, которая затем используется для вычисления теоретических маршрутов движе-

⁷ Rapid Detection of Disobedient Forwarding.

ния пакетов в сети. Теоретические маршруты вычисляются с помощью алгоритма вычисления достижимости из модели Header Space Analysis [22].

Для определения реальных маршрутов, пройденных пакетами, система использует систему NetSight [23]. NetSight — это решение для обнаружения аномалий в сети, которое позволяет ПКС-приложению получать информацию о пройденных пакетом коммутаторах. NetSight требует модификации контура данных для того, чтобы генерировать отчеты о пройденных пакетом коммутаторах.

После сбора информации о реальных маршрутах, пройденных пакетами, система WedgeTail сравнивает эти маршруты с теоретическими. Если реальные маршруты пакетов не являются подмножеством теоретических маршрутов, то некоторые коммутаторы на пути пакета являются скомпрометированными.

Для ускорения обнаружения скомпрометированных коммутаторов система WedgeTail упорядочивает коммутаторы на основе приоритета. Приоритет вычисляется на основании трафика, обрабатываемого коммутатором. Основная идея заключается в том, что проверка должна проводиться в первую очередь на тех коммутаторах, которые обрабатывают большее число пакетов. Для определения таких коммутаторов WedgeTail отслеживает траектории всех пакетов на всех портах и определяет наиболее часто задействованные коммутаторы.

Одним из недостатков системы WedgeTail является то, что для ее использования необходимо изменять логику работы коммутатора, потому что эта система использует NetSight для сбора информации о маршрутах, пройденных пакетами. Таким образом, ограничивается область применения этой системы в существующих сетях.

Система WedgeTail может не обнаружить кратковременные атаки из-за того, что проверка проводится поочередно на различных коммутаторах в сети. Также система WedgeTail не разграничивает легитимный и вредоносный сброс трафика.

Таким образом, система WedgeTail не удовлетворяет критериям *K3*, *K4* и *K5*.

2.10. DYNAPFV

DYNAPFV [15] — это система для проверки корректности обработки пакетов на коммутаторах. Первый коммутатор, обрабатывающий пакет, называется входным для этого пакета. Выходным коммутатором для пакета, называется коммутатор, который отправил этот па-

кет во вне сети, например, на некоторый хост. Проверка происходит с помощью сравнения числа пакетов, полученных на входных и выходных коммутаторах.

Система устанавливает на граничные коммутаторы правила маршрутизации, которые с помощью *packet-in*-сообщений отправляют пакеты на контроллер для анализа. Система выбирает случайные промежутки времени для сбора пакетов из различных потоков, т. е. на контроллер отправляется только часть пакетов, проходящих в сети.

Каждый пакет, пришедший на контроллер в *packet-in*-сообщении, сохраняется до того момента, пока этот пакет не придет в другом сообщении с выходного коммутатора, либо когда истечет время, равное максимальному времени приема-передачи (*RTT*) в сети. Таким образом, происходит проверка того, что пакеты не были модифицированы в процессе их обработки промежуточными коммутаторами.

Если отношение числа корректно обработанных пакетов (которые пришли и со входного и с выходного коммутаторов) к числу всех пакетов, отправленных на контроллер, превышает заранее заданное пороговое значение, то система сообщает об ошибке, и начинается поиск скомпрометированных коммутаторов.

Во время поиска скомпрометированных коммутаторов выбирается набор промежуточных коммутаторов между входным и выходным коммутаторами, на которых удаляются соответствующие потоковые правила так, чтобы эти коммутаторы начинали отправлять пакеты на контроллер.

Также система проверяет корректность обработки пакетов в случае, когда таймауты потоковых правил истекают. Система извлекает соответствующие значения сетевой статистики и проверяет корректность обработки пакетов, сравнивая значения статистики с разных коммутаторов. Если отношение числа пакетов, обработанных на некотором коммутаторе S , к числу пакетов, обработанных входным коммутатором, меньше заранее определенного порогового значения, то коммутатор, предшествующий S , считается скомпрометированным.

Для снижения нагрузки на управляющий канал в системе используется динамическая настройка вероятности сбора пакетов для различных потоков.

Недостатком системы DYNAPFV является тот факт, что рассматривается только одна процедура маршрутизации — динамическая установка правил на коммутатор в случае появления пакета, для которого нет правила об-

работки. Кроме того, не учитываются пакеты, сброшенные из-за перегрузки сети, т. е. наличие легитимно сброшенного трафика будет интерпретировано системой как атака на сеть.

Из-за того, что система DYNAPFV запрашивает большое число пакетов из сети, она может негативно повлиять на производительность контроллера и на загрузку управляющего канала.

Система DYNAPFV не обнаруживает атаки, когда в сети есть несколько скомпрометированных коммутаторов, которые передают на контроллер некорректные данные о пакетах, проходящих в сети.

Таким образом, система DYNAPFV не удовлетворяет критериям **K1**, **K2**, **K3** и **K6**.

3. Сравнительный анализ

В таблице представлены сводные результаты анализа и сравнение систем обнаружения скомпрометированных ПКС коммутаторов, приведенные выше.

Сравнение систем обнаружения скомпрометированных коммутаторов

	K1	K2	K3	K4	K5	K6	K7
ATPG	✓	✓		✓		✓	✓
FADE		✓		✓	✓		
FlowMon				✓		✓	✓
FDWD		✓		✓		✓	
MLPC				✓			
PDMD				✓			✓
SPHINX	✓	✓		✓	✓		✓
RDDF	✓	✓		✓		✓	✓
WedgeTail	✓	✓				✓	✓
DYNAPFV				✓	✓		✓

Из этой таблицы хорошо видно, что существующие системы обнаружения скомпрометированных коммутаторов в ПКС используют три основных механизма обнаружения:

- анализ значений счетчиков правил маршрутизации;
- генерацию тестовых пакетов и проверку маршрутов, которые они проходят;
- сбор существующих пакетов и анализ их маршрутов.

Одним из недостатков существующих систем обнаружения скомпрометированных ком-

мутаторов заключается в том, что подобные системы рассматривают только самый простой механизм обработки пакетов в сети — динамическую установку отдельных правил маршрутизации для каждого потока в сети. Не учитывается групповая маршрутизация, агрегация правил и балансировка нагрузки. Таким образом, не учитывается тот факт, что ПКС позволяет реализовывать различные сложные механизмы маршрутизации и оптимизации потоков в сети. Сложные механизмы маршрутизации могут привести к очень непростым комбинациям потоковых правил, установленных на коммутаторах. Поэтому необходим алгоритм обнаружения скомпрометированных коммутаторов, который сможет работать при произвольной комбинации потоковых правил в сети.

Из таблицы также видно, что ни одна из существующих систем не способна отличить вредоносный сброс пакетов от легитимного сброса из-за перегрузок в сети. Поскольку перегрузки в сети — достаточно частое явление, то невозможность выявлять причину сброса пакета будет вызывать большое число ошибок второго рода.

Системы, использующие тестовые пакеты или сбор существующих пакетов, не могут различать типы сброса трафика из-за того, что протокол OpenFlow не предполагает отправки коммутатором на контроллер уведомления о каждом сброшенном пакете. Даже учитывая тот факт, что информация о числе сброшенных пакетов записана в статистике порта, у контроллера нет информации о том, что был сброшен именно тестовый пакет.

Также существующие системы, которые используют анализ сетевой статистики, не учитывают информацию о числе сброшенных пакетов, которая хранится в счетчиках портов.

Еще одной проблемой существующих систем, которые анализируют сетевую статистику, является уязвимость к сокрытию факта атаки, т. е. злоумышленник, сбрасывая некоторое число пакетов, может при этом генерировать такое же число новых пакетов для того, чтобы правило сохранения потока на коммутаторе выполнялось.

Кроме того, системы, применяющие анализ сетевой статистики, используют статистику только с граничных коммутаторов по отношению к проверяемому. Таким образом, если атакующий скомпрометировал несколько соседних коммутаторов, у него появляется возможность отправлять контроллеру некорректную сетевую статистику так, что атака не будет обнаружена.

Заключение

В данной статье проведен обзор и сравнение существующих систем обнаружения скомпрометированных ПКС коммутаторов. Сравнение проводилось по определенным критериям, которые охватывают различные аспекты систем обнаружения. Также в обзоре были выделены основные недостатки существующих систем. Необходимо отметить, что в обзоре рассмотрены только системы обнаружения скомпрометированных коммутаторов, которые уже были скомпрометированы и использованы для различных атак на контур данных. Системы обнаружения процесса компрометации не рассмотрены, так как эта область информационной безопасности уже является хорошо исследованной [16].

Из проведенного обзора средств обнаружения скомпрометированных коммутаторов в ПКС следует, что для надежного выявления скомпрометированных коммутаторов необходимо разработать систему, которая будет работать при условии использования в сети различных сложных механизмов маршрутизации и оставаться прозрачной для атакующего. Для этого система должна не зависеть от внутренней логики работы контроллера. Система, удовлетворяющая этим требованиям, должна строить представление сети на основе *OpenFlow* сообщений, отправленных контроллером на коммутаторы.

Важно, чтобы система обнаружения учитывала информацию о пакетах, сброшенных вследствие перегрузки в сети. Это необходимо для того, чтобы избежать ошибок второго рода во время перегрузок.

Список литературы

1. Kreutz D., Ramos F. M., Verissimo P. E., Rothenberg C. E., Azodolmolky S., Uhlig S. Software-defined networking: A comprehensive survey // Proceedings of the IEEE. 2015. Vol. 103, N. 1. P. 14–76.
2. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Turner J. OpenFlow: enabling innovation in campus networks // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38, N. 2. P. 69–74.
3. Петров И. С. Задача обнаружения скомпрометированных коммутаторов в SDN сетях // REDS: Телекоммуникационные устройства и системы. 2017. Т. 7, № 4. С. 515–518.
4. Scott-Hayward S., O'Callaghan G., Sezer S. SDN security: A survey // Future Networks and Services (SDN4FNS), 2013 IEEE SDN For. IEEE, 2013. P. 1–7.
5. Шендяпин А. С., Петров И. С. Исследование методов проведения атаки Man-in-the-Middle в программно-конфигурируемых сетях // Программные системы и инструменты.

Тематический сборник / Под общей редакцией Р. Л. Смелянского. 2017. С. 73–82.

6. Zeng H., Kazemian P., Varghese G., McKeown N. Automatic test packet generation // Proceedings of the 8th international conference on Emerging networking experiments and technologies. ACM, 2012. P. 241–252.
7. Pang C., Jiang Y., Li Q. FADE: Detecting forwarding anomaly in software-defined networks // Communications (ICC), 2016 IEEE International Conference on. IEEE, 2016. P. 1–6.
8. Kamisiński A., Fung C. Flowmon: Detecting malicious switches in software-defined networks // Proceedings of the 2015 Workshop on Automated Decision Making for Active Cyber Defense. ACM, 2015. P. 39–45.
9. Chi P. W., Kuo C. T., Guo J. W., Lei C. L. How to detect a compromised sdn switch // Network Softwarization (NetSoft), 2015 1st IEEE Conference on. IEEE, 2015. P. 1–6.
10. Chao T. W., Ke Y. M., Chen B. H., Chen J. L., Hsieh C. J., Lee S. C., Hsiao H. C. Securing data planes in software-defined networks // NetSoft Conference and Workshops (NetSoft), 2016 IEEE. IEEE, 2016. P. 465–470.
11. Mohammadi A. A., Kazemian P., Pakravan M. R. Detecting malicious packet drops and misroutings using Header Space Analysis // Telecommunications (IST), 2016 8th International Symposium on. IEEE, 2016. P. 521–526.
12. Dhawan M., Poddar R., Mahajan K., Mann V. SPHINX: Detecting Security Attacks in Software-Defined Networks // NDSS. 2015. Vol. 15. P. 8–11.
13. Chiu Y. C., Lin P. C. Rapid detection of disobedient forwarding on compromised OpenFlow switches // Computing, Networking and Communications (ICNC), 2017 International Conference on. IEEE, 2017. P. 672–677.
14. Shaghghi A., Kaafar M. A., Jha S. Wedgetail: An intrusion prevention system for the data plane of software defined networks // Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017. P. 849–861.
15. Li Q., Zou X., Huang Q., Zheng J., Lee P. P. Dynamic Packet Forwarding Verification in SDN // IEEE Transactions on Dependable and Secure Computing. 2018.
16. Intrusion detection systems: A survey and taxonomy. Technical report, 2000. Vol. 99.
17. Петров И. С., Смелянский Р. Л. Минимизация группового трафика и обеспечение его отказоустойчивости в программно-конфигурируемых сетях // Известия Российской академии наук. Теория и системы управления. 2018. № 3. С. 64–75.
18. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to algorithms. Cambridge: MIT press, 2001. Vol. 6.
19. Dilworth R. P. A decomposition theorem for partially ordered sets // Annals of Mathematics. 1950. P. 161–166.
20. Hopcroft J. E., Karp R. M. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs // SIAM Journal on computing. 1973. Vol. 2, N. 4. P. 225–231.
21. Duffield N. G., Grossglauser M. Trajectory sampling for direct traffic observation // IEEE/ACM Transactions on Networking (ToN). 2001. Vol. 9, N. 3. P. 280–292.
22. Kazemian P., Varghese G., McKeown N. Header Space Analysis: Static Checking for Networks // NSDI. 2012. Vol. 12. P. 113–126.
23. Handigol N., Heller B., Jeyakumar V., Mazières D., McKeown N. I Know What Your Packet Did Last Hop: Using Packet Histories to Troubleshoot Networks // NSDI. 2014. Vol. 14. P. 71–85.

Systems for Compromised Switch Detection in Software-Defined Networking

Compromised network switches present a serious threat for the security of the network since they can be used by an attacker to perform a wide range of network attacks. This paper presents a survey and comparison of existing systems for compromised switch detection in software-defined networking, shows their main advantages and disadvantages.

Keywords: software-defined networking, intrusion detection systems, compromised network switches

DOI: 10.17587/it.25.131-142

References

1. Kreutz D., Ramos F. M., Verissimo P. E., Rothenberg C. E., Azodolmolky S., Uhlig S. Software-defined networking: A comprehensive survey, *Proceedings of the IEEE*, 2015, vol. 103, no. 1, pp. 14–76.
2. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Turner J. OpenFlow: enabling innovation in campus networks, *ACM SIGCOMM Computer Communication Review*, 2008, vol. 38, no. 2, pp. 69–74.
3. Petrov I. S. Zadacha obnaruzhenija skomprometirovannykh kommutatorov v SDN setjah (The problem of detecting compromised switches in SDN), *REDS: Telekom Systems and Infrastructure*, 2017, vol. 7, no. 4, pp. 515–518 (in Russian).
4. Scott-Hayward S., O’Callaghan G., Sezer S. SDN security: A survey, *Future Networks and Services (SDN4FNS)*, 2013 IEEE SDN For, IEEE, 2013, pp. 1–7.
5. Shendiapin A. S., Petrov I. S. Issledovanie metodov provedeniya ataki Man-in-the-Middle v programmno-konfiguriruemyykh setjah (Research of the Man-in-the-Middle in Software-Defined Networks), *Journal of Program Systems and Instruments*, Moscow, 2017, pp. 73–8 (in Russian).
6. Zeng H., Kazemian P., Varghese G., McKeown N. Automatic test packet generation, *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, ACM, 2012, pp. 241–252.
7. Pang C., Jiang Y., Li Q. FADE: Detecting forwarding anomaly in software-defined networks, *Communications (ICC), 2016 IEEE International Conference on*, IEEE, 2016, pp. 1–6.
8. Kamisiński A., Fung C. Flowmon: Detecting malicious switches in software-defined networks, *Proceedings of the 2015 Workshop on Automated Decision Making for Active Cyber Defense*, ACM, 2015, pp. 39–45.
9. Chi P. W., Kuo C. T., Guo J. W., Lei C. L. How to detect a compromised sdn switch, *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, IEEE, 2015, pp. 1–6.
10. Chao T. W., Ke Y. M., Chen B. H., Chen J. L., Hsieh C. J., Lee S. C., Hsiao H. C. Securing data planes in software-defined networks, *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*, IEEE, 2016, pp. 465–470.
11. Mohammadi A. A., Kazemian P., Pakravan M. R. Detecting malicious packet drops and misroutings using Header Space Analysis, *Telecommunications (IST), 2016 8th International Symposium on*, IEEE, 2016, pp. 521–526.
12. Dhawan M., Poddar R., Mahajan K., Mann V. SPHINX: Detecting Security Attacks in Software-Defined Networks, *NDSS*, 2015, vol. 15, pp. 8–11.
13. Chiu Y. C., Lin P. C. Rapid detection of disobedient forwarding on compromised OpenFlow switches, *Computing, Networking and Communications (ICNC), 2017 International Conference on*, IEEE, 2017, pp. 672–677.
14. Shaghaghi A., Kaafar M. A., Jha S. Wedgetail: An intrusion prevention system for the data plane of software defined networks, *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ACM, 2017, C. 849–861.
15. Li Q., Zou X., Huang Q., Zheng J., Lee P. P. Dynamic Packet Forwarding Verification in SDN, *IEEE Transactions on Dependable and Secure Computing*, 2018.
16. Axelsson S. Intrusion detection systems: A survey and taxonomy, *Technical report*, 2000, vol. 99.
17. Petrov I. S., Smeliansky R. L. Minimizatsiya gruppovogo trafika i obespechenie ego otkazoustojchivosti v programmno-konfiguriruemyykh setjah (Minimization of Multicast Traffic and Ensuring Its Fault Tolerance in Software-Defined Networks), *Journal of Computer and Systems Sciences International*, 2018, vol. 57, no. 3, pp. 407–419 (in Russian).
18. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to algorithms, Cambridge, MIT press, 2001, vol. 6.
19. Dilworth R. P. A decomposition theorem for partially ordered sets, *Annals of Mathematics*, 1950, pp. 161–166.
20. Hopcroft J. E., Karp R. M. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM Journal on computing*, 1973, vol. 2, no. 4, pp. 225–231.
21. Duffield N. G., Grossglauser M. Trajectory sampling for direct traffic observation, *IEEE/ACM Transactions on Networking (ToN)*, 2001, vol. 9, no. 3, pp. 280–292.
22. Kazemian P., Varghese G., McKeown N. Header Space Analysis: Static Checking for Networks, *NSDI*, 2012, vol. 12, pp. 113–126.
23. Handigol N., Heller B., Jeyakumar V., Mazières D., McKeown N. I Know What Your Packet Did Last Hop: Using Packet Histories to Troubleshoot Networks, *NSDI*, 2014, vol. 14, pp. 71–85.

Д. В. Ефанов, д-р техн. наук, доц., e-mail: TrES-4b@yandex.ru,
ООО "ЛокоТех-Сигнал",
Российский университет транспорта (МИИТ), Москва

Условия построения полностью самопроверяемых дискретных устройств на основе метода логического дополнения до равновесного кода "2 из 4"

В статье показано, что следуя известному методу логического дополнения, дающему разработчику самопроверяемых дискретных устройств более широкий спектр возможностей, чем традиционные подходы, необходимо учитывать ограничения, накладываемые на структуры как объекта диагностирования, так и блока контрольной логики. Установлены особенности, которые присущи методу логического дополнения, а именно, использованию в нем каскада параллельно расположенных сумматоров по модулю два. Приводятся примеры, иллюстрирующие данные особенности. Сформулированы практические рекомендации, которых необходимо придерживаться при разработке самопроверяемых дискретных устройств по методу логического дополнения до равновесного кода "2 из 4".

Ключевые слова: самопроверяемые дискретные устройства, самопроверяемая схема встроенного контроля, метод логического дополнения, равновесный код, код "2 из 4", условия обнаружения неисправности

Введение

Непрерывное развитие в области информационных и компьютерных технологий связано с постоянным совершенствованием элементов и систем, реализованных на микроэлектронных компонентах. С каждым годом производители по всему миру объявляют о новейших достижениях в области микроэлектроники: уменьшаются габаритные размеры элементов, достигаются техпроцессы реализации длины затвора транзистора в 10 и даже 7 нм, ведутся исследования в области реализации техпроцессов по технологии 3 нм [1–3]. Все это ведет к уменьшению габаритных размеров реализуемых устройств автоматики и вычислительной техники, способствует увеличению рабочих частот, повышению функциональности, быстродействия, вычислительной мощности и т. д. Соответственно развитию технологий должны развиваться и методы технической диагностики и обеспечения надежности и безопасности функционирования реализуемых устройств [4–6].

Одним из важнейших методов технической диагностики является функциональное (рабочее) диагностирование, позволяющее проводить процедуру технического диагностирования устройства без отключения его от выполнения основных функций. При этом рабочие воздействия на объект диагностирования оказываются одновременно и тестовыми [7]. Использование рабочего диагностирования позволяет оперативно реагировать на возникающие в процессе эксплуатации неисправности в объ-

екте диагностирования, блокировать неверные результаты вычислений и отключать некорректно функционирующие выходы устройств от работы [8]. Этим обеспечивается безопасность реализуемого технологического процесса.

При синтезе систем рабочего диагностирования используются идеи кратного и дробного резервирования [9]. Сам объект диагностирования снабжается специализированной схемой контроля, параллельно проводящей вычисления контрольных функций и процедуры сравнения их значений со значениями рабочих функций объекта диагностирования. На этапе проектирования системы диагностирования устанавливается однозначное соответствие между значениями рабочих и контрольных функций на каждом входном наборе. Это позволяет косвенно, по результатам вычислений, определять неисправности в системе диагностирования и сигнализировать об их наличии, а также использовать данную информацию для отключения объекта диагностирования от работы. Поскольку схема контроля наделена ответственными функциями по подтверждению корректности формируемых на рабочих выходах значений, она должна иметь самопроверяемую структуру [10]. Это обстоятельство накладывает определенные ограничения на способы реализации схем контроля в системах рабочего диагностирования.

Данная статья посвящена развитию теории синтеза самопроверяемых дискретных устройств автоматики и вычислительной техники на основе метода логического дополнения [11].

1. Основные методы синтеза самопроверяемых схем встроенного контроля

Традиционная структурная схема, по которой реализуются системы рабочего диагностирования, изображена на рис. 1 [12]. Она копирует идею передачи информации по каналам связи и основана на использовании свойств равномерных двоичных кодов [13]. В традиционной структурной схеме выходы объекта диагностирования $F(x)$ отождествляются с некоторым информационным вектором $\langle f_1 f_2 \dots f_m \rangle$ длиной m , а схема контроля синтезируется с учетом свойств выбранного избыточного кода. Блок контрольной логики вычисляет ряд контрольных функций, образуя контрольный вектор $\langle g_1 g_2 \dots g_k \rangle$ длиной k , а тестер выбранного кода в процессе эксплуатации системы диагностирования сравнивает информационный и контрольный векторы, формируя контрольный сигнал $\langle z^0 z^1 \rangle$. При отсутствии неисправностей в системе диагностирования на выходе тестера формируется парафазный сигнал $\langle z^0 z^1 \rangle = \langle 01 \rangle$ либо $\langle z^0 z^1 \rangle = \langle 10 \rangle$. Нарушение парафазности свидетельствует о наличии неисправности (или искажения значений вследствие, например, электромагнитных воздействий) в системе диагностирования.

При синтезе системы диагностирования по структурной схеме, приведенной на рис. 1, могут использоваться любые равномерные коды: разделимые или неразделимые. В случае применения последних контрольные функции дополняют информационные таким образом, чтобы на входы тестера поступало кодовое слово неразделимого кода. Основными неразделимыми кодами для систем рабочего диагностирования являются равновесные коды [14]. К основным разделимым кодам относятся

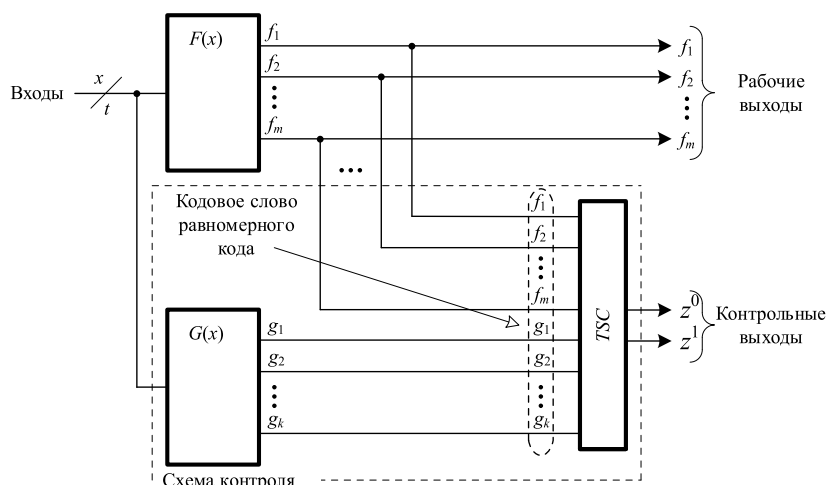


Рис. 1. Традиционная структурная схема системы диагностирования

коды с повторением (код с простым повторением, инверсный код, корреляционный код) и коды с суммированием (коды Хэмминга и Бергера и их модификации) [15—17].

В автоматике и вычислительной технике широко распространена стандартная структура дублирования, которая реализуется на основе кода с простым повторением [18]. В такой системе диагностирования блок основной логики $F(x)$ снабжается своей копией, а тестер схемы контроля проверяет соответствие одноименных разрядов обоих блоков. Для реализации самопроверяемой схемы контроля выходы блока контрольной логики предварительно инвертируются, что позволяет реализовывать тестер в виде самопроверяемого компаратора, представляющего собой схему сжатия парафазных сигналов. Такой компаратор реализуется наиболее просто в виде каскадного подключения элементарных модулей сжатия парафазных сигналов TRC (*two-rail checker*) [19].

При использовании дублирования в объекте диагностирования обнаруживаются любые неисправности, вызывающие искажения на его выходах. Однако существенным недостатком такого подхода к построению системы диагностирования является высокая структурная избыточность — для реальных устройств она может превышать величину в 300...400 % от сложности объекта диагностирования.

В целях снижения структурной избыточности конечного устройства разработчики стремятся уменьшить сложность технической реализации схемы контроля. Это возможно за счет снижения требований к классам идентифицируемых в устройстве неисправностей. Например, если разработчик ограничивается моделью одиночных константных неисправностей выходов внутренних логических элементов

или же моделью "временной задержки" пути, то возможно эффективное снижение структурной избыточности конечного устройства [9]. В этих случаях при построении системы диагностирования возможно применение разнообразных равномерных кодов с избыточностью, меньшей избыточности кода с повторением.

Известны методы синтеза схем контроля на основе кодов паритета [9]. Они подразумевают поиск групп независимых выходов (Н-групп) на множестве выходов объекта диагностирования и отдельный их контроль на основе кода паритета с последующим объединением выходов схем контроля на входах самопроверяемого компара-

тора. Другим вариантом реализации контрольного оборудования является преобразование схемы самого объекта диагностирования в схему с независимыми выходами. Аналогично используются классические коды Бергера [20—22], за тем лишь исключением, что группы выходов являются монотонно независимыми (или монотонно и асимметрично независимыми) — так называемые МН-группы (или МАН-группы) [23]. При использовании классических кодов Хэмминга [24] ищутся группы выходов, в которых допускаются одно- и двукратные искажения (H^2 -группы). Во всех этих случаях используются свойства кодов, связанные с обнаружением ошибок определенной кратностью или вида [25].

Структурная схема, приведенная на рис. 1, "жестко" связана с тем, какой именно равномерный код выбран при ее реализации. Более гибкой с точки зрения синтеза схем контроля является структурная схема метода логического дополнения [26]. В такой схеме помимо блоков, имеющих в традиционной структурной схеме, использован блок логического дополнения (рис. 2). Данный блок реализуется в виде каскада параллельно расположенных сумматоров по модулю два (элементов XOR), позволяющих осуществлять преобразование любого информационного вектора в кодовое слово заранее выбранного кода. В качестве основы системы диагностирования в случае использования метода логического дополнения гарантированно подходят только неразделимые коды. Это связано с особенностью реализации схемы контроля — в ней значения всех разрядов вычисляются объектом диагностирования, а значит, при использовании делимого кода существует возможность одновременного искажения и информационных и контрольных разрядов (возникновения

необнаруживаемой ошибки). Кроме неразделимых кодов при синтезе схем контроля на основе метода логического дополнения могут использоваться особые классы функций алгебры логики, например, самодвойственные функции [27, 28].

Вопросам синтеза схем контроля на основе метода логического дополнения посвящено множество публикаций, например [29—32]. Однако до сих пор не исследованным остается вопрос, связанный с тем, какие ограничения накладываются на структуры блоков основной и контрольной логики для достижения свойства самопроверяемости схемы контроля и синтезируемого дискретного устройства. Установим эти условия для использования при синтезе системы рабочего диагностирования по методу логического дополнения равновесного кода "2 из 4" (2/4-кода).

2. Метод логического дополнения до равновесного кода "2 из 4"

При построении системы диагностирования по методу логического дополнения наиболее удобно применять неразделимые равновесные коды с малой длиной кодовых слов. У таких кодов достаточно простые структурные схемы тестеров, для полной проверки которых требуется небольшое число тестовых комбинаций. К одним из таких кодов относится 2/4-код.

Использование описываемого метода подразумевает разбиение выходов блока $F(x)$ на подмножества по четыре выхода в каждом (возможны пересечения подмножеств). Для каждого подмножества синтезируется своя подсхема контроля, а выходы всех схем контроля объединяются на входах самопроверяемого компаратора [19]. Базовая структурная схема системы диагностирования, реализованная для группы из четырех выходов, изображена на рис. 3.

В ней блок $F(x)$ представляет собой объект диагностирования или его подсхему с четырьмя выходами. Контролируемая подсхема устройства снабжается схемой контроля в составе трех блоков: контрольной логики ($G(x)$), логического дополнения — блока, образованного каскадом двух сумматоров по модулю два, тестера 2/4-кода (2/4-TSC).

Каскад сумматоров по модулю два позволяет преобразовывать любой четырехбитный вектор рабочих функций в вектор, принадлежащий 2/4-коду.

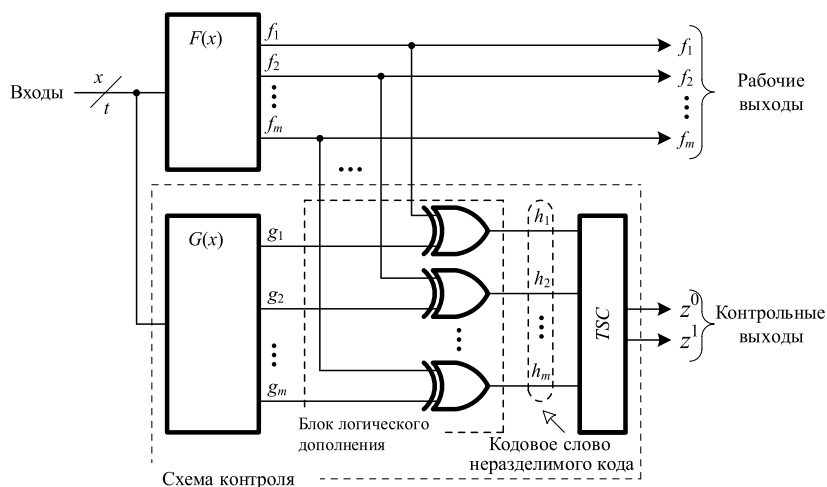


Рис. 2. Структурная схема системы диагностирования на основе метода логического дополнения

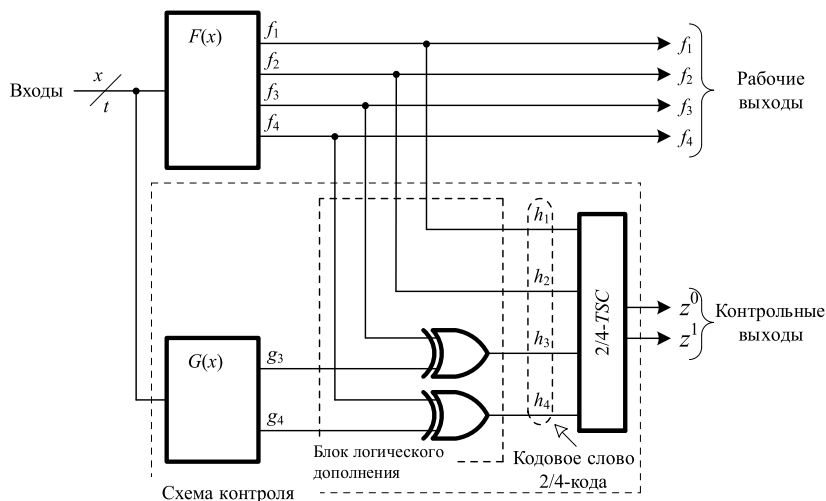


Рис. 3. Базовая структурная схема системы диагностирования, реализованная по методу логического дополнения до 2/4-кода

Условия синтеза систем диагностирования с самопроверяемыми блоками логического дополнения и 2/4-TSC рассмотрены в работах [33–35]. Остановимся на рассмотрении условий обеспечения тестируемости блоков основной и контрольной логики в системе диагностирования.

3. Условия обеспечения самопроверяемости схемы контроля

Определим условия, при которых базовая структурная схема системы диагностирования, приведенная на рис. 3, будет обладать свойством полной самопроверяемости.

Система диагностирования будет полностью самопроверяемой относительно одиночных константных неисправностей в том случае, если любая неисправность блоков $F(x)$ и $G(x)$ будет хотя бы на одном входном наборе вызывать искажения в значениях выходных сигналов, и это событие будет зафиксировано на выходах тестера 2/4-TSC в виде установления непарафазных сигналов. Кроме того, тестер системы должен иметь самопроверяемую структуру, а также должны обнаруживаться любые неисправности элементов блока логического дополнения. Последние два условия требуют при реализации системы диагностирования обеспечивать подачу на каждый элемент XOR блока логического дополнения тестовых комбинаций из множества {00, 01, 10, 11} [36], а также формирования множества тестовых комбинаций для полной проверки 2/4-TSC. Для наиболее простой структуры данного тестера это множество образуется комбинациями {0011, 1100, 1001, 0110} [37].

Рассмотрим подробнее особенности функционирования блоков $F(x)$ и $G(x)$ и те условия, при которых на выходах блока логического дополнения не будут возникать необнаруживаемые тестером искажения (т. е. не будут формироваться кодовые слова 2/4-кода).

Теорема 1. Для того чтобы неисправности блоков основной и контрольной логики в системе диагностирования, синтезированной по методу логического дополнения, были обнаружены, необходимо и достаточно, чтобы они не вызывали симметричных ошибок на выходах блока логического дополнения.

Доказательство. Необходимость.

В основе системы диагностирования лежит равновесный код (2/4-код).

Одиночная неисправность в системе диагностирования не будет обнаружена в том случае, если в кодовом слове, поступающем на входы самопроверяемого тестера, сохранится вес кодового слова при отсутствии неисправности в системе. Другими словами, неисправность не будет обнаружена в том случае, если вызовет симметричную ошибку (ошибку, содержащую одинаковое число искажений нулевых и единичных разрядов) [25] в кодовом слове, сформированном на выходах блока логического дополнения. Все остальные виды ошибок идентифицируются тестером равновесного кода.

Достаточность. В системе диагностирования имеется четыре независимых блока: блок основной логики, блок контрольной логики, блок логического дополнения и самопроверяемый тестер. Одиночные неисправности тестера обнаруживаются на его выходах при подаче на входы рабочих воздействий, включающих в себя и тестовые комбинации. Одиночные неисправности в блоке логического дополнения вызывают только однократные ошибки и также идентифицируемые тестером, так как нарушается вес кодового слова на входах тестера. Неисправности элементов XOR устанавливаются на рабочих воздействиях при формировании тестовых комбинаций из множества {00, 01, 10, 11}. Для полной проверки тестера и элементов XOR требуется формирование определенного числа информационных и контрольных векторов на выходах блоков $F(x)$ и $G(x)$.

Симметричная ошибка на выходах блока логического дополнения может быть вызвана либо неисправностью в блоке основной логики, либо неисправностью в блоке контрольной логики. Одновременное искажение выходов обоих блоков ввиду принятой модели неисправности невозможно. **Доказательство завершено.**

Условия тестируемости блоков логического дополнения и тестера известны [33—35]. Требуется установить особенности влияния одиночных неисправностей блоков основной и контрольной логики на выходы блока логического дополнения. При этом необходимо учитывать, что функции f_1 и f_2 (рис. 3) не преобразуются и напрямую подключаются к входам 2/4-TSC, а изменяются только значения функций f_3 и f_4 : $f_3 \oplus g_3 = h_3$ и $f_4 \oplus g_4 = h_4$.

Рассмотрим блок $F(x)$ и его выходы. Любая однократная и трехкратная ошибки в векторе $\langle f_1 f_2 f_3 f_4 \rangle$ и, соответственно, на входах блока логического дополнения будут проявляться в виде однократных и трехкратных ошибок соответственно на выходах блока логического дополнения. Это гарантированно вызывает искажение веса кодового слова $\langle h_1 h_2 h_3 h_4 \rangle$ при исправной работе объекта диагностирования. Таким образом, ошибка на выходах блока логического дополнения будет являться либо монотонной (содержать только однонаправленные искажения значений), либо асимметричной (содержать искажения и нулевых, и единичных разрядов при неравном их числе) [25], что обнаружится схемой 2/4-TSC.

Необнаруживаемыми могут оказаться двукратные и четырехкратные ошибки на выходах блока $F(x)$ в том случае, если они вызывают симметричные ошибки на выходах блока логического дополнения. Рассмотрим многообразие двукратных ошибок.

Двукратные ошибки могут возникать в трех принципиально различных вариантах:

1. При одновременном искажении не преобразуемых выходов.
2. При искажении не преобразуемого и преобразуемого выходов.
3. При одновременном искажении преобразуемых выходов.

В первом случае, когда ошибка возникла на не преобразуемых выходах f_a и f_b , она напрямую транслируется на выходы блока логического дополнения. Для того чтобы ошибка была обнаружена, необходимо и достаточно, чтобы до искажения значения выходов совпадали: $f_a = f_b$. Это гарантированно приведет к возникновению монотонной двукратной ошибки.

Второй случай связан с одновременным искажением не преобразуемого и преобразуемого выходов. Пусть преобразуется выход f_b с помощью функции g_b . Положим $g_b = 0$. Тогда $f_b \oplus g_b = f_b$. Таким образом, значение на входе f_b напрямую, без изменений, транслируется на выход блока логического дополнения. Отсюда следует, что при $f_a = f_b$ при наличии ошибки на обоих входах на выходы блока логического до-

полнения транслируется ошибка, содержащая либо только искажения $1 \rightarrow 0$, либо только $0 \rightarrow 1$. Подобные искажения соответствуют монотонной ошибке и будут обнаружены на входах тестера. Положим $g_b = 1$. Тогда $f_b \oplus 1 = \overline{f_b}$. Отсюда следует, что значение на выходе f_b при трансляции на выход блока логического дополнения инвертируется. Отсюда следует, что при $f_a \neq f_b$ и при наличии ошибки на обоих входах на выходы блока логического дополнения, так же как и в первом случае, транслируется ошибка, содержащая либо только искажения $1 \rightarrow 0$, либо только $0 \rightarrow 1$. Подобная ошибка является монотонной и обнаруживается на входах тестера.

В случае одновременного искажения обоих преобразуемых выходов накладывается следующее ограничение. Если $f_a \neq f_b$, то монотонная ошибка будет транслироваться на выходы блока логического дополнения только в том случае, если функции дополнения не будут равны друг другу. В этом случае одно из значений f_a (или f_b) будет инвертировано, а другое — нет. Аналогичное обстоятельство имеет место при $f_a = f_b$. Если на обоих выходах имеется ошибка, то она останется монотонной на выходах блока логического дополнения в случае $g_a = g_b$. Монотонная ошибка обнаружится на входах тестера.

Четырехкратная ошибка на не преобразуемых выходах f_a и f_b и на преобразуемых выходах f_c и f_d будет симметричной в любом случае. Это следует из того факта, что до возникновения ошибки вектор $\langle h_1 h_2 h_3 h_4 \rangle$ принадлежит 2/4-коду, а его четырехкратное искажение возможно только в вектор равновесного кода.

Теорема 2. Структура блока основной логики в системе диагностирования, организованной по методу логического дополнения до 2/4-кода, будет проверяемой в том случае, если:

- а) при условии одновременного искажения не преобразуемого и преобразуемого выходов блока основной логики $f_a = f_b$ при $g_b = 0$ и $f_a \neq f_b$ при $g_b = 1$ (f_b — преобразуемый выход);
- б) при условии одновременного искажения обоих преобразуемых выходов блока основной логики $f_a \neq f_b$ при $g_a \neq g_b$ и $f_a = f_b$ при $g_a = g_b$;
- в) будут исключены четырехкратные ошибки.

На рис. 4 и рис. 5 приводятся различные случаи распространения ошибок на выходы блока логического преобразователя, иллюстрирующие условия теоремы 2.

Аналогичные рассуждения могут быть проведены при анализе ошибок на выходах блока $G(x)$.

Теорема 3. Структура блока контрольной логики в системе диагностирования, организованной по методу логического дополнения до 2/4-кода, будет проверяемой в том случае, если при условии одновременного искажения обоих

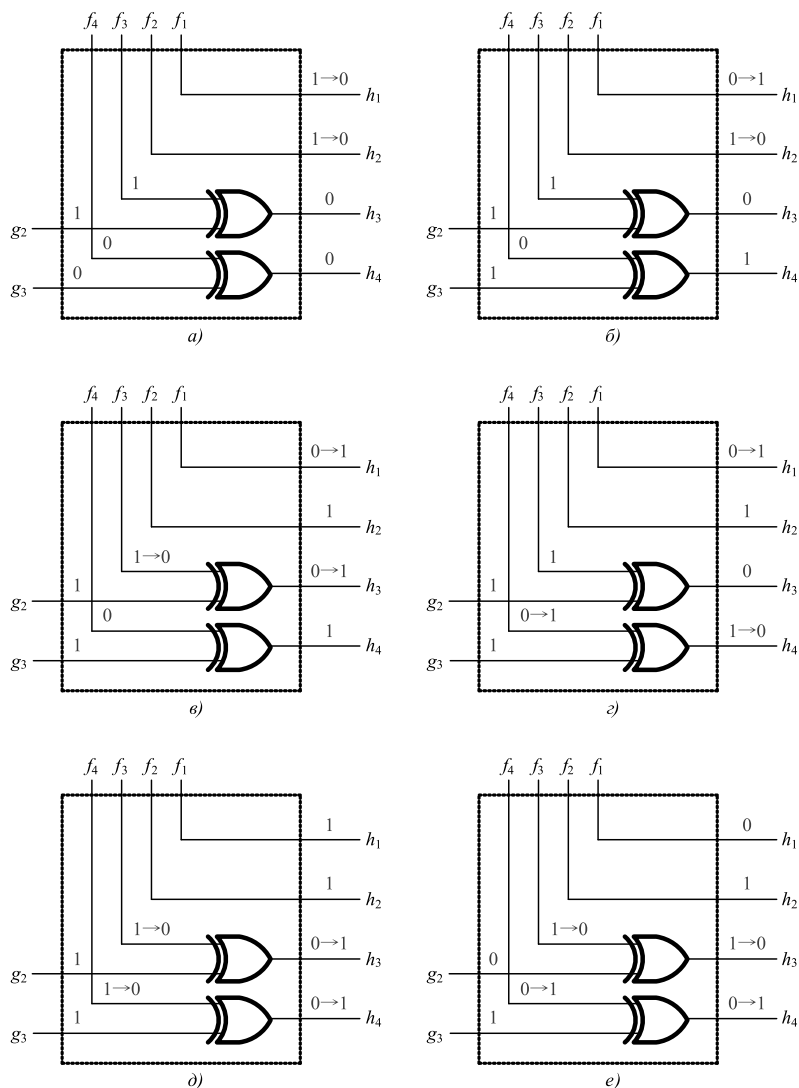


Рис. 4. Варианты трансляции двукратных ошибок на выходы блока логического преобразователя:

а — обнаруживаемая ошибка на не преобразуемых выходах; *б* — необнаруживаемая ошибка на не преобразуемых выходах; *в* — обнаруживаемая ошибка на преобразуемом и не преобразуемом выходах; *г* — необнаруживаемая ошибка на преобразуемом и не преобразуемом выходах; *д* — обнаруживаемая ошибка на преобразуемых выходах; *е* — необнаруживаемая ошибка на преобразуемых выходах

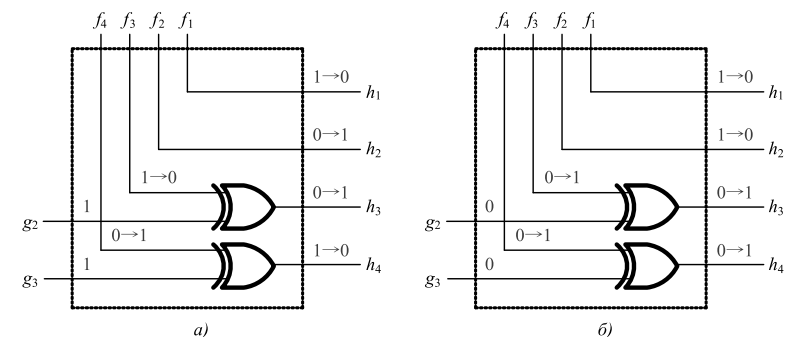


Рис. 5. Варианты трансляции четырехкратных ошибок на выходы логического преобразователя:

а — необнаруживаемая ошибка при неравенстве значений на преобразуемых выходах и не преобразуемых выходах; *б* — необнаруживаемая ошибка при равенстве значений на преобразуемых выходах и не преобразуемых выходах

функций дополнения $g_a = g_b$ при $f_a = f_b$ и $g_a \neq g_b$ при $f_a \neq f_b$.

Непосредственно из формулировок теорем 2 и 3 и известных условий тестируемости элементов XOR в блоке логического дополнения и условий тестируемости тестера следуют условия построения полностью самопроверяемой системы диагностирования.

Теорема 4. Система диагностирования, организованная по методу логического дополнения до 2/4-кода, будет самопроверяемой в том случае, если:

а) на входах тестера будут формироваться хотя бы по разу все необходимые тестовые комбинации (для наиболее простого тестера это комбинации из множества {0011, 1100, 1001, 0110});

б) на входах каждого сумматора в блоке логического дополнения будут формироваться хотя бы по разу комбинации проверяющего теста {00, 01, 10, 11};

в) блоки основной и контрольной логики будут реализованы в соответствии с условиями теорем 2 и 3.

Условия обеспечения полной самопроверяемости системы диагностирования

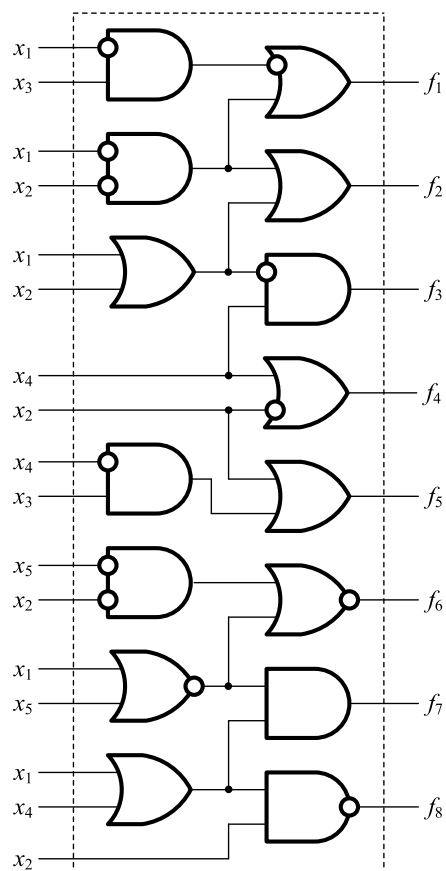


Рис. 6. Комбинационное дискретное устройство

вания на основе 2/4-кода являются достаточно жесткими. Требуется, чтобы блоки основной и контрольной логики были реализованы в виде устройств с независимыми выходами.

Для многовыходных объектов диагностирования решить задачу обеспечения полной самопроверяемости системы диагностирования оказывается несколько проще.

На рис. 6 для примера изображено комбинационное дискретное устройство с восемью выходами. Данную схему можно эффективно контролировать на основе 2/4-кода путем выделения двух групп выходов $\{f_1, f_3, f_5, f_7\}$ и $\{f_2, f_4, f_6, f_8\}$ при условии независимости функций логического дополнения отдельных подсхем контроля. Отдельные функции разных подсхем контроля могут быть реализованы совместно.

Заключение

Установленные в данной статье условия обеспечения полной самопроверяемости структурной схемы системы диагностирования на основе метода логического дополнения до равновесного кода "2 из 4" позволяют судить о возможности построения контролепригодных дискретных устройств на ее основе. Полученные условия являются "жесткими" и выполняются только для частных случаев объектов диагностирования с определенной структурой. Практически всегда потребуется реализация блока контрольной логики в виде схемы с независимыми выходами. Исключение составляет лишь возможность совместной реализации контрольных функций в разных подсхемах контроля для многовыходных объектов диагностирования. Обозначенные обстоятельства существенно сужают поле приложения метода логического дополнения. Тем не менее в ряде случаев удается синтезировать устройства с полностью самопроверяемыми структурами.

Следует отметить, что полученные результаты для использования 2/4-кода при построении систем диагностирования по методу логического дополнения могут быть легко обобщены до случая использования в обозначенных задачах и произвольных равновесных кодов.

Учет представленных особенностей систем диагностирования, реализованных по методу логического дополнения, целесообразен при разработке самопроверяемых дискретных устройств.

Список литературы

1. Chang K., Acharya K., Sinha S., Cline B., Yeric G., Lim S. K. Impact and Design Guideline of Monolithic 3-D IC at the 7-nm Technology Node // IEEE Transactions on Very Large Scale Inte-

gration (VLSI) Systems. 2017. Vol. 25, Iss. 7. P. 2118–2129. DOI: 10.1109/TVLSI.2017.2686426.

2. Yoon J.-S., Jeong E.-Y., Baek C.-K., Kim Y.-R., Hong J.-H., Lee J.-S., Baek R.-H., Jeong Y.-H. Junction Design Strategy for Si Bulk FinFETs for System-on-Chip Applications Down to the 7-nm Node // IEEE Electron Device Letters. 2015. Vol. 36, Iss. 10. P. 994–996. DOI: 10.1109/LED.2015.2464706.

3. Imec and Cadence Tape Out Industry's First 3nm Test Chip. URL: <https://www.hpcwire.com/off-the-wire/imec-cadence-tape-industrys-first-3nm-test-chip/>, дата обращения 19.08.2018.

4. Ubar R., Raik J., Vierhaus H.-T. Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source). Information Science Reference, Hershey — New York, IGI Global, 2011, 578 p.

5. Kharchenko V., Kondratenko Yu., Kacprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures // Springer Book series "Studies in Systems, Decision and Control". 2017. Vol. 74. 305 p.

6. Hahanov V. Cyber Physical Computing for IoT-driven Services. New York, Springer International Publishing AG, 2018. 279 p.

7. Пархоменко П. П., Согомоян Е. С. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства). М.: Энергоатомиздат, 1981. 320 с.

8. Дрозд А. В., Харченко В. С., Антошук С. Г., Дрозд Ю. В., Дрозд М. А., Сулима Ю. Ю. Рабочее диагностирование безопасных информационно-управляющих систем / Под ред. А. В. Дрозда и В. С. Харченко. Харьков: Национальный аэрокосмический университет им. Н. Е. Жуковского "ХАИ", 2012. 614 с.

9. Согомоян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 207 с.

10. Сапожников В. В., Сапожников Вл. В. Самопроверяемые дискретные устройства. СПб.: Энергоатомиздат, 1992. 224 с.

11. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science + Business Media B. V., 2008. 184 p.

12. Mitra S., McCluskey E. J. Which Concurrent Error Detection Scheme to Choose? // Proceedings of International Test Conference. 2000. USA, Atlantic City, NJ, 03–05 October 2000. P. 985–994. DOI: 10.1109/TEST.2000.894311.

13. Elias P. Computation in the Presence of Noise // IBM Journal of Research & Development. 1958. Vol. 2, Iss. 4. P. 346–353.

14. Freiman C. V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control. 1962. Vol. 5, Iss. 1. P. 64–71. DOI: 10.1016/S0019-9958(62)90223-1.

15. Hamming R. W. Error Detecting and Correcting Codes // Bell System Technical Journal. 1950. 29 (2). P. 147–160. MR0035935.

16. Berger J. M. A Note on Error Detecting Codes for Asymmetric Channels // Information and Control. 1961. Vol. 4, Iss. 1. P. 68–73. DOI: 10.1016/S0019-9958(61)80037-5.

17. Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej, 1995. 111 p.

18. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В., Дмитриев В. В. Новые структуры систем функционального контроля логических схем // Автоматика и телемеханика. 2017. № 2. С. 127–143.

19. Lala P. K. Self-Checking and Fault-Tolerant Digital Design. San Francisco: Morgan Kaufmann Publishers, 2001. 216 p.

20. Sogomonyan E. S., Gössel M. Self-Testing and Self-Checking Combinational Circuits with Weakly Independent Outputs // Proceedings of 10th IEEE VLSI Test Symposium, Atlantic City, 1992. P. 298–303.

21. Busaba F. Y., Lala P. K. Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors // Journal of Electronic Testing: Theory and Applications. 1994. Iss. 1. P. 19–28. DOI: 10.1007/BF00971960.

22. Morosow A., Sapozhnikov V. V., Sapozhnikov Vl. V., Goessel M. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. 1998. Vol. 5, Iss. 4. P. 333–345. DOI: 10.1155/1998/20389.

23. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Условия обнаружения неисправности логического элемента в комбинационном устройстве при функциональном кон-

троле на основе кода Бергера // Автоматика и телемеханика. 2017. № 5. С. 152—165.

24. **Ефанов Д. В., Сапожников В. В., Сапожников Вл. В.** Синтез самопроверяемых комбинационных устройств на основе выделения специальных групп выходов // Автоматика и телемеханика. 2018. № 9. С. 79—94.

25. **Сапожников В. В., Сапожников Вл. В., Ефанов Д. В.** Классификация ошибок в информационных векторах систематических кодов // Известия вузов. Приборостроение. 2015. Т. 58, № 5. С. 333—343. DOI: 10.17586/0021-3454-2015-58-5-333-343.

26. **Гессель М., Дмитриев А. В., Сапожников В. В., Сапожников Вл. В.** Самотестируемая структура для функционального обнаружения отказов в комбинационных схемах // Автоматика и телемеханика. 1999. № 11. С. 162—174.

27. **Saposhnikov V. V., Dmitriev A., Goessel M., Saposhnikov V. V.** Self-Dual Parity Checking — a New Method for on Line Testing // Proceedings of 14th IEEE VLSI Test Symposium, USA, Princeton. 1996. P. 162—168.

28. **Saposhnikov V. V., Moshanin V., Saposhnikov V. V., Goessel M.** Self-Dual Multi-Output Combinational Circuits with Output Data Compaction // Compendium of Papers IEEE European Test Workshop (ETW'97), Cagliari, Italy, May 28—30, 1997. P. 107—111.

29. **Гессель М., Морозов А. В., Сапожников В. В., Сапожников Вл. В.** Логическое дополнение — новый метод контроля комбинационных схем // Автоматика и телемеханика. 2003. № 1. С. 167—176.

30. **Гессель М., Морозов А. В., Сапожников В. В., Сапожников Вл. В.** Контроль комбинационных схем методом логического дополнения // Автоматика и телемеханика. 2005. № 8. С. 161—172.

31. **Sen S. K.** A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint

Don't Cares // National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE 2010), Sikkim Manipal Institute of Technology, Sikkim, held during 22—24 December, 2010.

32. **Das D. K., Roy S. S., Dmitriev A., Morozov A., Gössel M.** Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proceedings of the 10th International Workshops on Boolean Problems, Freiburg, Germany, September, 2012. P. 33—40.

33. **Sapozhnikov V., Sapozhnikov V., Efanov D.** Concurrent Error Detection of Combinational Circuits by the Method of Boolean Complement on the Base of "2-out-of-4" Code // Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016), Yerevan, Armenia, October 14—17, 2016. P. 126—133. DOI: 10.1109/EWDTS.2016.7807677.

34. **Сапожников В. В., Сапожников Вл. В., Ефанов Д. В.** Построение самопроверяемых структур систем функционального контроля на основе равновесного кода "2 из 4" // Проблемы управления. 2017. № 1. С. 57—64.

35. **Ефанов Д. В., Сапожников В. В., Сапожников Вл. В., Пивоваров Д. В.** Организация полностью самопроверяемой схемы встроенного контроля на основе метода логического дополнения до равновесного кода "2 из 4" // Труды Института системного программирования РАН. 2018. Т. 30, № 2. С. 99—112. DOI: 10.15514/ISPRAS-2018-30(2)-6.

36. **Аксенова Г. П.** Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2 // Автоматика и телемеханика. 1979. № 9. С. 126—135.

37. **Сапожников В. В., Сапожников Вл. В.** Самопроверяемые тестеры для равновесных кодов // Автоматика и телемеханика. 1992. № 3. С. 3—35.

D. V. Efanov, D. Sc., Associate Professor, Head of the Direction of monitoring and diagnosis systems department of "LocoTech-Signal" LLC, Professor of Russian University of Transport, Moscow, Russian Federation

The Conditions of the Formation of the Totally Self-Checking Discrete Devices Based on the Boolean Complement Method up to 2-out-of-4 Constant-Weight Code

The paper shows that following the well-known Boolean complement method, which gives the developer of self-checking discrete devices a wider range of abilities than traditional approaches give, it is necessary to take into account the limitations imposed on the structures of both the object of diagnosis and the check logic block. The paper established the features that are inherent in the Boolean complement method, particularly, the use of a cascade of parallel adders by modulo two. Examples illustrating these features are given. The paper formulates practical recommendations that should be followed in the development of self-checking discrete devices based on the Boolean complement method up to 2-out-of-4 constant-weight code.

Keywords: self-checking discrete devices, self-checking check circuit, Boolean complement method, constant-weight code, "2-out-of-4" code, conditions for fault detecting

DOI: 10.17587/it.25.143-151

References

1. **Chang K., Acharya K., Sinha S., Cline B., Yeric G., Lim S. K.** Impact and Design Guideline of Monolithic 3-D IC at the 7-nm Technology Node, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017, vol. 25, iss. 7, pp. 2118—2129, DOI: 10.1109/TVLSI.2017.2686426.

2. **Yoon J.-S., Jeong E.-Y., Baek C.-K., Kim Y.-R., Hong J.-H., Lee J.-S., Baek R.-H., Jeong Y.-H.** Junction Design Strategy for Si Bulk FinFETs for System-on-Chip Applications Down to the

7-nm Node, *IEEE Electron Device Letters*, 2015, vol. 36, iss. 10, pp. 994—996, DOI: 10.1109/LED.2015.2464706.

3. **Imec and Cadence Tape Out Industry's First 3nm Test Chip**, available at: <https://www.hpcwire.com/off-the-wire/imec-cadence-tape-industrys-first-3nm-test-chip/> (date of access 19.08.2018).

4. **Ubar R., Raik J., Vierhaus H.-T.** Design and Test Technology for Dependable Systems-on-Chip (Premier Reference Source), *Information Science Reference*, Hershey — New York, IGI Global, 2011, 578 p.

5. Kharchenko V., Kondratenko Yu., Kacprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures, *Springer Book series "Studies in Systems, Decision and Control"*, 2017, vol. 74, 305 p.
6. Hahanov V. Cyber Physical Computing for IoT-driven Services, New York, Springer International Publishing AG, 2018, 279 p.
7. Parkhomenko P. P., Sogomonyan E. S. *Osnovy tekhnicheskoy diagnostiki (optimizatsiya algoritmov diagnostirovaniya, apparaturnye sredstva)* (Technical Diagnosis Fundamentals (Diagnostic Algorithm Optimization, Apparatus Means)), Moscow, Energoatomizdat, 1981, 320 p. (in Russian).
8. Drozd A. V., Kharchenko V. S., Antoshchuk S. G., Drozd Yu. V., Drozd M. A., Sulima Yu. Yu. *Rabochee diagnostirovanie bezopasnykh informatsionno-upravlyayustchikh sistem* (Objects and Methods of On-Line Testing for Safe Instrumentation and Control Systems), Kharkov, National Aerospace University "KhAI", 2012, 614 p. (in Russian).
9. Sogomonyan E. S., Slabakov E. V. *Samoproverjaemye ustrojstva i otkazoustojchivyye sistemy* (Self-checking devices and fault-tolerant systems), Radio & Svyaz', Moscow, 208 p. (in Russian).
10. Sapozhnikov V. V., Sapozhnikov VI. V. *Samoproveryaemye diskretnye ustrojstva* (Self-checking discrete devices), St. Petersburg, Energoatomizdat, 1992, 224 p. (in Russian).
11. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. *New Methods of Concurrent Checking: Edition 1*. Dordrecht, Springer Science + Business Media B. V., 2008, 184 p.
12. Mitra S., McCluskey E. J. Which Concurrent Error Detection Scheme to Choose?, *Proceedings of International Test Conference*, 2000, USA, Atlantic City, NJ, 03–05 October 2000, pp. 985–994, DOI: 10.1109/TEST.2000.894311.
13. Elias P. Computation in the Presence of Noise, *IBM Journal of Research & Development*, 1958, vol. 2, iss. 4, pp. 346–353.
14. Freiman C. V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels, *Information and Control*, 1962, vol. 5, iss. 1, pp. 64–71, DOI: 10.1016/S0019-9958(62)90223-1.
15. Hamming R. W. Error Detecting and Correcting Codes, *Bell System Technical Journal*, 1950, 29 (2), pp. 147–160. MR0035935.
16. Berger J. M. A Note on Error Detecting Codes for Asymmetric Channels, *Information and Control*, 1961, vol. 4, iss. 1, pp. 68–73, DOI: 10.1016/S0019-9958(61)80037-5.
17. Piestrak S. J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes, Wrocaw, Oficyna Wydawnicza Politechniki Wrocawskiej, 1995, 111 p.
18. Sapozhnikov V. V., Sapozhnikov VI. V., Efanov D. V., Dmitriev V.V. *Novye struktury sistem funkcional'nogo kontrolya logicheskikh skhem* (New structures of the concurrent error detection systems for logic circuits), *Avtomatika i Telemekhanika*, 2017, no. 2, pp. 127–143 (in Russian).
19. Lala P. K. Self-Checking and Fault-Tolerant Digital Design, San Francisco, Morgan Kaufmann Publishers, 2001, 216 p.
20. Sogomonyan E. S., Gössel M. Self-Testing and Self-Checking Combinational Circuits with Weakly Independent Outputs, *Proceedings of 10th IEEE VLSI Test Symposium*, Atlantic City, 1992, pp. 298–303.
21. Busaba F. Y., Lala P. K. Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors, *Journal of Electronic Testing: Theory and Applications*, 1994, iss. 1, pp. 19–28, DOI: 10.1007/BF00971960.
22. Morosov A., Sapozhnikov V. V., Sapozhnikov VI. V., Goessel M. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs, *VLSI Design*, 1998, vol. 5, iss. 4, pp. 333–345, DOI: 10.1155/1998/20389.
23. Efanov D. V., Sapozhnikov V. V., Sapozhnikov VI. V. *Usloviya obnaruzheniya neispravnosti logicheskogo ehlementa v kombinatsionnom ustrojstve pri funkcional'nom kontrole na osnove koda Bergera* (Conditions for detecting a logical element fault in a combination device under concurrent checking based on Berger's code), *Avtomatika i Telemekhanika*, 2017, iss. 5, pp. 152–165 (in Russian).
24. Efanov D. V., Sapozhnikov V. V., Sapozhnikov VI. V. *Sintez samoproveryaemykh kombinatsionnykh ustrojstv na osnove vydeleniya special'nykh grupp vyhodov* (Synthesis of self-checking combinational devices based on allocating special groups of outputs), *Avtomatika i Telemekhanika*, 2018, no. 9, pp. 79–94 (in Russian).
25. Sapozhnikov V. V., Sapozhnikov VI. V., Efanov D. V. *Klassifikatsiya oshibok v informatsionnykh vektorakh sistemicheskikh kodov* (Errors Classification in Information Vectors of Systematic Codes), *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie*, 2015, vol. 58, no. 5, pp. 333–343 (in Russian).
26. Goessel M., Dmitriev A. V., Sapozhnikov V. V., Sapozhnikov VI. V. *Samotestiruemaya struktura dlya funkcional'nogo obnaruzheniya otkazov v kombinatsionnykh skhemah* (A functional fault-detection self-test combinational circuits), *Avtomatika i Telemekhanika*, 1999, no. 11, pp. 162–174 (in Russian).
27. Sapozhnikov VI. V., Dmitriev A., Goessel M., Sapozhnikov V. V. Self-Dual Parity Checking — a New Method for on Line Testing, *Proceedings of 14th IEEE VLSI Test Symposium*, USA, Princeton, 1996, pp. 162–168.
28. Sapozhnikov VI. V., Moshanin V., Sapozhnikov V. V., Goessel M. Self-Dual Multi-Output Combinational Circuits with Output Data Compaction, *Compendium of Papers IEEE European Test Workshop (ETW'97)*, Cagliari, Italy, May 28–30, 1997, pp. 107–111.
29. Goessel M., Morozov A. V., Sapozhnikov V. V., Sapozhnikov VI. V. *Logicheskoe dopolnenie — novyj metod kontrolya kombinatsionnykh skhem* (Logic complement, a new method of checking the combinational circuits), *Avtomatika i Telemekhanika*, 2003, no. 1, pp. 167–176 (in Russian).
30. Goessel M., Morozov A. V., Sapozhnikov V. V., Sapozhnikov VI. V. *Kontrol' kombinatsionnykh skhem metodom logicheskogo dopolneniya* (Checking combinational circuits by the method of logic complement), *Avtomatika i Telemekhanika*, 2005, no. 8, pp. 161–172 (in Russian).
31. Sen S. K. A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don't Cares, *National Conference on Emerging trends and advances in Electrical Engineering and Renewable Energy (NCEEERE 2010)*, Sikkim Manipal Institute of Technology, Sikkim, held during 22–24 December, 2010.
32. Das D. K., Roy S. S., Dmitriev A., Morozov A., Gössel M. Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes, *Proceedings of the 10th International Workshops on Boolean Problems*, Freiberg, Germany, September, 2012, pp. 33–40.
33. Sapozhnikov V., Sapozhnikov VI., Efanov D. Concurrent Error Detection of Combinational Circuits by the Method of Boolean Complement on the Base of "2-out-of-4" Code, *Proceedings of 14th IEEE East-West Design & Test Symposium (EWDTS'2016)*, Yerevan, Armenia, October 14–17, 2016, pp. 126–133, DOI: 10.1109/EWDTS.2016.7807677.
34. Sapozhnikov V. V., Sapozhnikov VI. V., Efanov D. V. *Postroenie samoproveryaemykh struktur sistem funkcional'nogo kontrolya na osnove ravnovesnogo koda "2 iz 4"* (Design of self-checking concurrent error detection systems based on "2-out-of-4" constant-weight code), *Control Sciences*, 2017, no. 1, pp. 57–64 (in Russian).
35. Efanov D. V., Sapozhnikov V. V., Sapozhnikov VI. V., Pivovarov D. V. *Organizatsiya polnost'yu samoproveryaemoj skhemy vstroennogo kontrolya na osnove metoda logicheskogo dopolneniya do ravnovesnogo koda "2 iz 4"* (The organization of the totally self-checking integrated control circuit based on the boolean complement method up to "2-out-of-4" constant-weight code), *Trudy Instituta sistemnogo programirovaniya RAN*, 2018, vol. 30, no. 2, pp. 99–112, DOI: 10.15514/ISPRAS-2018-30(2)-6 (in Russian).
36. Aksjonova G. P. *Neobhodimye i dostatochnye usloviya postroeniya polnost'yu proveryaemykh skhem svertki po modulyu 2* (Necessary and sufficient conditions for design of completely checkable modulo convolution circuits), *Avtomatika i Telemekhanika*, 1979, iss. 9, pp. 126–135 (in Russian).
37. Sapozhnikov V. V., Sapozhnikov VI. V. *Samoproveryaemye testery dlya ravnovesnykh kodov* (Self-checking checkers for constant-weight codes), *Avtomatika i Telemekhanika*, 1992, no. 3, pp. 3–35 (in Russian).

F. H. Pashayev, D. Sc. in Engineering, Head Researcher, e-mail: pasha.farhad@gmail.com,
Institute of Control Systems of the Azerbaijan National Academy of Sciences,

S. E. Pashayeva, Teacher, e-mail: pasayevasevinc5@gmail.com,
Nakhchivan State University,

H. T. Najafov, Tutor, e-mail: hasan_nacafov@mail.ru,
"Nakhchivan" University,

B. A. Suleymanli, Dissertant, e-mail: bahram.suleymanli@gmail.com
Institute of Control Systems of the Azerbaijan National Academy of Sciences

Operating Algorithms for Folders and Files

The paper presents tools for storing data of large files when creating a database of a search information system, technological corporate computer networks, various project evaluation systems and other systems and algorithms for managing such files. It is noted that these systems and many other systems face similar problems when processing large text files. In search systems developed using a database of guidance documents and various databases, in electronic document management systems, the problem of saving files in separate folders remains relevant. The need to solve similar problems arises during the creation of systems for the electronic evaluation of projects submitted to various contests and competitions. Here, a database of projects submitted for the competition, databases of experts are developed, a knowledge base and other necessary data for evaluating the authors of projects are created, the files associated with these databases are stored, an intelligent analysis of the data is provided. When developing technological corporate computer networks and systems like these networks, the software must support and supply maintenance personnel with the necessary operational and technical documentation. It is necessary to create an interface for the maintenance personnel, which will ensure their quick learning of the system. The software interface should provide access to operational documents and technical documents on basic hardware. The software interface should provide the options of selection, recording and reading of documents, as well as their correction if necessary. For this purpose, the paper covers the advantages of creating same-level folders by the software and storing large files in such folders. To simplify the writing of software codes, such folders and the program folders of the software are to be stored in one folder. Program fragments for accessing such folders and files are given.

Keywords: guidance documents database, technological corporate networks, project evaluation systems, same-level folders, automatic reading of folders

УДК 004.021

DOI: 10.17587/it.25.152-156

Ф. Г. Пашаев, д-р техн. наук., гл. науч. сотр., e-mail: pasha.farhad@gmail.com,
Institute of Control Systems of the Azerbaijan National Academy of Sciences,

С. Е. Пашаева, преподаватель, e-mail: pasayevasevinc5@gmail.com,
Nakhchivan State University,

Г. Т. Наджафов, преподаватель, e-mail: hasan_nacafov@mail.ru, "Nakhchivan" University,

Б. А. Сулейманлы, соискатель, e-mail: bahram.suleymanli@gmail.com,
Institute of Control Systems of the Azerbaijan National Academy of Sciences

Алгоритмы работы с папками и файлами

Представлены инструменты для хранения данных больших файлов при создании базы данных поисковой информационной системы, технологических корпоративных компьютерных сетей, различных систем оценки проектов и других систем и алгоритмов работы с такими файлами. В статье отмечается, что эти системы и многие другие системы сталкиваются с похожими проблемами при работе с большими текстовыми файлами.

В системах поиска, разработанных с использованием базы директивных документов и различных баз данных, в системах управления электронными документами задача сохранения файлов в отдельных папках остается актуальной. Необходимость решения аналогичных проблем возникает во время создания систем электронной оценки проектов, представленных на различные конкурсы. Здесь разрабатываются база данных проектов, представленных на конкурс, базы данных экспертов, создается база знаний и других необходимых данных для оценки авторов проектов, организуется хранение файлов, связанных с этими базами, проводится интеллектуальный анализ представленных данных. При разработке технологических корпоративных компьютерных сетей и систем, подобных этим сетям, программное обеспечение должно поддерживать и обеспечивать обслуживающий персонал необходимыми эксплуатационными и техническими документами. Необходимо создать интерфейс для обслуживающего персонала, который обеспечит ускоренное освоение системы. Интерфейс программного обеспечения должен обеспечивать доступ эксплуатационным документам и техническим документам по основным техническим средствам. Интерфейс программного обеспечения должен обеспечить выбор, запись и чтение, а также корректировку документов при необходимости. Для этой цели даются преимущества создания с помощью программного обеспечения папок одного уровня и хранения больших файлов в таких папках. Для упрощения написания программных кодов предусмотрено хранение таких папок и папки с программным обеспечением в одной папке. В статье приведены программные фрагменты для обращения к таким папкам и файлам.

Ключевые слова: база данных директивных документов, технологические корпоративные сети, системы оценки проектов, папки одинакового уровня, автоматическое чтение папок

1. Introduction

It is known that the detailed information on the documents included in the database of the search information system and the organizations sending these documents are collected in the Directive Document Database [1, 2]. The large files included in the system are written to the Data Base. The detailed information about the documents included in the Data Database of the search information system of the Directive Databases and the organizations sending these documents is kept as follows:

- Registration number of the document — ID;
- The type of the document;
- The sender organization;
- Source of the document;
- The date of entry of the document;
- Executive Officer;
- Brief content of the document. The document's summary or abstract may be in the document itself. It can be created by special software only if not available [3—5];
- The keywords in your mouth. These can also be found in the document itself. If not, can be created by special software;
- The status of the work on the document;
- The deadline for the last application;
- Information about the last applicant;
- Period of execution of the document;
- The address of the physical folder where the original is stored. Special folders are created to keep information accompanying the original documents and documents because of possible being large size. Physical folders are also created to keep up-to-date about the current status of the document and its associated data. The address of these folders and files must be stored in the Database.

Except for the above attributes, other attributes may also be included in the Database, depending on the issues that are necessary. For example:

- List of Executives;
- A list of sections that are intended to perform work on the document, and so on.

As you can see from the list, the system is intended to create custom folders to store large files or files.

The same problem is observed during the creation of many other systems. Examples of such systems are Technology Corporate Computer Networks and various Project Evaluation Systems. The same approach can be used to create information bases for these or other systems. When creating these networks, it is essential that the data storage and retrieval details of the files contained within this network and the addresses of the files stored on the network are stored in the database.

2. Problem statement

The article provides for the creation of such folders, the names, dates of creation, and other attributes of files that are collected in these folders, automatically generating algorithms for reading and arranging arrays within software tools. It is suggested that the fragments of the created algorithms are implemented in DELPHI programming language.

3. Solution of the problem

Keeping files in equal level folders

It is known that the software that generates multiple files, enabling them to store, archive, and maintain a large number of files first of all should solve the problem of storing these files. Analysis of

the software of various information systems shows that one of the ways to solve this problem is to create folders with the same folder as the folder on which the software is stored and to store the files in those folders. For example, we can specify the sequence of the following folders:

```
Y_Q_A
  Q_A_Proq
  Q_A_Dat
```

The Q_A_Proq and Q_A_Dat folders are included in the Y_Q_A folder and are the same level folders. The software can be written to the first folder, while the second folder contains files with "dat" extension, which stores code from analogue dialers. As you can see from the Delphi fragment, the codes to write are written to the AR_BT array and the number of bytes written to the F_NN integer-type variable.

File to be written The Q_A_Dat folder can be written as '.. \ Q_A_Dat \' as it is identical to the application folder. The name of the file has been adapted to the folder name, which has been added to the name of the toolbar and was obtained by the datetimestostr (now) team. Due to the datetime type that can be used in the future, the dots are replaced by dashes. All of these statements were made in the following DELPHI program fragment.

```
str_r := ' ' + datetimestostr(now);
for v_ij: = 1 to length(str_r) do begin
```

```
if str_r[v_ij] = '.' then str_r[v_ij] := '-';
end;
fil_name := '..\Q_A_Dat\' + 'S_DAT04' + str_r + '.dat';
AssignFile(tof, fil_name);
Rewrite(tof, 1); //.. Record size = 1
BlockWrite(tof, AR_BT, F_NN, NumWritten);
CloseFile(tof);
```

The program fragment can be organized as a procedure (subprogram) to be used repeatedly. The name of this subprogram can be written by Write_Proc (Write_Proc).

In this section, str_r and fil_name are string-type variables. The Str_r variable is added to the file name by typing the current date and time. In the fil_name variable, the name of the file is generated according to the above. This name contains the name of the Q_A_Dat folder with the same software as well as the current time and date on the file name. ToF is a variable file type and is used to work with.dat files. The Ar_BT array is an array of bytes of type and the code to be written to the file is written to the array as mentioned above. The F_NN variable is a integer-type variable and saves the number of bytes to be written to the file.

As a result of repeated requests, the files in the folder can be written as in Figure 1.

The advantage of the specified structure is that the Y_Q_A folder can be placed on any logical disk or any folder with folders inside it.

Depending on the issues that the local information system faces and solves the corporate, technological network, other folders of the same level may also be created. For example, the "Intelligent Monitoring System for latent period offailures for Compressor Devices"-has some of the key software technical passports, maintenance papers and other essential software software placed in the "doc" folder [6–10]. Here's also the Operator.doc file, which was created to guide the system operator. A system information screen has been created to access the documentation collected here (Figure 2, see the second side of the cover).

Documents written in the "Doc" folder are shown in Figure 3.

As mentioned above, the information search systems like the directive document database information retrieval system create folders to keep large volumes of documents.

S_DAT04	01.12.2017 0-00-31	01.12.2017 0:00	DAT File	16 KB
S_DAT04	01.12.2017 0-02-14	01.12.2017 0:02	DAT File	16 KB
S_DAT04	01.12.2017 0-03-57	01.12.2017 0:03	DAT File	16 KB
S_DAT04	01.12.2017 0-05-40	01.12.2017 0:05	DAT File	16 KB
S_DAT04	01.12.2017 0-07-24	01.12.2017 0:07	DAT File	16 KB
S_DAT04	01.12.2017 0-09-07	01.12.2017 0:09	DAT File	16 KB
S_DAT04	01.12.2017 0-10-50	01.12.2017 0:10	DAT File	16 KB
S_DAT04	01.12.2017 0-12-33	01.12.2017 0:12	DAT File	16 KB
S_DAT04	01.12.2017 0-14-16	01.12.2017 0:14	DAT File	16 KB
S_DAT04	01.12.2017 0-15-59	01.12.2017 0:16	DAT File	16 KB
S_DAT04	01.12.2017 0-17-42	01.12.2017 0:17	DAT File	16 KB
S_DAT04	01.12.2017 0-19-25	01.12.2017 0:19	DAT File	16 KB

Fig. 1. The sequence of writing files in the Q_A_Dat folder

ac297e_atex_1	13.04.2006 16:23	Adobe Acrobat D...	359 KB
algor01.doc	01.03.2007 15:55	Документ Micros...	298 KB
AS-062	13.04.2006 17:23	Adobe Acrobat D...	200 KB
ASA 062	14.04.2006 15:09	Adobe Acrobat D...	178 KB
Bp188212	22.10.2002 10:18	Adobe Acrobat D...	424 KB
Cpu188-5	29.11.2002 19:01	Adobe Acrobat D...	541 KB
liter01.doc	06.03.2007 11:59	Документ Micros...	64 KB
mc35it_hd_v0200	17.12.2006 11:16	Adobe Acrobat D...	3 590 KB
Operator	27.10.2012 13:18	Документ Micros...	134 KB

Fig. 3. Doc folder of Intelligent Monitoring System for Compressor Devices

Commands for reading and editing text files

To ensure that the files stored in the system's Doc folder or other folders are accessed by software, reading these files or performing corrections in text files the following types of commands can be used to:

— The command to read text files and make corrections if necessary:

```
begin
ShellExecute(handle,nil,'C:\mk301_2_b\doc\
Operator.doc',nil,nil,sw_shownormal);
end;
```

This command provides open and read the Operator.doc file that is written in the "C: \ mk301_2_b \ doc" folder and guides the system operator's operation. If there is a need to make any additions to this document during the exploitation process, the document can be routinely corrected as it is opened as a word document. Attachments can be stored in the document.

— The command to read the system's algorithms and correct them if necessary:

```
begin
ShellExecute(handle,nil,'C:\mk301_2_b\doc\al
gor01.doc',nil,nil,sw_shownormal);
end;
```

This command will open and read the algor01.doc file, which is written in the "C: \ mk301_2_b \ doc" folder and contains the system's algorithms.

— The command that provides reading of passport of the vibrator converter of Intelligent Monitoring System of hidden periods of accidents for Compressor Devices:

```
begin
ShellExecute(handle,nil,'C:\mk301_2_b\doc\
AS-062.pdf',nil,nil,sw_shownormal);
end;
```

This command will read the technical and operational characteristics of the AS-062 vibrator converter which are written in the "C: \ mk301_2_b \ doc" folder and produced by Bruel and Kjaer. This information is provided in the AS-062.pdf file.

The Intellectual Monitoring System of hidden periods of accidents for Compressor Devices has been designed to monitor the changes in the technical condition of compressor and other strategic facilities, to assess the seismic stability of these facilities. The signals from the AS-062 vibrator converter are stored as 12-bit codes using the Analog Digital Converter (ARC) of MICRO PC type controls. The signal's 4-second codes are written as dat files in the same BASA_S folder as the software. If the signal read frequency is 2 Khs, then 8000 bytes will be recorded per file.

Algorithms for working with Dat files

The following two issues should be resolved in relation to this base. The first issue is to convert the data in a single file into a type of real number, and the second issue is to read the names and dates of the files in the directory.

The solution of the first issue.

— Open the selected file type and write it to AR_BT. This array must have at least 16,000 bytes. The first byte of each pair is the large byte, the second is the small byte. If the code is a positive number, the big byte will be less than 8, meaning the 12th bit will be zero. Otherwise, the large byte will be greater than 8, meaning that the 12th bit will be 1;

— We can store real numbers that can be retrieved after transformation in big_r real numbers. Write a large part of each pair of AR_BT in byte_text. Therefore, if big_k < 8 is big_r number

$$\text{big_r}[i] = \text{AR_BT}[i*2-1]*256 + \text{AR_BT}[i*2]$$

Calculated like that. Otherwise

$$\text{big_r}[i] = -1*((255-(\text{AR_BT}[i*2-1]))*256 + \text{not}(\text{AR_BT}[i*2]))$$

can be calculated as.

As a result, the big_r real numbers massive can be used for different purposes, including analysis and graphic design.

The solution of the second issue.

To save the names and creation dates of the files in the folderFp:

TSearchRec;

FAge: Integer;

FileParam: TDateTime;

A structured record should be created. At this time FindFirst('.\BASA_S*.dat', faAnyFile, Fp) with the command the necessary information about the first file is written to the Fp record. FindNext(Fp) and the command writes the files in the folder to the Fp record. Both two command Fp.Name file's name in the parameter, FileAge('.\BASA_S\' + filename); the date of the file being written in the parameter, by FileParam: = FileDateToDateTime(filedate), DateToStr(FileParam), TimeToStr(Fileparam) parameters give accordingly the date and time of writing. By using these commands and parameters properly, the names and record dates of all files in the BASA_S folder can be written to the arrays. As a result, it will be possible to carry out the intended operations:

```
procedure TForm1.Button2Click(Sender: TObject);
var
Fp:TSearchRec;
FAge: Integer;
FileParam: TDateTime;
label m_end_7;
begin
```

```

    f_c := 0;
    memol.Clear;
    ifFindFirst('..\BASA_S\*.dat', faAnyFile, Fp) = 0
    then
    begin
        f_c := f_c + 1;

        mas_string[f_c] := Fp.Name;
        fileDate := FileAge('..\BASA_S\' + mas_
string[f_c]);
        FileParam := FileDateToDateTime(filedate);
        date_01 := DateToStr(FileParam);
        time_01 := TimeToStr(Fileparam);
        X_X := STRTODATE(DATE_01) +
        STRTOTIME(time_01);
        if fileDate > -1 then
        x_m[f_c] := X_X;
        end;

    whileFindNext(Fp) = 0 do
    begin
        f_c := f_c + 1;
        mas_string[f_c] := Fp.Name;
        fileDate := FileAge('..\BASA_S\' + mas_
string[f_c]);
        FileParam := FileDateToDateTime(filedate);
        date_01 := DateToStr(FileParam);
        time_01 := TimeToStr(Fileparam);
        X_X := STRTODATE(DATE_01) +
        STRTOTIME(time_01);
        if fileDate > -1 then
        x_m[f_c] := X_X;
        end;
    FindClose(Fp);

```

Result

In the article saving methods information about large files, algorithms of working with such files while creating Data Bases of various systems. It has been shown that information about such files, including files in which folders are stored, is written in the

Database. The advantages of keeping files in equal folders with software are provided. It was noted that, information about software tools used in the creation of system, exploitation documents of system can be kept in such folders and this can increase exploitation quality of the system. The article recommends creating software screens to work with such files. The software contains algorithms for working with folders and folders for files stored in such folders.

Methods, algorithms and software fragments can be used to create software for different systems.

Reference

1. **Pashayeva S. E.** Creating a Directive Document Search System database, Nakhchivan State University, *Scientific Works*, 2016, no. 3 (77), pp. 59–67.
2. **Pashaeva E.** Local search documents in corporate life, *VII All-Ukrainian Scientific-Practical Conference, Computer Science and Systems Sciences*, Poltava, 10–12 March 2016, pp. 223–225.
3. **Aliguliyev R., Hacirahimova M. Sh.** Optimization model for auto-synthesis of texts, *Problems of Information Technologies*, 2015, no. 2, pp. 96–102.
4. **Alguliev R. M., Aliguliyev R. M., Hajirahimova M. S.** Quadratic Boolean programming model and binary differential evolution algorithm for text summarization, *Problems of Information Technologies*, 2012, no. 2 (6), pp. 20–29.
5. **Pashayeva S. E.** Methods for working with text files, *News of the Azerbaijan National Academy of Sciences, Information and Management Problems*, 2016, vol. XXXVI, no. 6, pp. 104–111.
6. **Guluev G. A.** Robust system of identification of the vibratory condition and forecasting of emergency conditions of compressor units, *SOCAR Proceedings*, Jun 2012, no. 3, pp. 65–70.
7. **Aliev T. A., Musaeva N. F., Guluyev G. A., Sattarova U. E.** Noise Indication of Change of Dynamical Condition of Production Objects, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2011, no. 8, pp. 2–5.
8. **Aliev T. A., Guluyev G. A., Pashayev F. H., Sadygov A. B.** Noise monitoring technology for objects in transition to the emergency state, *Mechanical Systems and Signal Processing*, February 2012, vol. 27, pp. 755–762.
9. **Aliev T. A., Guluyev G. A., Pashayev F. H., Sadygov A. B.** Algorithms for determining the coefficient of correlation and cross-correlation function between a useful signal and noise of noisy technological parameters, *Cybernetics and System Analysis*, May 2011, vol. 47, no. 3, pp. 481–489.
10. **Guluyev G. A.** Application of the neural network for identification of the vibration condition of the compressor unit, *Azerbaijan Oil Industry*, 2013, no. 1, pp. 40–45.

Е. А. Юлюгин, инженер по разработке программного обеспечения, e-mail: evgeny.yulyugin@intel.com, Intel Sweden AB

Корректное и быстрое исполнение отдельных инструкций архитектуры Intel® 64 в виртуальном окружении

Архитектура Intel®64 постоянно развивается. Новые процессоры обычно сохраняют обратную совместимость и включают всю программно доступную функциональность предыдущих поколений. Было обнаружено, что поведение некоторых инструкций может нарушить совместимость при исполнении внутри виртуальной машины. В работе описываются разработанные методы и инструменты, позволяющие создавать быстрое и корректное виртуальное окружение, использующее аппаратные средства виртуализации.

Ключевые слова: Wind River Simics®, виртуализация, моделирование, симуляция, x86, Intel VT-x, гипервизор, Izcnt

Введение

В настоящее время программное моделирование является одним из важнейших элементов процесса создания вычислительных систем. Моделирование позволяет начать разработку и отладку программного стека задолго до появления аппаратного решения, тем самым минимизируя временной интервал между выходом аппаратной платформы и программного обеспечения.

Использование программных симуляторов остается актуальным даже после выпуска новой аппаратной платформы, так как предоставляет возможности для отладки, отсутствующие в физических системах [1]. Для того чтобы быть применимой для загрузки и отладки полного программного стека, модель должна демонстрировать высокую скорость работы, сохраняя при этом функциональную корректность. Увеличение производительности возможно с помощью прямого исполнения кода целевой системы на хозяйском оборудовании, если моделируемая и хозяйская архитектуры частично совпадают.

В работе описываются современные подходы к функциональному моделированию вычислительных систем. Акцент делается на изучении технологии прямого исполнения, позволяющей достичь скорость моделирования, близкую к скорости работы аппаратуры. В рамках этой работы был обнаружен и описан новый класс инструкций, расширяющий классификацию Попека и Голдберга [2] и делающий существующий способ организации прямого исполнения, основанный на технологии "trap and emulate", неприменимым для ряда важных сценариев. Цель исследования заключается в разработ-

ке нового метода, позволяющего организовать эффективное и корректное прямое исполнение кода, содержащего инструкции обнаруженного класса. Работа была проведена на основе функционального симулятора Wind River* Simics* [1].

1. Обзор литературы

Существующие подходы к моделированию процессоров можно разделить на три типа: программное моделирование, аппаратная виртуализация и их комбинация.

1.1. Программное моделирование

Программное моделирование является гибким решением, основанным на технологиях интерпретации и двоичной трансляции [3]:

- *интерпретация* — наиболее универсальный и аппаратно независимый, однако самый медленный подход к моделированию процессоров. Для увеличения производительности Vochs [4] и другие симуляторы, основанные исключительно на технологии интерпретации, обычно используют только переносимые и архитектурно независимые оптимизации, такие как трассы инструкций и кэширование результатов трансляции адресов;
- *двоичная трансляция* (англ. binary translation) позволяет наиболее эффективно моделировать простые инструкции, сокращая накладные расходы, связанные с декодированием и выборкой следующей инструкции для исполнения.

Многие симуляторы, такие как Zsim [5], совмещают использование технологий интерпре-

тации и двоичной трансляции. Данные подходы являются наиболее гибкими, так как не полагаются на аппаратные возможности конкретной платформы. Однако часто они не обеспечивают необходимую производительность.

1.2. Аппаратная виртуализация

Возможность создания эффективного монитора виртуальных машин была теоретически обоснована в 1974 г. Ж. Попеком и Р. Голдбергом [2]. Для доказательства они использовали упрощенную модель вычислительной системы, состоящую из центрального процессора и линейной сегментированной оперативной памяти. Периферийные устройства в рамках предложенной модели опускаются; центральный процессор поддерживает два режима работы: режим пользователя (англ. user mode), исполняющий прикладные программы, и режим супервизора (англ. supervisor mode), в котором исполняется код операционной системы. Память поддерживает механизм сегментации, который используется для организации виртуальной памяти.

Считается, что для некоторых входных условий инструкция рассматриваемого процессора вызывает ловушку (англ. trap), если результатом ее исполнения является сохранение состояния процессора в одну ячейку памяти и загрузка нового состояния из другой. Иначе говоря, ловушка позволяет сохранить состояние программы перед исполнением инструкции и передать управление обработчику, обычно работающему в режиме супервизора. Управление может быть возвращено обратно с помощью восстановления сохраненного состояния. Ловушки бывают двух типов: 1) ловушки потока управления — вызванные попыткой изменить состояние процессора; 2) ловушки защиты памяти — вызванные обращением к содержимому памяти, выходящему за пределы заданного сегмента. Данные типы не взаимоисключающие, т. е. результатом исполнения инструкции могут быть одновременно ловушки защиты памяти и потока управления.

Инструкции рассматриваемого процессора можно разделить на три класса [3]:

- привилегированные (англ. privileged) — инструкции, которые безусловно вызывают ловушку при исполнении в режиме пользователя;
- служебные (англ. sensitive). Данный класс состоит из двух подклассов: 1) инструкции, которые могут менять режим работы процессора или размер и положение доступного сегмента памяти; 2) инструкции, поведение которых зависит от режима процессора или конфигурации используемого сегмента памяти;

- безвредные (англ. innocuous) — самый широкий класс, включающий в себя все инструкции, не являющиеся служебными. Поведение инструкций этого класса не зависит ни от режима работы процессора, ни от настроек сегмента памяти.

Достаточное условие построения монитора виртуальных машин (ВМ): множество служебных инструкций является подмножеством привилегированных инструкций.

Изначально архитектура Intel® 64 не выполняла достаточные условия построения монитора ВМ, что было исправлено с появлением расширения Intel VT-x [6]. Монитор виртуальных машин исполняется в режиме VMX root, в котором доступны команды, управляющие поведением виртуальных машин, работающих в режиме VMX non-root. Исполнение привилегированных или служебных инструкций в режиме VMX non-root вызывает VM-exit — передачу управления монитору виртуальных машин с последующим моделированием данной инструкции и возвращением управления гостевому программному обеспечению.

Аппаратные виртуальные машины программно моделируют только привилегированные и служебные инструкции, исполняя остальные операции на хозяйском процессоре (рис. 1). Данная технология называется "trap and emulate" [7] и используется во многих гипервизорах, таких как Oracle VirtualBox [8], Microsoft* Hyper-V [9], KVM [10], Vmware* ESXi [11].

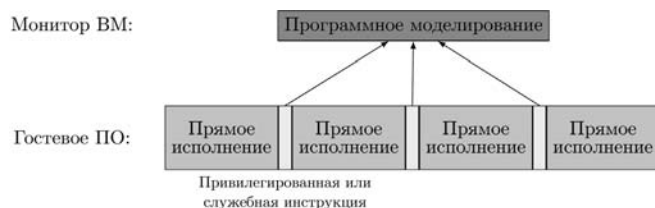


Рис. 1. Технология "trap and emulate", используемая при создании аппаратных систем виртуализации

1.3. Комбинированный подход

Программное моделирование позволяет создавать гибкие модели, предоставляя возможности для изучения недоступных в аппаратуре технологий. Однако производительность таких виртуальных машин остается достаточно низкой. Аппаратная виртуализация, напротив, позволяет достичь высокой скорости моделирования, не давая возможности использовать отсутствующие в аппаратуре инструкции.

Simics комбинирует технологии программного и аппаратного моделирования. Интерпретация и двоичная трансляция используется для

моделирования привилегированных, служебных и отсутствующих в аппаратуре инструкций. Режим прямого исполнения VMP [12], основанный на технологии Intel VT-x, позволяет обеспечить максимальную производительность при моделировании процессоров, основанных на архитектуре Intel 64. Технологии программного моделирования также используются в случаях, когда прямое исполнение оказывается неэффективным [13].

Simics — функционально точный полноплатформенный симулятор, позволяющий создавать высокопроизводительное виртуальное окружение, способное моделировать поведение любых вычислительных схем, начиная с одной платы и заканчивая многопроцессорными и многомашинными системами. Разработанные с помощью Simics модели позволяют исследовать, тестировать, а также заниматься отладкой немодифицированного программного обеспечения. Созданное виртуальное окружение способно работать как с низкоуровневым ПО, таким как BIOS и операционные системы, так и с любыми приложениями, оперирующими на уровне пользователя.

2. Одна кодировка — разные инструкции

Теория, разработанная Попеком и Голдбергом в 1974 г., все еще актуальна для задачи создания виртуального процессора, имеющего в точности такую же архитектуру, что и хозяйский. Однако существует ряд задач, для которых архитектура виртуализуемого процессора не полностью совпадает с хозяйской:

- динамическая миграция (англ. live migration) — перенос виртуальной машины с одной физической системы на другую без прекращения ее работы. При этом архитектура хозяйских систем, между которыми происходит миграция, может отличаться. Например, гипервизоры, такие как Microsoft* Hyper-V и VMware* ESXi, реализуют динамическую миграцию между системами с процессорами Intel и AMD* разных поколений, т. е. поддерживающих разные наборы инструкций;
- программное моделирование, основанное на комбинированном подходе, подразумевает, что архитектуры гостевого и хозяйского процессоров могут отличаться. При этом нередко гостевая система содержит функциональность, недоступную на данный момент в реальной аппаратуре.

В рамках выполненного исследования был обнаружен новый класс команд процессора, расширяющий теорию Попека и Голдберга.

На первый взгляд, исследуемые инструкции могут быть отнесены к классу безвредных, но результат их исполнения может отличаться на разных поколениях процессоров одной и той же архитектуры. Аналогично привилегированным данные инструкции угрожают корректному исполнению в виртуальном окружении, однако в отличие от первых они не могут быть перехвачены с помощью аппаратуры. Существующие механизмы виртуализации не могут быть использованы для обнаружения и предотвращения некорректного исполнения инструкций, входящих в обнаруженный класс, так как они не вызывают исключения ловушки.

2.1. Переиспользование кодировок

В 2007 г. компания AMD* выпустила процессор с кодовым именем Barcelona [14], в котором комбинация REP префикса (байт 0xF3) и кодировки инструкции BSR была переиспользована в качестве кодировки для новой инструкции — LZCNT. Такое переиспользование привело к тому, что одна и та же последовательность байтов стала соответствовать разным инструкциям в зависимости от поколения процессора. Данные инструкции имеют разную семантику, так что попытка моделирования новой инструкции с помощью прямого исполнения на старой аппаратуре приводит к неправильному результату, поскольку заданная последовательность байтов интерпретируется аппаратурой как другая инструкция. Инструкция LZCNT была добавлена в процессоры Intel начиная с Intel® Core™ четвертого поколения (кодовое имя Haswell) в 2013 г.

Единственный публично известный способ для уменьшения эффектов обнаруженной проблемы — ограничение возможностей виртуальной машины до безопасного набора команд [9], выпущенного в 2010 г. Ограничение возможностей виртуального процессора реализуется с помощью инструкции идентификации процессора CPUID. Например, Microsoft Hyper-V всегда сообщает отсутствие поддержки инструкции LZCNT, если включен режим динамической миграции. Такое ограничение устанавливает излишне жесткие границы на производительность системы, так как только более старые инструкции могут быть использованы. Более того, данный подход неприменим для задачи программного моделирования, поскольку возможности, которые должна реализовывать виртуальная машина, строго зафиксированы в этом случае.

Инструкция CPUID может быть использована для определения наличия поддержки новой инструкции заданным процессором.

Тестовая программа, написанная на языке C (листинг 1), проверяет согласованность CPUID информации и результата исполнения последовательности байтов, соответствующей инструкции LZCNT. Программа выполняет инструкцию CPUID для того, чтобы определить наличие поддержки LZCNT, а затем исполняет последовательность байтов 0xF30FBDD8, которая соответствует либо инструкции "lzcnc ebx, eax", либо "bsr ebx, eax" в зависимости от поколения процессора, на котором эта программа будет выполняться. Результат исполнения, записанный в регистр EBX, затем используется для того, чтобы определить, какая инструкция была исполнена. Ожидаемым результатом исполнения программы считается печать сообщения "LZCNT is supported. 0xF30FBD corresponds to LZCNT." или "LZCNT is not supported. 0xF30FBD corresponds to BSR."

Ограничение возможностей виртуального процессора на самом деле не позволяет достичь корректного поведения виртуальной машины, и запуск тестовой программы покажет различный результат в зависимости от хозяйского оборудования, на котором будет исполняться виртуальная машина. Исполнение данной тестовой программы внутри виртуальной машины, работающей под управлением Microsoft* Hyper-V в режиме динамической миграции, при использовании хозяйского процессора Intel Core i7-2600, не поддерживающего инструкцию LZCNT, показало правильный результат: "LZCNT is not supported. 0xF30FBD corresponds to BSR.". Результат исполнения той же конфигурации с использованием хозяйского процессора Intel Core i5-4300U, поддерживающего LZCNT, оказался неверным: "LZCNT is not supported. 0xF30FBD corresponds to LZCNT."

Обратная ситуация происходит, когда программные симуляторы, такие как Simics, используют аппаратную поддержку виртуализа-

ции для моделирования гостевого кода через прямое исполнение. Модель более нового процессора сообщает о поддержке LZCNT, тогда как более старый хозяйский процессор, на котором будет происходить прямое исполнение, эту инструкцию не реализует. Кодировка, соответствующая новой инструкции, будет исполнена как старая, что приведет к неверному результату исполнения тестовой программы: "LZCNT is supported. 0xF30FBD corresponds to BSR."

Одновременно с инструкцией LZCNT в процессорах Intel Core четвертого поколения была добавлена инструкция TZCNT, входящая в состав расширения архитектуры команд BMI. Кодировка TZCNT является комбинацией REP префикса и кодировки инструкции BSF, которая не вызывает исключение ловушки при исполнении на процессорах предыдущих поколений. Операции, выполняемые инструкциями BSF и TZCNT, не взаимозаменяемы и поэтому также угрожают корректности моделирования при использовании прямого исполнения.

2.2. Инструкции сохранения и восстановления контекста XSAVE

Семейство инструкций XSAVE было впервые добавлено в процессорах Intel Core второго поколения одновременно с набором векторных инструкций Intel Advanced Vector Extension (Intel AVX). Расширение XSAVE реализует сохранение и восстановление компонент состояния (регистров или частей регистров) процессора в память. Данное расширение состоит из восьми инструкций:

- XSAVE, XSAVEC, XSAVEOPT, XSAVES — инструкции, сохраняющие состояние регистров процессора в память;
- XRSTOR, XRSTORS — инструкции, восстанавливающие состояние процессора из памяти;
- XGETBV, XSETBV — инструкции, оперирующие над расширенными регистрами управления (англ. extended control register, XCR).

Инструкции сохранения и восстановления состояния процессора принимают значение контрольного регистра XCR0 неявным аргументом. Каждый бит регистра XCR0 соответствует определенному компоненту состояния процессора. Различные поколения процессоров могут поддерживать разное число компонент состояния. Наборы команд, требующие использования XSAVE, называются XSAVE-enabled расширениями. Intel AVX, Intel AVX2, Intel AVX-512 и Intel Memory Protection Extension (Intel MPX) являются XSAVE-enabled-расширениями набора команд. Данный список

Листинг 1. Проверка корректности отображения функциональности LZCNT

```
#include <stdio.h>
#include <stdint.h>
#include <stdbool.h>

int main () {
    uint32_t ecx, cpuid_leaf = 0x80000001;
    asm volatile("cpuid" : "=c"(ecx) : "a"(cpuid_leaf) : "ebx", "edx");
    bool has_lzcnt = ecx & (1 << 5);
    printf("LZCNT is%s supported.", has_lzcnt ? "" : " not");

    uint32_t out, in = 0x11aa00bb;
    // "lzcnc ebx, eax" or "bsr ebx, eax" depending on CPU generation
    asm volatile(".byte 0xf3, 0x0f, 0xbd, 0xd8" : "=b"(out) : "a"(in));
    bool lzcnt = out == 0x3;
    printf(" 0xF30FBD corresponds to %s.\n", lzcnt ? "LZCNT" : "BSR");

    return 0;
}
```

может быть расширен в дальнейшем при введении новых расширений для архитектуры Intel 64.

Моделирование инструкций расширения XSAVE через прямое исполнение не вызывает никаких проблем, если гостевое программное обеспечение использует компоненты состояния, которые поддерживаются хозяйским оборудованием. Если же гостевое ПО настроило регистр XCR0 так, что активен хотя бы один компонент, не поддерживаемый хозяйской аппаратурой, то прямое исполнение инструкций расширения XSAVE должно быть запрещено, так как хозяйские инструкции не могут корректно обработать отсутствующие компоненты состояния. Исполнение инструкций XSAVES и XRSTORS в режиме VMX non-root зависит от настроек контрольной структуры VMCS и может быть перехвачено. Инструкция XSETBV в режиме VMX non-root безусловно вызывает передачу управления монитору виртуальных машин. Поведение остальных инструкций расширения XSAVE в режиме VMX non-root никак не изменено, однако исполнение всех инструкций расширения может быть запрещено через контрольный регистр CR4.

В отличие от инструкций LZCNT и TZCNT исполнение инструкций, входящих в расширение XSAVE, может быть отключено при прямом исполнении и тем самым перехвачено. Однако вместе с инструкциями XSAVE также будет запрещено исполнение всех XSAVE-enabled-инструкций. В момент исследования XSAVE-enabled-расширения включали в себя несколько сотен инструкций, которые были недоступны для прямого исполнения из-за расширения XSAVE, если гостевое ПО включило хотя бы одну компоненту состояния, не поддерживаемую хозяйским оборудованием.

2.3. Известные проявления проблемы

Было обнаружено неправильное поведение некоторых операционных систем при исполнении в виртуальной среде. Например, ОС Google* Android* версии 4.4, основанная на ядре Linux* 3.10, использует инструкцию LZCNT независимо от идентификационной информации, доступной через CPUID. Ограничение возможностей виртуального процессора, реализованное с помощью инструкции CPUID, не даст ожидаемого эффекта в данном случае. Загрузка операционной системы Android в виртуальной среде, использующей технологию прямого исполнения, на хозяйском оборудовании, не поддерживающем инструкцию LZCNT, завершится ошибкой, так как вместо инструкции LZCNT будет исполнена инструкция BSR.

Операционные системы Linux и Android используют инструкцию TZCNT. На старом оборудовании, не поддерживающем расширение BMI, кодировка этой инструкции будет исполнена как инструкция BSF. Семантика этих инструкций очень близка, но не взаимозаменяема ввиду нескольких ключевых особенностей — результатом исполнения TZCNT при нулевом входном значении является размер операнда, тогда как результат исполнения инструкции BSF не определен в этом случае. Инструкция TZCNT также обновляет флаг переноса (англ. carry flag), тогда как значение этого флага после исполнения BSF не определено. Компилятор GCC может генерировать инструкцию TZCNT даже если целевая архитектура не поддерживает ее [15], таким образом, делая эту инструкцию достаточно популярной в современном программном обеспечении. Использование BSF вместо TZCNT в виртуальном окружении обычно не является опасным вследствие незначительной разницы в их поведении, однако подобная замена все же может привести к некорректному исполнению.

Инструкции семейства XSAVE используются всеми известными современными операционными системами, работающими на архитектуре Intel 64, для переключения контекста (англ. context switch) и инициализации процессов. Также большинство гипервизоров, включая Simics, используют XSAVE-инструкции для управления состоянием виртуальных машин.

3. Обеспечение корректного моделирования

Решение задачи обеспечения корректного моделирования исследуемых инструкций заключается в том, чтобы предотвратить прямое исполнение этих инструкций в виртуальном окружении — они должны быть промоделированы программно. Выявление подобных инструкций должно быть сделано заранее, так как исправить ошибку после исполнения невозможно.

В аппаратуре код виртуальных машин организован в страницы — выровненные блоки памяти фиксированного размера (типичный размер 4 Кбайт). С каждой страницей связан набор флагов, определяющих права доступа к соответствующей странице: страница может быть доступна для записи, чтения, исполнения или для комбинации этих прав.

Перед тем как разрешить исполнение кода, находящегося на какой-либо гостевой странице, она должна быть просканирована на наличие шаблонов, описывающих проблемные

инструкции. Если совпадения не найдены, то данная страница помечается безопасной. В дальнейшем исполнение с этой страницы происходит напрямую с помощью аппаратуры. Страница должна быть защищена от записи, так как ее содержимое должно быть пересканировано после каждого изменения.

Задача определения наличия заданных инструкций в памяти не является тривиальной ввиду особенностей архитектуры Intel 64 — переменной длины команды с неопределенными границами и эффектов, возникающих на границах страниц памяти. Исполнение может происходить по любому смещению внутри страницы, кодировка инструкции может начинаться на одной странице, а заканчиваться на следующей. Полное декодирование является медленным и требует знания всех точек входа на заданной странице, что требует полного анализа потока управления. В данной работе предлагается более простое и быстрое решение — частичное декодирование с поиском предопределенного набора битовых последовательностей. Например, для того чтобы убедиться, что на заданной странице памяти нет инструкции LZCNT, сканер шаблонов должен убедиться, что на этой странице не встречаются последовательности байтов [0xF3, 0x0F, 0xBD] и [0xF3, REX.W, 0x0F, 0xBD], где REX.W имеет значение в интервале 0x40-0x4F.

В результате сканирования страницы гостевого кода делятся на три типа (рис. 2):

- безопасные — просканированные страницы, не содержащие проблемных инструкций;



Рис. 2. Классификация страниц памяти в зависимости от наличия проблемных инструкций



Рис. 3. Цикл жизни страницы

Листинг 2. Последовательность инструкций, интерпретируемая как LZCNT

```
0fbaf30f    btr ebx, 0xf
bdddcbbaa    mov ebp, 0xaabccdd
```

- опасные — просканированные страницы, вероятно содержащие проблемные инструкции, хотя это не может быть доказано без изучения истории исполнения кода, находящегося на этой странице;
- новые страницы, еще не прошедшие сканирование.

Следует отметить, что описанный механизм сканирования позволяет гарантировать отсутствие опасных инструкций на заданной странице, но не позволяет с точностью определить наличие данных инструкций. Точное определение наличия таких инструкций на заданной странице требует анализа потока управления или изучения истории исполнения. Листинг 2 показывает пример последовательности инструкций, которая будет определена сканером шаблонов как LZCNT, так как кодировка, соответствующая последовательности этих инструкций, содержит непрерывный набор байтов 0xf30fbd.

Содержимое страницы должно быть пересканировано, если была проведена запись в эту страницу, так как инструкции, содержащиеся на этой странице, могли быть изменены. Цикл жизни страницы с кодом изображен на рис. 3. Начальное состояние — невыделенная страница.

4. Измерения

Рис. 4 (см. вторую сторону обложки) показывает время загрузки основных операционных систем, измеренное при использовании трех различных режимов моделирования:

- оригинальный VMP — немодифицированный алгоритм прямого исполнения, основанный на технологии аппаратной виртуализации Intel VT-x;
- адаптированный VMP — алгоритм прямого исполнения, реализующий изменения, описанные в разделе 3;
- выключенный VMP — программное моделирование, основанное на технологиях интерпретации и двоичной трансляции.

Эксперименты проводили на рабочей станции с одним центральным процессором Intel® Xeon® E3-1270 v5 3.60 ГГц (микроархитектура Skylake) и 64 Гбайт ОЗУ. Использовалась 64-битная операционная система SUSE Linux Enterprise Server 11. Измерения проводили на симуляторе Wind River Simics версии 5. Для оценки влияния нового алгоритма на производительность симулятор сканировал и исключал из прямого исполнения страницы гостевой памяти, содержащие шаблоны инструкций LZCNT и TZCNT, несмотря на то что в этом не было необходимости, так как и моделируе-

мая, и хозяйская системы поддерживали эти инструкции.

Измерения показали, что новый алгоритм в среднем в 1,18 раз медленнее, чем традиционный (но небезопасный) подход к организации прямого исполнения. Разработанная технология также оказалась в 3,32 раза быстрее, чем моделирование, основанное только на интерпретации и двоичной трансляции. Сканирование гостевой памяти не оказывает видимого влияния на производительность симулятора, что подтверждается на примере сценария загрузки операционной системы FreeBSD 10.3, время работы которого не изменилось. При этом все страницы памяти, с которых происходило исполнение, были изучены на предмет наличия шаблонов, соответствующих небезопасным инструкциям. Исследование показало, что все исполняемые страницы были безопасными.

4.1. Загрузка операционной системы Fedora 23

Загрузка операционной системы Fedora 23 (ядро 4.2.3) продемонстрировала наиболее существенное замедление при использовании модифицированного VMP-алгоритма — в 1,7 раз медленнее по сравнению с оригинальным механизмом прямого исполнения VMP. Моделируемая система состояла из двух процессоров Intel Xeon (микроархитектура Skylake) и имела 4 Гбайт оперативной памяти. 66 из 8706 просканированных страниц были помечены как небезопасные по причине наличия шаблона, соответствующего инструкции TZCNT. Около 9,7 млн инструкций TZCNT были промоделированы во время исполнения сценария, состоящего из 40,1 млрд инструкций. Все исполненные TZCNT инструкции находились на 64 различных страницах памяти. Шаблон, соответствующий инструкции LZCNT, был обнаружен только на одной странице. Ни одной инструкции LZCNT не было исполнено во время загрузки сценария.

Эти измерения показывают, что небезопасные инструкции нечасто используются современным программным обеспечением — 0,02 % от общего числа исполненных инструкций во время загрузки Fedora 23. Только 0,77 % страниц памяти были помечены как небезопасные и исключены из прямого исполнения. 99,5 % этих страниц на самом деле содержали инструкцию TZCNT. На первый взгляд, число небезопасных для прямого исполнения страниц может показаться незначительным, но блокирование этих страниц привело к существенному увеличению доли гостевого кода, исполненного с использованием технологий интерпретации и

двоичной трансляции, — с 1,3 до 9,7 %. Также блокировка прямого исполнения с небезопасных страниц привела к значительному увеличению числа дорогостоящих переходов между режимами программного моделирования и прямого исполнения — 3,7 млн переключений для оригинального VMP алгоритма и 4,3 млн переключений для модифицированного VMP. Вместе с существенным увеличением числа переключений между режимами моделирования и рост доли программного моделирования привели к увеличению времени работы сценария.

Заключение

Исследование показало, что существующая технология аппаратной виртуализации не может быть использована для моделирования будущих поколений процессоров той же архитектуры. Для обеспечения корректного и быстрого исполнения в виртуальной машине необходима комбинация аппаратной виртуализации и описанной программной технологии.

Данное исследование было начато во время работы над моделью процессора Intel Core четвертого поколения (кодовое имя Haswell), который расширил набор команд Intel 64 инструкциями LZCNT и TZCNT. Описанная технология позволила разработать быструю и корректную модель, которая использовалась для разработки программного обеспечения до появления процессора на рынке. Новая технология моделирования позволила загрузить и отладить основные операционные системы и гипервизоры, используя модель процессора Intel Core четвертого поколения, исполняющуюся на процессоре второго поколения, что было бы невозможно при использовании традиционной технологии аппаратной виртуализации.

Проведенное исследование показывает, что проблемные инструкции встречаются достаточно редко, так что большинство инструкций может быть исполнено напрямую, если известно, что они являются безвредными или привилегированными. Измерения показывают, что небольшое число гостевых страниц памяти требует принудительного программного моделирования. В результате работы было обнаружено, что современные процессоры Intel 64 содержат восемь инструкций, прямое исполнение которых может угрожать корректности моделирования.

Список литературы

1. Aarno D., Engblom J. Software and System Development using Virtual Platforms. 1st Edition. Elsevier, 2014. 366 p.

2. **Popek G. J., Goldberg R. P.** Formal Requirements for Virtualizable Third Generation Architectures // Communications of the ACM. 1974. P. 412–421.

3. **Речистов Г. С., Юлюгин Е. А., Иванов А. А. и др.** Основы программного моделирования ЭВМ: Учеб. пособие. М.: Издательство МФТИ, 2013. 222 с.

4. **Mihoka Darek, Shwartsman Stanislav.** Virtualization Without Direct Execution or Jitting: Designing a Portable Virtual Machine Infrastructure // ISCA-35 Proceedings of the 1st Workshop on Architectural and Microarchitectural Support for Binary Translation. 2008.

5. **Yair Lifshitz, Robert Cohn, Inbal Livni et al.** Zsim: A Fast Architectural Simulator for ISA Design-Space Exploration // 3rd Workshop on Infrastructures for Software/Hardware Co-Design. WISH-3. 2011.

6. **Intel Corporation.** Intel® 64 and IA-32 Architectures Software Developer's Manual. 2018. Vol. 1–4.

7. **Adams Keith, Agesen Ole.** A Comparison of Software and Hardware Techniques for x86 Virtualization // ASPLOS. 2006.

8. **Oracle Corporation.** Oracle VM VirtualBox® User Manual.

9. **Tulloch Mitch.** Understanding Microsoft Virtualization Solutions / Microsoft. 2 edition. Microsoft Press, 2010. 452 p.

10. **KVM wiki.** URL: http://www.linux-kvm.org/page/Main_Page (дата обращения: 16.08.2018).

11. **VMware ESXi: The Purpose-Built Bare Metal Hypervisor.** URL: <https://www.vmware.com/products/esxi-and-esx.html> (дата обращения: 16.08.2018).

12. **Wind River.** Simics Hindsight Guide 5.0. 2018.

13. **Agesen Ole, Mattson Jim, Rugina Radu, Sheldon Jeffrey.** Software Techniques for Avoiding Hardware Virtualization Exits // USENIX. 2012. P. 373–385.

14. **AMD.** Software Optimization Guide for AMD Family 10h and 12h Processors, 2011. 348 p.

15. **x86: emit tzcnt unconditionally.** URL: <https://gcc.gnu.org/ml/gcc-patches/2012-04/msg01765.html> (дата обращения: 16.08.2018).

E. A. Yulyugin, Software Engineer, e-mail: evgeny.yulyugin@intel.com, Intel Corporation

Fast and Accurate Direct Execution of Particular Intel® 64 Instructions in Virtual Environment

Intel® 64 architecture processors are constantly evolving, with new generations regularly introduced on the market. A new processor is usually backwards compatible and includes all the software-visible functionality of previous generations. This allows existing software to run on the new hardware without changes or recompilation. However, the guarantee of backwards compatibility can break in certain scenarios when software is running in a virtual machine. We discovered that certain machine instructions behave differently on past and present generations of AMD Intel processors. The identified instructions extend traditional understanding of hardware-assisted virtualization developed by Popek and Goldberg in 1974. To solve this problem new methods and tools have been developed to establish correct and fast hardware-assisted virtual platform solution. The key to solving this problem is to make sure that no such instruction is allowed to be directly executed in a virtualized environment — they all have to be emulated using software techniques. Detection of the instructions has to be done in advance since the problem cannot be corrected after the execution. The detection is achieved by a pattern matching decoder that looks through the guest code in prior to its execution in hardware-assisted virtual environment. The developed solution for the problem has been implemented, tested and proven to work using Wind River Simics® virtual platform framework.

Keywords: Wind River Simics, virtualization, software simulation, x86, Intel VT-x, hypervisor, lzcnt, virtual machine monitor, virtual platform, Simics, live migration

DOI: 10.17587/it.25.157-164

References

1. **Aarno Daniel, Engblom Jakob.** Software and System Development using Virtual Platforms, 1st Edition. Elsevier, 2014. 366 p.

2. **Popek G. J., Goldberg R. P.** Formal Requirements for Virtualizable Third Generation Architectures // Communications of the ACM, 1974. P. 412–421.

3. **Rechistov G., Yulyugin E., Ivanov A. et al.** Fundamentals of computer simulation software. 2nd edition revised and enlarged. Textbook. MIPT, 2013. 222 p. (in Russian).

4. **Mihoka Darek, Shwartsman Stanislav.** Virtualization Without Direct Execution or Jitting: Designing a Portable Virtual Machine Infrastructure // ISCA-35 Proceedings of the 1st Workshop on Architectural and Microarchitectural Support for Binary Translation, 2008.

5. **Yair Lifshitz, Robert Cohn, Inbal Livni et al.** Zsim: A Fast Architectural Simulator for ISA Design-Space Exploration // 3rd Workshop on Infrastructures for Software/Hardware Co-Design. WISH-3, 2011.

6. **Intel Corporation.** Intel® 64 and IA-32 Architectures Software Developer's Manual. Vol. 1–4, 2018.

7. **Adams Keith, Agesen Ole.** A Comparison of Software and Hardware Techniques for x86 Virtualization // ASPLOS, 2006.

8. **Oracle Corporation.** Oracle VM VirtualBox® User Manual.

9. **Tulloch Mitch.** Understanding Microsoft Virtualization Solutions / Microsoft. 2 edition. Microsoft Press, 2010. 452 p.

10. **KVM wiki.** Available at http://www.linux-kvm.org/page/Main_Page (date of access: 16.08.2018).

11. **VMware ESXi: The Purpose-Built Bare Metal Hypervisor.** Available at <https://www.vmware.com/products/esxi-and-esx.html> (date of access: 16.08.2018).

12. **Wind River.** Simics Hindsight Guide 5.0. — 2018.

13. **Agesen Ole, Mattson Jim, Rugina Radu, Sheldon Jeffrey.** Software Techniques for Avoiding Hardware Virtualization Exits // USENIX, 2012. P. 373–385.

14. **AMD.** Software Optimization Guide for AMD Family 10h and 12h Processors, 2011. 348 p.

15. **x86: emit tzcnt unconditionally.** Available at <https://gcc.gnu.org/ml/gcc-patches/2012-04/msg01765.html> (date of access: 16.08.2018).

В. Е. Гвоздев, д-р техн. наук, проф., e-mail: wega55@mail.ru,
Д. В. Блинова, канд. техн. наук, доц., e-mail: blinova.darya@gmail.com,
К. В. Кириллов, аспирант, e-mail: kir-konst@yandex.ru,
Уфимский государственный авиационный технический университет, г. Уфа

Информационная поддержка принятия решений по устранению дефектов, выявленных при эксплуатации систем обработки данных¹

Рассматривается формализованная процедура поддержки принятия решений при устранении дефектов, обнаруженных на стадии эксплуатации систем обработки данных (СОД). Основу процедуры составляет системное сочетание моделей "Профиль дефектов", "Интервальное оценивание вероятностей на основе выборочных данных", "Boston Consulting Group — BCG".

Вводится в рассмотрение понятие "профиль дефектов". Цель построения профиля дефектов — разработка систематического базиса для принятия решений при распределении ресурсов на поиск и устранение дефектов с учетом их значимости с точки зрения определенной целевой группы пользователей и степени изученности дефектов.

Описан подход к построению профиля дефектов по фактическим данным при ограниченном числе наблюдений. Особенностью подхода является интервальное оценивание вероятностей состояний, соответствующих различным слоям профиля.

Предложен способ, позволяющий адаптировать описанный в литературе по маркетингу подход к обеспечению информационной поддержки принятия решений по инвестированию средств в деятельность подразделений организации в условиях неопределенности внешней и внутренней среды — подход Boston Consulting Group (BCG), — применительно к задачам информационной поддержки принятия решений по устранению дефектов в СОД.

Использование предлагаемого подхода создает информационную основу целенаправленного распределения ограниченных ресурсов на устранение дефектов с учетом их влияния на оценку пользователями качества СОД. Приводится пример использования предлагаемого подхода по результатам опытной эксплуатации системы электронного документооборота DIRECTUM.

Ключевые слова: дефект, система обработки данных, аппаратно-программный комплекс, профиль дефектов, подход BCG

Введение

Удовлетворенность целевых групп пользователей потребительскими свойствами систем обработки данных (СОД) относится к ключевым условиям инвестирования средств в их развитие [1, 2]. Проявление дефектов в программных компонентах СОД на стадии эксплуатации систем относится к числу факторов, негативно влияющих на представление пользователей о качестве систем. Дефекты в программных компонентах СОД, проявляющиеся на стадии эксплуатации, возникают на разных стадиях жизненного цикла программных средств, обусловлены разными

причинами, относятся к разным классам (проектирования, кодирования, постановки задачи и др.) [3]. Проявления разных дефектов оказывают различное влияние на такую характеристику качества, как функциональная безопасность СОД. Отмеченные обстоятельства обуславливают необходимость дифференцированного подхода к устранению разных дефектов [4–6].

В настоящей работе рассматривается систематическая процедура информационной поддержки принятия решений о целесообразности устранения дефектов в программных компонентах, основанная на учете степени изученности разных дефектов в разных компонентах, а также того влияния, которое оказывает проявление дефектов на оценку пользователями качества СОД.

¹ Работа поддержана грантом РФФИ 16-08-00442.

Профили как компонент системы информационной поддержки принятия решений по управлению свойствами СОД

Профили являются разновидностью структурных (иерархических) моделей СОД. Под профилем понимается полное множество альтернатив (функций, операций, пользователей и т. д.), для каждой из которых существует определенная вероятность появления [5, 7]. Агрегаты профилей [8] являются инструментом информационной поддержки решения задач, связанных с управлением качеством СОД. Концептуальную основу построения профилей составляет стремление разработчиков на ранних стадиях жизненного цикла систем взглянуть на систему глазами пользователей, а также учесть предполагаемые условия их эксплуатации [5]. Обеспечение требуемого качества СОД в условиях ограниченных ресурсов проектов обеспечивается за счет избирательного подхода к назначению требований к свойствам компонентов системы с учетом роли, которую играют разные компоненты в обеспечении качества СОД.

В литературе описано значительное число разных профилей, соответствующих различным подходам к исследованию свойств систем [5]:

- потребительский профиль, предназначенный для исследования целевых групп пользователей;
- профиль системных режимов, предназначенный для исследования возможных условий использования системы;
- функциональный профиль, предназначенный для исследования функций, которые может реализовать система;
- операционный профиль, предназначенный для исследования операций, выполняемых системой и т. д.

В данной работе вводится в рассмотрение "профиль дефектов", представленный на рис. 1. Целью построения этого профиля является создание систематической основы для оценки степени изученности дефектов. Изученность трактуется как свойство, обратное неопределенности, и характеризует возможность успешного устранения дефекта при ограничениях на доступные ресурсы.

Подчеркнем методическое отличие профиля дефектов на стадии эксплуатации СОД от других профилей, используемых на ранних стадиях жизненного цикла систем.

При построении профилей дефектов на ранних стадиях жизненного цикла систем используются точечные оценки вероятностей альтернатив, значения которых обеспечивают выполнение условия нормировки, т. е. равенства единице сум-

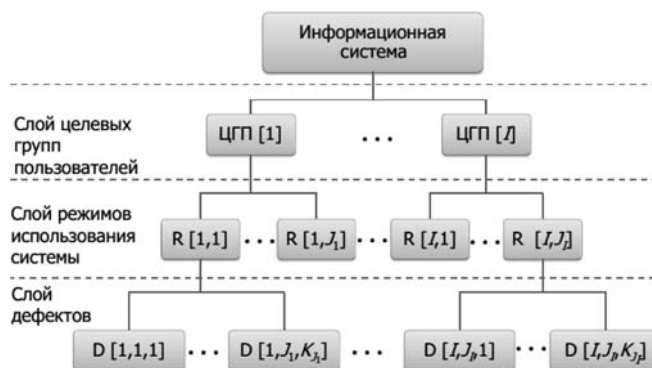


Рис. 1. Профиль дефектов

мы вероятностей. Основанием для построения профиля дефектов по результатам эксплуатации являются фактические сведения о проявлениях дефектов. Если сведения представлены в виде документальных данных, их объем является малым (в статистическом смысле). Часто документальные данные вообще отсутствуют и заменяются экспертными оценками, которые получают при опросе специалистов, проводивших обслуживание и эксплуатацию системы.

Следствием этого является тот факт, что вероятностные оценки характеристик профиля оказываются интервальными [9]. Использование же интервальных оценок осложняет выполнение условия нормировки для полной группы независимых событий [10]. В этом случае применение уже существующих методов анализа профилей для исследования профиля дефектов невозможно.

Ниже приводится описание задачи исследования свойства профиля дефектов при интервальных оценках вероятностей реализации альтернатив.

Содержание задачи построения профиля дефектов

Исходные данные

1. Полагается, что известно число I целевых групп пользователей (ЦГП), число режимов использования СОД J_i ($i = \overline{1; I}$) разными ЦГП, число классов дефектов K_{J_i} , соответствующих разным режимам использования СОД каждой из ЦГП.

2. Из опыта эксплуатации СОД полагаются известными:

2.1. Число обращений n_i ($i = \overline{1; I}$) к СОД каждой из i -й ЦГП, при которых наблюдались проявления дефектов.

2.2. Число использований m_{ij} ($i = \overline{1; I}$, $j = \overline{1; J_i}$) СОД в разных режимах разными ЦГП, при которых наблюдались проявления дефектов.

2.3. Число дефектов l_{ijk} , соотносимых с заранее определенными классами ($k = \overline{1; K_{J_i}}$), зарегистрированных в j -м режиме использования СОД i -й ЦГП.

Допущения

1. Если при разных режимах использования СОД различными ЦГП встречаются проявления одинаковых дефектов, при построении профиля они рассматриваются как разные дефекты. Это обусловлено тем, что одно и то же отклонение поведения СОД от ожидаемого по-разному воспринимается разными субъектами — представителями различных ЦГП (обоснование этого утверждения приводится в работе [5]).

2. Классификация дефектов инвариантна к режимам использования и к характеристикам ЦГП. При построении профиля дефектов выделяются следующие классы дефектов:

2.1. Ошибки пользователей и обслуживающего персонала.

2.2. Дефекты кодирования программной компоненты СОД.

2.3. Дефекты проектирования СОД.

2.4. Дефекты технического задания.

Особенности построения профиля дефектов при ограниченном числе наблюдений

Сложность построения профиля дефектов в условиях ограниченного числа наблюдений обусловлена необходимостью интервального оценивания вероятностей альтернатив, соответствующих различным слоям профиля, что диктуется малым объемом выборочных данных [11]. Следствием интервального оценивания вероятностей является возможность нарушения условия нормировки (равенства единице суммы вероятностей). Соблюдение условия нормировки является принципиальным требованием построения профилей [5, 7].

Кроме того, по мере продвижения от корня к листьям дерева (рис. 1), объемы выборочных данных уменьшаются, следствием чего является расширение границ, в которых с заданной доверительной вероятностью β находится истинное значение вероятности [11].

Основу построения профиля дефектов при интервальном оценивании вероятностей альтернатив составляет следующее. Вероятность пребывания объекта в одном из возможных состояний может трактоваться как весовая характеристика состояния [12]. Известно правило преобразования весовых характеристик α_i ($i = \overline{1; n}$) к виду нормированных весовых характеристик $\alpha_i^{(N)}$:

$$\alpha_i^{(N)} = \frac{\alpha_i}{\sum_{i=1}^n \alpha_i},$$

так что $\alpha_i^{(N)} \in [0; 1] \forall i$.

Предположим, что в результате независимого случайного выбора значений вероятностей p_i , каждая из которых принадлежит интервалу $p_i \in [p_i^{(H)}; p_i^{(B)}]$ ($p_i^{(H)}; p_i^{(B)}$ — соответственно нижняя и верхняя границы интервала), получены значения, для которых не соблюдается условие нормировки $\sum_{i=1}^n p_i \neq 1$. Преобразуем значение p_i к нормированному виду

$$p_i^{(N)} = \frac{p_i}{\sum_{i=1}^n p_i},$$

что обеспечит возможность реализации формальной процедуры построения профиля, описанного в работах [5, 13].

Основу оценивания степени изученности дефектов на основе профиля дефектов составляет проведение имитационного эксперимента по представленной ниже схеме:

Шаг 1. Инициализация исходных данных в виде:

- n_i — заданное число обращений i -й ЦГП к СОД ($i = \overline{1; I}$), при которых наблюдались проявления дефектов, I — число ЦГП;
- m_{ij} — заданное число обращений i -й ЦГП к СОД в j -м режиме использования ($i = \overline{1; I}, j = \overline{1; J_i}$), при которых наблюдались проявления дефектов, J_i — число режимов использования СОД i -й ЦГП;
- l_{ijk} — заданное число обращений i -й ЦГП к СОД в j -м режиме использования ($i = \overline{1; I}, j = \overline{1; J_i}$), при которых наблюдались проявления дефектов k -го типа ($k = \overline{1; K_{J_i}}$), K_{J_i} — число типов дефектов в j -м режиме использования i -й ЦГП;
- H — граничное значение числа повторений эксперимента.

Шаг 2. Расчет частот обращений к СОД каждой из ЦГП в каждом из режимов использования, в которых проявляются различные типы дефектов:

- $\hat{p}_i = \frac{n_i}{\sum_i n_i}$ — частота обращений i -й ЦГП к СОД ($i = \overline{1; I}$), при которых наблюдались проявления дефектов;
- $\hat{q}_{ij} = \frac{m_{ij}}{\sum_j m_{ij}}$ — частота обращений i -й ЦГП к СОД в j -м режиме использования ($i = \overline{1; I}, j = \overline{1; J_i}$), при которых наблюдались проявления дефектов;

- $\hat{w}_{ij} = \frac{l_{ijk}}{\sum_k l_{ijk}}$ — частота обращений i -й ЦГП

к СОД в j -м режиме использования ($i = \overline{1, I}; j = \overline{1, J_i}$), при которых наблюдались проявления дефектов k -го типа ($k = \overline{1, K_{J_i}}$).

Шаг 3. Задание значения доверительной вероятности β и расчет на основе вычисленных частот доверительных границ. Верхним индексом $^{(H)}$ и $^{(B)}$ обозначены соответственно нижние и верхние доверительные границы значенных частот, указанных в шаге 2:

$$p_i \in [p_i^{(H)}, p_i^{(B)}]; q_{ij} \in [q_{ij}^{(H)}, q_{ij}^{(B)}];$$

$$w_{ijk} \in [w_{ijk}^{(H)}, w_{ijk}^{(B)}].$$

Шаг 4. Генерация случайного значения частоты $p_i^{(\eta)}$, где η — номер эксперимента ($\eta = \overline{1, H}$), i — номер ЦГП ($i = \overline{1, I}$), соответствующего равномерному закону распределения с плотностью, обратно пропорциональной величине $(p_i^{(B)} - p_i^{(H)})$.

Шаг 5. Расчет случайного числа дефектов в i -й ЦГП в η -м эксперименте:

$$n_i^{(\eta)} = N p_i^{(\eta)} \quad (\eta = \overline{1, H}; i = \overline{1, I}).$$

Шаг 6. Генерация случайного значения частоты $q_{ij}^{(\eta)}$ ($i = \overline{1, I}; j = \overline{1, J_i}$), i — номер ЦГП, j — номер режима использования СОД, соответствующего равномерному закону распределения с плотностью, обратно пропорциональной величине $(q_{ij}^{(B)} - q_{ij}^{(H)})$.

Шаг 7. Расчет случайного числа дефектов при j -м режиме использования i -й ЦГП в η -м эксперименте:

$$m_{ij}^{(\eta)} = n_i^{(\eta)} q_{ij}^{(\eta)} \quad (\eta = \overline{1, H}; i = \overline{1, I}; j = \overline{1, J_i}).$$

Шаг 8. Генерация случайного значения частоты $w_{ijk}^{(\eta)}$ ($i = \overline{1, I}; j = \overline{1, J_i}; k = \overline{1, K_{J_i}}$), соответствующего равномерному закону распределения с плотностью, обратно пропорциональной величине $(w_{ijk}^{(B)} - w_{ijk}^{(H)})$.

Шаг 9. Расчет случайного числа дефектов k -го типа при j -м режиме использования i -й ЦГП в η -м эксперименте:

$$l_{ijk}^{(\eta)} = m_{ij}^{(\eta)} w_{ijk}^{(\eta)} \quad (\eta = \overline{1, H}; i = \overline{1, I}; j = \overline{1, J_i}; k = \overline{1, K_{J_i}}).$$

Шаг 10. Расчет эмпирической оценки частоты обращений к СОД с проявлением дефектов, т. е. нормированных характеристик явных профилей дефектов в η -м эксперименте:

$$p_i^{*(\eta)} = \frac{n_i^{(\eta)}}{\sum_i n_i^{(\eta)}}; q_{ij}^{*(\eta)} = \frac{m_{ij}^{(\eta)}}{\sum_j m_{ij}^{(\eta)}}; w_{ijk}^{*(\eta)} = \frac{l_{ijk}^{(\eta)}}{\sum_k l_{ijk}^{(\eta)}}.$$

Шаг 11. Расчет характеристик неявного профиля в η -м эксперименте:

$$\text{Вер}_{ijk}^{*(\eta)} = p_i^{*(\eta)} q_{ij}^{*(\eta)} w_{ijk}^{*(\eta)}.$$

Увеличение значения $\eta = \eta + 1$. Переход к шагу 4, если $\eta \leq H$.

Шаг 12. Нахождение граничных значений характеристик неявных профилей:

$$\text{Вер}_{ijk}^{(H)} = \min_{\eta} \{ \text{Вер}_{ijk}^{*(\eta)} \}; \text{Вер}_{ijk}^{(B)} = \max_{\eta} \{ \text{Вер}_{ijk}^{*(\eta)} \}.$$

Шаг 13. Расчет оценки степени изученности дефектов, соответствующих неявным профилям, в виде

$$H_{ijk} = \frac{1}{\text{Вер}_{ijk}^{(B)} - \text{Вер}_{ijk}^{(H)}}.$$

Основанием для оценивания степени изученности дефектов соотношением, представленным в шаге 13, является следующее. Количественной характеристикой неявного профиля согласно работе [7] является произведение вероятностей, соответствующих ребрам простого пути от корня к ijk -му листу дерева (рис. 1).

В случае отсутствия эмпирических данных обоснованной оценкой диапазона возможных значений неявного профиля будет $\text{Вер}_{ijk} \in [0, 1]$, чему соответствует длина интервала $d = 1 - 0 = 1$. После оценки диапазона вероятных значений $\{ \text{Вер}_{ijk}^{(H)}, \text{Вер}_{ijk}^{(B)} \}$ интервал сократится до $d_{ijk} = \text{Вер}_{ijk}^{(B)} - \text{Вер}_{ijk}^{(H)}$. Иными словами, изученность дефекта увеличится в $H_{ijk} = 1 / (\text{Вер}_{ijk}^{(B)} - \text{Вер}_{ijk}^{(H)})$ раз.

Особенностью представленной схемы эксперимента является тот факт, что объемы случайных выборок, соответствующих разным слоям $m_i^{(\eta)}$, $k_i^{(\eta)}$ при η -й реализации эксперимента (шаг 3), определяются с учетом случайных значений вероятностей, находящихся в границах интервалов, соответствующих вышерасположенному слою. Соблюдение условия нормировки обеспечивается тем, что при каждом η -м эксперименте на основе сгенерированных случайных выборок определяются точечные оценки вероятностей (частоты), соответствующие разным состоянием различных слоев профиля (шаг 10). Обозначения, используемые в шаге 3, подчеркивают то обстоятельство, что ширина интервалов определяется выбранной доверительной вероятностью β .

Пример построения профиля дефектов

В табл. 1 приведены значения показателя изученности дефектов, рассчитанные посред-

ством описанной схемы для профиля, представленного на рис. 2.

Результаты, представленные в табл. 1, соответствуют следующим исходным данным:

$$n_1 = 9000; n_2 = 1000;$$

$$m_{1,1} = 8100; m_{1,2} = 900; m_{2,1} = 900; m_{2,2} = 100;$$

$$l_{1,1,1} = 3240; l_{1,1,2} = 2430; l_{1,1,3} = 1620; l_{1,1,4} = 810;$$

$$l_{1,2,1} = 360; l_{1,2,2} = 270; l_{1,2,3} = 180; l_{1,2,4} = 90;$$

$$l_{2,1,1} = 360; l_{2,1,2} = 270; l_{2,1,3} = 180; l_{2,1,4} = 90;$$

$$l_{2,2,1} = 40; l_{2,2,2} = 30; l_{2,2,3} = 20; l_{2,2,4} = 10.$$

В ходе расчетов принималось $\beta = 95\%$. Число повторений эксперимента составляло $H = 10^3$.

Таблица 1

Результаты вычислительного эксперимента

Номер ЦГП	Номер режима	Класс дефекта	$Вер_{ijk}^{(H)}$	$Вер_{ijk}^{(B)}$	$Вер_{ijk}^{(B)} - Вер_{ijk}^{(H)}$	Характеристика изученности дефекта
1	1	1	0,3563	0,4454	0,0891	11,2233
1	1	2	0,2541	0,3503	0,0962	10,3950
1	1	3	0,1535	0,2564	0,1029	9,7182
1	1	4	0,0575	0,1683	0,1108	9,0253
1	2	1	0,2774	0,5366	0,2592	3,8580
1	2	2	0,1788	0,4575	0,2787	3,5881
1	2	3	0,0895	0,3888	0,2993	3,3411
1	2	4	0,0212	0,3634	0,3422	2,9223
2	1	1	0,2774	0,5366	0,2592	3,8580
2	1	2	0,1788	0,4575	0,2787	3,5881
2	1	3	0,0895	0,3888	0,2993	3,3411
2	1	4	0,0212	0,3634	0,3422	2,9223
2	2	1	0,1267	0,7539	0,6272	1,5944
2	2	2	0,0654	0,7240	0,6586	1,5184
2	2	3	0,0236	0,7210	0,6974	1,4339
2	2	4	0,0035	0,7803	0,7768	1,2873

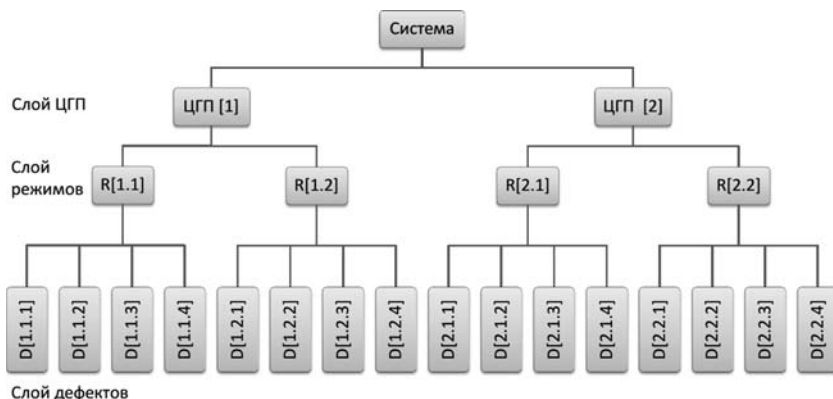


Рис. 2. Пример профиля дефектов

Поддержка принятия решений по устранению дефектов на основе подхода VCG

Уточненная модель Project Triangle, предложенная в 2014 г. [1, 2], ориентирует разработчиков на создание программных систем, во-первых, представляющих ценность для пользователей; во-вторых, приводящих к удовлетворению от работы с ними [14].

Одним из базовых положений применения профилей как инструмента управления качеством СОД является избирательный подход к улучшению потребительских свойств системы, предоставляющей сервисы для разных целевых групп пользователей и функционирующей в разных режимах [5].

Профиль дефектов позволяет оценить степень изученности дефектов. Однако при принятии решения о целесообразности затрат ресурсов на устранение дефекта следует также принимать во внимание степень негативного влияния дефекта на оценку разными пользователями качества СОД. Принимая во внимание различие ролей, которые разные ЦГП могут играть при выделении ресурсов на решение задач в рамках стратегического управления потребительскими свойствами СОД, при анализе дефектов следует учитывать, интересы какой ЦГП и в какой степени затрагиваются проявлениями дефектов.

В работе [15] описан инструмент к обеспечению информационной поддержки принятия решений по инвестированию средств в деятельность разных подразделений организации с учетом (а) доли рынка, контролируемой подразделением, (б) темпов роста рынка (подход Boston Consulting Group — BCG). Рассматривая значимость дефектов как аналог темпа роста рынка, а изученность дефекта как аналог относительной доли рынка, можно адаптировать подход BCG в область принятия решений о

целесообразности расходования ресурсов на устранение дефектов (заметим, что целесообразность адаптации подходов, хорошо зарекомендовавших себя при решении задач управления сложными системами, в область программной инженерии отмечается в работе [6]).

Структура матрицы, соответствующей задаче устранения дефектов, аналогичной матрице, используемой в подходе BCG, приведена на рис. 3.

Выделенным блокам матрицы можно дать следующие толкования.

Если дефекты соотносятся с областью {"Высокая значимость", "Высокая изученность"}, то такие дефекты должны устраняться в первую очередь, так как это создает основу для инвестирования ресурсов в совершенствование потребительских свойств СОД.

Если дефекты соотносятся с областью {"Низкая значимость", "Высокая изученность"}, то такие дефекты могут устраняться даже без дополнительного непосредственного инвестирования ресурсов со стороны разработчиков, поскольку этот вид деятельности способствует повышению репутации разработчиков (обслуживающего персонала) в глазах пользователей. Это, в свою очередь, создаст предпосылки для появления заказов в будущем.

Дефекты, соотносящиеся с областью {"Высокая значимость", "Низкая изученность"}, соответствуют подключению к системе новых функций. Выделение инвестиций на устранение дефектов целесообразно, так как подключение новых функций обусловлено решением задач развития возможностей СОД.

Если дефекты соотносятся с областью {"Низкая значимость", "Низкая изученность"}, то тратить ресурсы на их устранение не имеет смысла.

Диаметр кругов, показанных на рис. 3, соответствует числу зафиксированных дефектов, соотносимых с каждым из выше определенных классов. Цифры в кругах соответствуют классам дефектов. Наличие этих сведений, а также экспертные оценки объемов ресурсов, затрачиваемых на устранение дефектов разных классов, создают основу для планирования суммарных объемов ресурсов, необходимых для устранения дефектов в целом.

Модель, учитывающая использование СОД разными целевыми группами пользователей (куб матриц ЦГП), представлена на рис. 4.

В модели, представленной на рис. 4, учитываются различные возможности влияния разных ЦГП на инвестирование в содержание и развитие СОД. Необходимость ранжирования пользователей (правообладателей) по степени их влияния на проекты и процессы обсуждается в работах [14, 16–22 и др.].

Пример из практики

На российском рынке информационных систем представлена система электронного документооборота (СЭД) DIRECTUM. Данная система может гибким образом подстраиваться под нужды каждого заказчика, специфику деятельности и особенности внутренних процессов и электронного документооборота того или иного предприятия. Это достигается путем разработки пользовательских программных компонентов системы (типовых маршрутов, мастеров действий и др.) как силами самого предприятия, так и силами компаний-партнеров, специализирующихся в разработке программного обеспечения.

В любом случае при разработке программных компонентов возможно возникновение дефектов.

На этапе опытной эксплуатации данной системы на одном из предприятий в течение одного месяца было выявлено 90 ошибок. Пользователи системы относятся к двум основным целевым группам:

- 1) рядовые специалисты;
- 2) руководители подразделений.

На этапе эксплуатации работа с системой каждой из ЦГП осуществлялась только в одном режиме. Выявленные ошибки (дефекты) были соотнесены с каждым из выше определенных классов, а именно:

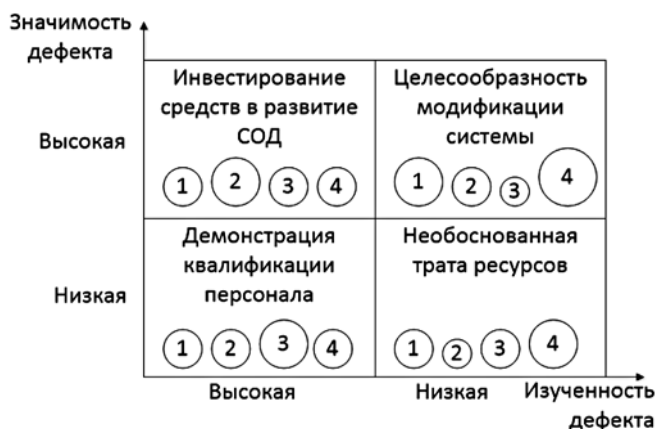


Рис. 3. Аналог матрицы BCG применительно к описанию дефектов признаками "изученность дефекта", "значимость дефекта"

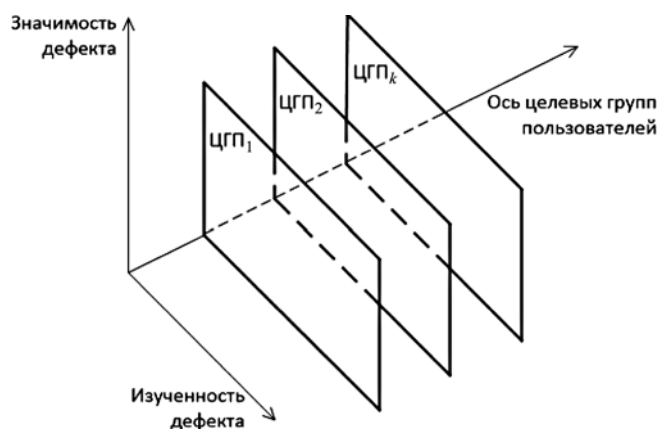


Рис. 4. Куб матриц BCG, соответствующих разным целевым группам пользователей

- 1) ошибки пользователей и обслуживающего персонала;
- 2) дефекты кодирования программного компонента;
- 3) дефекты проектирования;
- 4) дефекты технического задания (в том числе — постановки задач).

Объемы выборочных данных составили:

- 1) $n_1 = 79, n_2 = 11$ — число дефектов, обнаруженных ЦГП "Рядовые специалисты" и "Руководители подразделений" соответственно;
- 2) $l_{1,1,1} = 54, l_{1,1,2} = 10, l_{1,1,3} = 8, l_{1,1,4} = 7$ — число дефектов вышеуказанных классов, обнаруженных ЦГП "Рядовые специалисты";
- 3) $l_{2,1,1} = 7, l_{2,1,2} = 2, l_{2,1,3} = 1, l_{2,1,4} = 1$ — число дефектов вышеуказанных классов, обнаруженных ЦГП "Руководители подразделений".

На основе вычислительного эксперимента, схема которого была представлена выше, были рассчитаны неявные профили, на основе которых была сформирована табл. 2, аналогичная табл. 1, для целевой группы пользователей "Рядовые специалисты".

На основе полученных значений была сформирована матрица (рис. 5), аналогичная матрице, представленной на рис. 3.

На основе матрицы, изображенной на рис. 5, с учетом доступных финансовых и вре-

менных ресурсов, выделяемых на устранение дефектов, были приняты решения о распределении ресурсов на устранение разных классов дефектов. На рис. 5 наглядно показано, каких дефектов было больше (диаметр кругов) и к какой категории в системе координат "Изученность — Значимость" они относятся. На основании данных матрицы возможно принятие решения о целесообразности устранения того или иного дефекта.

Аналогичные исследования были проведены для ЦГП "Руководители подразделений".

В качестве примеров ошибок, имевших место при эксплуатации системы, можно привести следующие:

1. Расчет сроков для согласования осуществляется не в рабочих, а в календарных днях. Это является ошибкой кодирования.

2. Неправильно вычисляется роль куратора (документ может попасть на согласование к работнику, не имеющему отношения к нему, в вычислениях используются нестандартные, специально разработанные реквизиты). Это ошибка проектирования.

3. Отсутствует проверка на наличие электронной подписи, но это не критично, поскольку для данной категории документов не требуется обеспечение юридической значимости с использованием электронной подписи. Это ошибка в постановке задач, поскольку данный аспект не был учтен при формулировании задания на программирование.

4. Возможен ввод отрицательного числа в поле суммы документа. Это пользовательская ошибка, обусловленная недочетами в эксплуатационной документации.

Согласно опыту разработчиков и аналитиков было установлено, что дефект № 1 относится к категории {"Высокая значимость", "Высокая изученность"}, дефект № 2 — {"Высокая значимость", "Низкая изученность"}, а дефекты № 3 и № 4 относятся к категории {"Низкая значимость", "Высокая изученность"}. Аналогичным образом происходила идентификация остальных дефектов в приведенном примере.

Таким образом, в соответствии с полученными результатами были приняты следующие решения по организации процесса сопровождения системы на этапе эксплуатации:

1. Направить дополнительные ресурсы на повышение качества разработки пользовательской документации. В связи с тем что номенклатура пользовательской документации обширна, то целесообразно увеличить штат системных аналитиков, ответственных за раз-

Таблица 2

Расчет неявных профилей на основе примера из практики для ЦГП "Рядовые специалисты"

Класс дефекта	$Ver_{ijk}^{(B)} - Ver_{ijk}^{(H)}$	Характеристика изученности дефекта
1	0,0279	35,8423
2	0,0464	21,5517
3	0,0472	21,1864
4	0,0475	21,0526

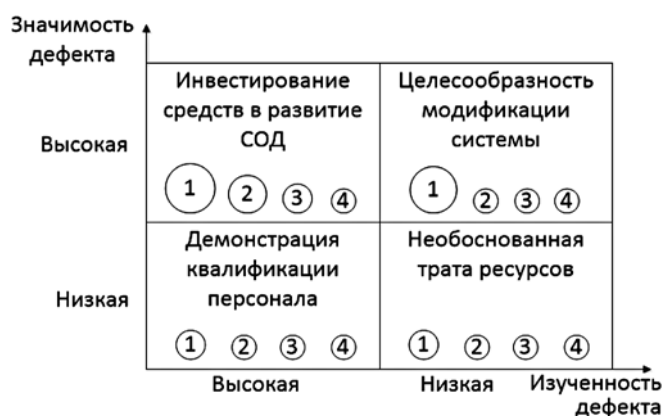


Рис. 5. Матрица "Изученность дефекта — значимость дефекта" для ЦГП "Рядовые специалисты" СЭД DIRECTUM

работку документации, по крайней мере, на одну штатную единицу.

2. Провести мониторинг пользовательской документации, провести ее обновление, привести ее к единому формату.

3. Направить дополнительные ресурсы на осуществление очного обучения пользователей работе в системе на регулярной основе. Для этого целесообразно увеличить штат системных аналитиков, ответственных за обучение пользователей, по крайней мере, на одну штатную единицу.

Заключение

В работе описана систематическая процедура поддержки принятия решений по устранению дефектов, выявленных на стадии эксплуатации СОД. Основу процедуры составляет сочетание моделей "Профиль дефектов", "Интервальное оценивание вероятностей", "BCG". Предлагаемый подход позволяет повысить обоснованность принятия решений по инвестированию ресурсов в повышение качества СОД за счет совместного использования формальных методов структурного моделирования, методов математической статистики и знаний экспертов.

Список литературы

1. **Chaos Manifesto** 2014: Value versus Success & The Orthogonal. The Standish Group International, Incorporated. URL: <http://blog.standishgroup.com/post/14> (дата обращения 19.02.2017).
2. **Тимофеев А. Н.** Почему падают IT-проекты // Практика проектирования систем. 2017. С. 2–11.
3. **Gvozdev V. E., Kirillov K. V., Subhangulova A. S., Morozov A.** Effect of different defect types on static parameters of reliability assessment // The 18th International Workshop on Computer Science and Information Technologies (CSIT'2016) September 26–29, 2016 Czech Republic, Prague (p. 57–61).
4. **Гвоздев В. Е., Абдрафиков М. А., Блинова Д. В., Кириллов К. В.** Анализ влияния дефектов различных типов на оценку показателей качества программно-аппаратных комплексов на

стадии сопровождения // Электротехнические и информационные комплексы и системы. 2017. Т. 13, № 2. С. 33–40.

5. **Мороз Г. Б., Коваль Г. И., Коротун Т. М.** Концепция операционных профилей в инженерии надежности программных систем // Математические машины и системы. 2004. № 1.

6. **Липаев В. В.** Функциональная безопасность программных средств. М.: СИНТЕГ, 2004. 348 с.

7. **Lyu M. R. et al.** Handbook of Software Reliability Engineering. IEEE Computer Society Press and McGraw-Hill Book Company, 1996.

8. **Антонов А. В.** Системный анализ. Учеб. пособ. для вузов. М.: Высшая школа, 2004. 454 с.

9. **Гвоздев В. Е., Костюкова А. П., Кириллов К. В.** Статистическое оценивание показателей надежности систем обработки данных и управления по неполным сведениям о выборочных данных // Электротехнические и информационные комплексы и системы. 2016. № 4, Т. 12. С. 73–78.

10. **Макконел С.** Сколько стоит программный проект. СПб.: Питер, 2007. 297 с.

11. **Вентцель Е. С.** Теория вероятностей. М.: Издательский центр "Академия", 2003. 576 с.

12. **Вентцель Е. С.** Исследование операций. Задачи, принципы, методология. М.: Наука, 1980. 208 с.

13. **Cheung R. A.** User-oriented Software Reliability Model // IEEE Trans. Soft. Eng. 1980, SE-6, N. 2.

14. **Брю Г.** Шесть сигм для менеджеров / Пер. с англ. В. Н. Егорова. М.: ФАИР-ПРЕСС. 2004. 272 с.

15. **Котлер Ф.** Маркетинг Менеджмент / Пер. с англ. под ред. С. Г. Божук. СПб.: Питер, 2006. 464 с.

16. **Гельруд Я. Д., Логиновский О. В.** Управление проектами: методы, модели, системы. Челябинск: Издательский центр ЮУрГУ, 2015. 330 с.

17. **Bourne L.** Project Relationships and stakeholder Circle // PMI Research Conference, 2006.

18. **Worthington D., Al-Chaith T., Brown D.** Stakeholder dynamics in strategic ICT projects: appreciate systems perspective. Paper presented at BAM, Liverpool, 2013.

19. **Olander S.** Stakeholder impact analysis in construction project management // Construction Management and Economics. 2007. Vol. 25 (3). P. 277–287.

20. **Виттих В. А.** Неоднородный актор и повседневность как ключевые понятия эвергетики: препринт. Самара: Изд. ИПУСС РАН, 2014. 12 с.

21. **Hu Wenhua, Jeffrey C. Carver, Vaibhav Anu, Gursimran Wilia, Gary Bradshaw.** Defect Prevention in Requirements Using Human Error Information: An Empirical Study // Springer International Publishing. Ab 2017. P. Grnbacher and A. Perinin (sds) REFSQ 2017. LNCS 10153. P. 61–76.

22. **Embrey D.** Understanding Human Behavior and Error // Human Reliability Associates, Ltd. 10 p.

V. E. Gvozdev, D. Sc., Professor, wega55@mail.ru,
D. V. Blinova, Ph. D., Associate Professor, blinova.darya@gmail.com,
K. V. Kirillov, Postgraduate Student, kir-konst@yandex.ru,
Ufa State Aviation Technical University, Ufa, Russian Federation

Information Decision-Making Support on Elimination of the Defects Detected at Exploitation of Data Processing Systems

We consider a formalized decision support procedure for eliminating defects found during the operation of data processing systems. The procedure is based on a system combination of the models "Defect Profile" and "Interval Probability Estimation Based on Sample Data" by "Boston Consulting Group — BCG".

The concept of "Defect Profile" is introduced into consideration. The purpose of building a defect profile is to develop a systematic basis for decision-making in the allocation of resources for finding and eliminating defects, taking into account their importance from the point of view of a certain target user group and the degree of knowledge of the defects.

An approach to constructing a defect profile using actual data with a limited number of observations is described. The approach features the interval estimation of the probabilities of states corresponding to different layers of the profile.

A method is proposed for adapting the approach described in the marketing literature to providing information support for decision-making on investing funds into the activities of the organization's units under uncertainty of the external and internal environment — the Boston Consulting Group (BCG) approach — to the problems of information support for decision-making on defects elimination in Data Processing Systems (DPS).

The use of the proposed approach creates an informational basis for the targeted allocation of limited resources for defects elimination, taking into account their impact on the users' assessment of quality of DPS. An example of the use of the proposed approach based on the results of trial operation of the DIRECTUM electronic document management system is given.

Keywords: defect, data processing system, hardware and software complex, Defects' Profile, BCG approach

Acknowledgements: This article was prepared with the financial support of Russian Foundation for Basic Research (project no. 16-08-00442).

DOI: 10.17587/it.25.165-173

References

1. **The Standish Group** International, Incorporated, CHAOS MANIFESTO 2014: Value versus Success & The Orthogonal, Available at: <http://blog.standishgroup.com/post/14>
2. **Timofeev A. N.** *Pochemu padayut IT-proekty* (Why are IT-projects fail?), *Praktika Proektirovaniya Sistem*, pp. 2—11, 2017 (in Russian).
3. **Gvozdev V. E., Kirillov K. V., Subhangulova A. S., Morozov A.** Effect of different defect types on static parameters of reliability assessment, *Proc. the 18th International Workshop on Computer Science and Information Technologies (CSIT'2016)*, Czech Republic, Prague, September 26—29, 2016, pp. 57—61.
4. **Gvozdev V. E., Abdrafikov M. A., Blinova D. V., Kirillov K. V.** *Analiz vliyaniya defektov razlichnyh tipov na ocenku pokazatelej kachestva programmno-apparatnyh kompleksov na stadii soprovozhdeniya* (Analysis of different defect types influence on quality indicators assessment of hardware and software complexes at the maintenance stage), *Jelektrotehnicheskie i Informacionnye Kompleksy i Sistemy*, 2017, vol. 13, no. 2, pp. 33—40 (in Russian).
5. **Moroz G. B., Koval G. I., Korotun T. M.** *Primenenie koncepcii operacionnyh profilej v inzhenerii nadezhnosti programnyh sistem* (The application of the operational profiles' concept in reliability engineering software systems), *Matematicheskie Mashiny i Sistemy*, 2004, no. 1 (in Russian).
6. **Lipaev V. V.** *Funktsional'naya bezopasnost' programnykh sredstv* (Software functional safety), Moscow, SINTEG, 2004, 348 p. (in Russian).
7. **Lyu M. R. et al.** *Handbook of Software Reliability Engineering*, IEEE Computer Society Press and McGraw-Hill Book Company, 1996.
8. **Antonov A. V.** *Sistemnyj analiz* (System analyses. Textbook for high schools), Moscow, Vysshaya shkola, 2004, 454 p. (in Russian).
9. **Gvozdev V. E., Kostjukova A. P., Kirillov K. V.** *Statisticheskoe otsenivanie pokazatelej nadezhnosti sistem obrabotki dannykh i upravleniya po nepolnym svedeniyam o vyborochnykh dannykh* (Statistical estimation of reliability indices of data processing and management systems based on incomplete information on sample data), *Jelektrotehnicheskie i Informacionnye Kompleksy i Sistemy*, 2016, vol. 12, no. 4, pp. 73—78 (in Russian).
10. **McConnel S.** *Software estimation: Demystifying the Black Art*, Saint-Petersburg, Piter, 2007, 297 p. (in Russian).
11. **Ventsel E. S.** *Teoriya veroyatnostej* (Probability theory. Textbook for students of high schools), Moscow, Izdatelskiy Tsentr "Akademiya", 2003, 576 p. (in Russian).
12. **Ventsel E. S.** *Operation research. Objectives, principles, methodology*, Moscow, Nauka, 1980, 208 p. (in Russian).
13. **Cheung R. A.** *User-oriented Software Reliability Model // IEEE Trans. Soft. Eng.* 1980, SE-6, N 2.
14. **Bru G.** *Shest' sigm dlya menedzherov* (The six sigma for managers), Moscow, FAIR-PRESS, 2004, 272 p. (in Russian).
15. **Kotler F.** *Marketing Menedzhment* (Marketing. Management), Saint-Petersburg, Piter, 2006, 464 p. (in Russian).
16. **Gelrud Ya. D.** *Upravlenie proektami: metody, modeli, sistemy* (Projects Management: methods, models, systems), Russia, Chelyabinsk, Publishing Center of SUSU, 2015, 330 p. (in Russian).
17. **Bourne L.** *Project Relationships and stakeholder Circle, Proc. the PMI Research Conference*, 2006.
18. **Worthington D., Al-Chaith T., Brown D.** *Stakeholder dynamics in strategic ICT projects: appreciate systems perspective*, Liverpool, BAM, 2013.
19. **Olander S.** *Stakeholder impact analysis in construction project management*, *Construction Management and Economics*, 2007, vol. 25 (3), pp. 277—287.
20. **Vittikh V. A.** *Neodnorodnyj aktor i povsednevnost' kak klyucheveye ponyatiya ehvergetiki: preprint* (Heterogeneous actor and everyday life as key concepts of evergetic), Russia, Samara, Publishing house of Institut problem upravleniya slozhnymi sistemami RAN, 2014, p. 64 (in Russian).
21. **Wenhua Hu, Jeffrey C. Carver, Vaibhav Anu, Gursimran Wilia, Gary Bradshaw.** *Defect Prevention in Requirements Using Human Error Information: An Empirical Study*, Springer International Publishing, Ab 2017, P. Grnbacher and A. Perinin (eds) REFSQ 2017, LNCS 10153, pp. 61—76.
22. **Embrey D.** *Understanding Human Behavior and Error*, Human Reliability Associates, Ltd, 10 p.

Н. Т. Абдуллаев, канд. техн. наук, доц., a.namik46@mail.ru,
Азербайджанский технический университет,
К. Ш. Исмаилова, канд. техн. наук, доц., Is_kamalya@yahoo.com,
Азербайджанский государственный университет нефти и промышленности

Выбор шага итерации в процессе обучения нейронной сети при использовании релаксационных алгоритмов

Рассматриваются методы выбора шага итерации при обучении нейронных сетей. Представлена таблица возможных методов выбора шага и рекомендуемые варианты применения этих методов. Разработан алгоритм выбора шага при использовании релаксационных алгоритмов с учетом эффективности и временных затрат реализации. Данный алгоритм эффективен в реальной работе и прост в реализации. Помимо этого такой алгоритм удобно применять вместе с алгоритмами "тяжелый шарик", PARTAN и даже BFGS, которые широко используются в реальных задачах.

Ключевые слова: нейронная сеть, релаксационные алгоритмы, шаг итерации, методы градиентного спуска, эффективность, простота реализации, оптимизация, обучение

Введение

В информационных источниках имеется достаточно много материалов, посвященных алгоритмам оптимизации и реализации этих алгоритмов для нейронных сетей. Тем не менее вопросы алгоритмического обеспечения при выборе шага итерации для обучения нейронных сетей достаточно актуальны.

В методе градиентного спуска [1] осуществляется поиск по самому крутому направлению спуска, противоположному направлению градиента целевой функции. Он может обеспечить уменьшение целевой функции до тех пор, пока текущая точка итерации не станет неподвижной. Различные варианты ступенчатых размеров приводят к различным алгоритмам градиентного спуска. Распространенным вариантом метода градиентного спуска является метод точного поиска по линии, который выбирает следующую итерацию путем достижения наименьшего значения функции. Однако этот метод имеет недостаток, поскольку часто имеет медленную сходимость.

В последние годы были проведены исследования, направленные на выбор размера шага для метода градиентного спуска согласно рекомендациям Барзилая и Борвеина [2], где дан

конкретный выбор размера шага, подтверждающий суперлинейную сходимость для двумерных выпуклых квадратичных задач, и указаны рекомендуемые размеры шага для метода градиентного спуска.

Авторы работы [3] используют один из вариантов метода градиентного спуска для повышения эффективности наиболее применяемого алгоритма при обучении многослойных нейронных сетей — Backpropagation, который использует адаптивную скорость обучения, суть которого заключается в обратном распространении ошибок с переменным шагом. Параллельные касательные методы используются в глобальной оптимизации для изменения и улучшения простого алгоритма градиентного спуска, в этом случае иногда при поиске направления спуска используется разница между текущей точкой и точкой до двух шагов от текущей. Для обучения нейронной сети авторы предлагают комбинацию метода BPVS с методом Parallel Tangent.

В работе [4] рассматривается вариация часто используемого алгоритма для решения проблемы выбора трафика (в транспорте) — алгоритм Франка — Вольфа. Вариацией этого алгоритма является алгоритм PARTAN (сокращение от параллельного касания), в котором ускоре-

ние конвергенции достигается путем введения дополнительного алгоритма поиска по линии. Показано, что в варианте PARTAN алгоритма Франка — Вольфа для вычисления минимальной длины шага для распределения равновесия могут быть использованы либо аналитические формулы, либо простые правила.

В последние годы методы стохастического градиентного спуска (СГС) стали стандартным инструментом для обучения нейронных сетей. В [5] проведен анализ конвергенции для СГС на богатом подмножестве двухслойных сетей прямого доступа с функциями активации ReLU (Rectified Linear Units). Это подмножество характеризуется специальной структурой, называемой "отображение идентичности". Доказывается, что если входные данные выбраны из гауссовского распределения при стандартной инициализации весов, то СГС сходится к глобальному минимуму за полиномиальное число шагов. В отличие от обычных сетей "отображение идентичности" делает такую сеть асимметричной и, следовательно, глобальный минимум уникальным. В дополнение к теории можно экспериментально показать, что многослойные сети с этим отображением имеют лучшую производительность по сравнению с нормальными сетями. Такой подход к сходимости алгоритма отличается от традиционных невыпуклых методов оптимизации.

В работе [6] исследуются алгоритмы автоматической адаптации скорости обучения нейронных сетей (НС). Начиная со стохастического градиентного спуска, большое разнообразие методов обучения было предложено для настройки НС. Однако эти методы обычно чувствительны к начальной скорости обучения, которая должна быть выбрана экспериментатором. В работе [6] исследуются несколько функций и показано, как адаптивный контроллер может настроить обучение без предварительного знания изучаемой проблемы.

В статье [7] рассмотрен метод обучения для двухслойных первичных нейронных сетей на основе анализа чувствительности, который использует алгоритм линейного обучения для каждого из двух уровней. Сначала выходам присваиваются случайные значения первого уровня; позже эти начальные значения обновляются на основе формул чувствительности, которые используют веса в каждом из слоев.

Поскольку эти веса изучаются при решении линейной системы уравнений, то сохраняется важная экономия вычислительного времени. При этом появляется возможность анализа

локальной чувствительности методом наименьших квадратов для оценки ошибок по входным и выходным данным без дополнительных вычислительных затрат, поскольку необходимая информация становится доступной без дополнительных вычислений. Основное преимущество анализа локальной чувствительности реализуется, в частности, в многомерных задачах, где глобальный анализ требует слишком много времени. Этот метод, называемый методом линейного обучения на основе чувствительности, также может быть использован для обеспечения начального набора весов, что значительно улучшает поведение других алгоритмов обучения. Дана теоретическая основа метода, и его производительность иллюстрируется применением на нескольких примерах, в которых его сравнивают с разными алгоритмами обучения на хорошо известных наборах данных. Результаты показали высокую скорость обучения по сравнению с другими существующими методами. Кроме того, его можно использовать в качестве инструмента инициализации для других хорошо известных методов со значительными улучшениями.

В работе [8] предложен алгоритм оптимизации обучения неглубоких нейронных сетей. Высокоразмерные стохастические задачи оптимизации представляют собой интересные проблемы для таких существующих алгоритмов обучения. Разрабатывается расширение, подходящее для алгоритмов оптимизации обучения, и демонстрируется, что изученный алгоритм оптимизации последовательно превосходит другие известные алгоритмы оптимизации даже для труднорешаемых задач и устойчив к изменениям стохастичности градиентов и архитектуры нейронной сети.

В работе [9] вводится новый метод "важности веса" (IW) для ускорения обучения на основе градиента. Этот метод особенно полезен для "сложных" наборов данных, включая несбалансированные данные, данные с высоконелинейными отношениями между переменными или с долгосрочными зависимостями в последовательностях. Характеристика "важность веса" присваивается каждой точке данных обучающей выборки и контролирует вклад точки данных в общую ошибку обучения в соответствии с его информативностью до получения результатов хорошей предсказуемости. Характеристику "важность веса" можно также использовать для индивидуального выбора размера шага для локального градиента в конкретной точке данных. Эта цель достига-

ется путем квадратичной оптимизации, которая сводит к минимуму абсолютное значение изменения вектора параметров во время этапа обучения с (мягким) ограничением, так чтобы общая ошибка была уменьшена на наименьшее заданное фиксированное значение. Для линейных классификаторов (способ решения задач классификации, когда решение принимается на основании линейного оператора над входными данными) показано, что данный метод поддерживает стандарты векторного обучения. Применяется метод IW и к многослойным перцептронам, и к рекуррентным нейронным сетям (LSTM).

Методы выбора шага итерации

Выбор в пользу градиентных методов обоснован тем, что, как правило, в задачах обучения критерий обучения можно выразить в виде дифференцируемой функции от весов нейронной сети. Тем не менее неопределенность выбора метода обучения сохраняется. В автоматизированных системах нейросетевого программирования следует стремиться к сокращению неопределенности, которая присуща этим технологиям. Неопределенность в выборе алгоритма обучения в некоторой степени устраняется в предлагаемом адаптивном методе обучения [10].

На основании проведенного анализа выбора шага итерации рассмотрены методы и их применение. Результаты сведены в таблицу.

Описание выбора шага и применение алгоритмов

Метод выбора шага	Применение
Задать постоянным (убывание градиента при приближении к минимуму)	Градиентные методы
Увеличить шаг при убывании ошибки и уменьшить при возрастании	Простые релаксационные алгоритмы (алгоритм "тяжелого шарика", PARTAN, BFGS (Broyden — Fletcher — Goldfarb — Shanno algorithm))
Последовательное приближение зависимости функционала ошибки от шага с помощью квадратичной зависимости	Методы сопряженных градиентов, BFGS, методы второго порядка
Вычислить производную функционала ошибки в нуле	Методы сопряженных градиентов, BFGS, методы второго порядка
Определение направления с большими вычислительными затратами	Метод статистического градиента

Различные формы регулярных итеративных методов отличаются друг от друга конкретным выбором шага [11].

Алгоритмическая реализация выбора шага

К алгоритмам оптимизации с "нелинейным" шагом и соответствующей дискретной системой с нелинейными коэффициентами усиления относятся релаксационные алгоритмы, в которых постоянная матрица на каждом шаге выбирается так, чтобы уменьшалась функция ошибки $E(w_k + \eta_k - \rho_k)$, где w_k — вектор коэффициентов, ρ_k — параметр, влияющий на изменение направления, η_k — размер шага на k -й итерации.

Как известно, если ρ_k — направление убывания, то алгоритмы "тяжелый шарик" и PARTAN удобно применять при выборе шага. Идея выбора состоит в том, чтобы увеличивать шаг при убывании ошибки и уменьшать при возрастании. Алгоритм выбора шага итерации более удачный и простой в реализации [12] и его удобно применять с релаксационными алгоритмами. К таким относится известный алгоритм наискорейшего спуска. Предлагается алгоритм выбора шага, который состоит из пяти шагов (см. рисунок).

Шаг 1. В начале всего алгоритма задается начальный шаг. Задаются параметры алгоритма, удовлетворяющие следующим неравенствам: $\gamma_1 > 1$; $0 < \gamma_2 < 1$.

Шаг 2. После нахождения очередного направления движения в качестве начального приближения берется шаг из предыдущей итерации.

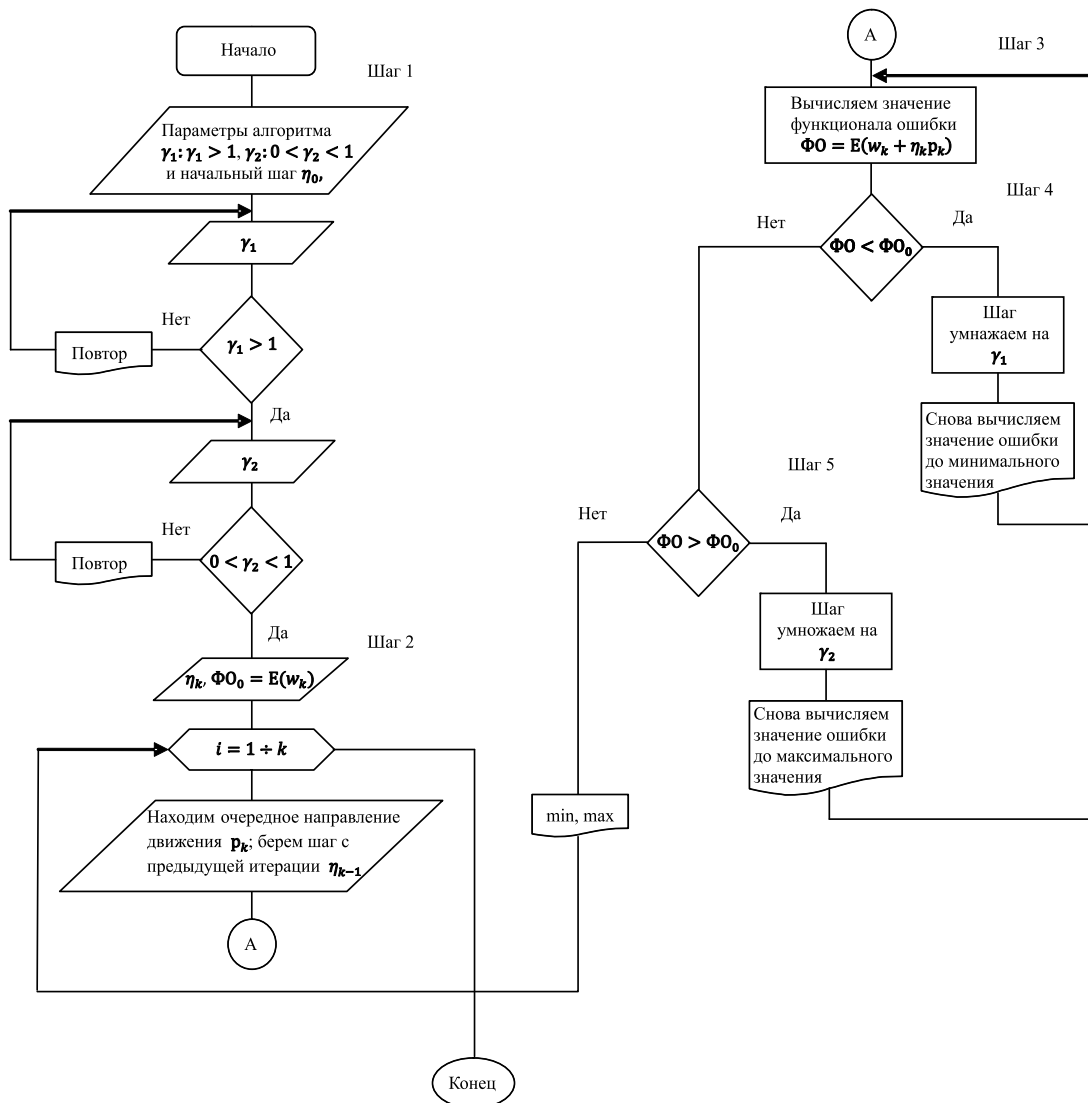
Шаг 3. Вычисляется значение функционала ошибки $\Phi_0 = E(w_k + \eta_k - \rho_k)$.

Шаг 4. Если это значение меньше предыдущего $E(w_k)$, тогда шаг умножаем на γ_1 и снова вычисляем значение ошибки до тех пор, пока ошибка не начинает возрастать. Предпоследняя минимальная точка фиксируется, и далее вычисляется новое направление.

Шаг 5. Если $E(w_k + \eta_k - \rho_k) > E(w_k)$, шаг умножается на γ_2 , и продолжается уменьшение шага, пока ошибка не станет меньше $E(w_k)$. Полученный шаг фиксируется, и вычисляется новое направление.

Заключение

Данный алгоритм эффективен в реальной работе и прост в реализации. Такой алгоритм удобно применять вместе с алгоритмами "тяжелый шарик", PARTAN и даже BFGS, кото-



Алгоритм выбора шага итерации

рые широко используются в реальных задачах. Таблица методов выбора шага и их реализации полезна пользователям при выборе шага итерации в экспериментах.

Список литературы

1. **Ya-xiang Y.** Step-Sizes for the Gradient Method. URL: <ftp://lsec.cc.ac.cn/pub/yyx/papers/p0504.pdf> 30.07.2018
2. **Barzilai J., Borwein J. M.** Two point step size gradient methods // IMA J. Numer. Anal. 1988. Vol. 8. P. 141–148.
3. **Petalas Y. G., Vrahatis M. N.** Parallel Tangent Methods with Variable Stepsize. URL: <http://www.math.upatras.gr/~petalas/ijcnn04.pdf> (дата обращения: 30.07.2018)
4. **Arezki Y., Van Vliet D.** A Full Analytical Implementation of the PARTAN/Frank-Wolfe Algorithm for Equilibrium Assignment. URL: https://www.researchgate.net/publication/244956239_A_Full_Analytical_Implementation_of_the_PARTANFrank-Wolfe_Algorithm_for_Equilibrium_Assignment (дата обращения: 30.07.2018)
5. **Yuanzhi Li, Yang Y.** Convergence Analysis of Two-layer Neural Networks with ReLU Activation // 31st Conference on

Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. 2017. 11 p.

6. **Daniel Ch., Taylor J., Nowozin S.** Learning Step Size Controllers for Robust Neural Network Training // Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16). 2016. P. 1519–1525.
7. **Castillo E., Guijarro-Berdinas B., Fontenla-Romero O., Alonso-Betanzos A.** A Very Fast Learning Method for Neural Networks Based on Sensitivity Analysis // Journal of Machine Learning Research. 2006. Iss.7. P. 1159–1182.
8. **Li K., Jitendra M.** Learning to Optimize Neural Nets. URL: <https://arXiv:1703.00441v2> [cs.LG]. 2017. 10 p. (дата обращения: 05.09.2018)
9. **Hochreiter S., Obermayer K.** Optimal Gradient-Based Learning Using Importance Weights. URL: www.bioinf.jku.at/publications/20051205.pdf (дата обращения: 05.09.2018)
10. **Борисович Л. В.** Алгоритм и программная реализация адаптивного метода обучения искусственных нейронных сетей. URL: <https://cyberleninka.ru/article/n/algoritm-i-programmnaya-realizatsiya-adaptivnogo-metoda-obucheniya-iskusstvennyh-neuronnyh-setey.pdf> (дата обращения: 10.08.2018)
11. **Разновидности** алгоритмов оптимизации. URL: http://sernam.ru/book_zip.php?id=17 (дата обращения: 01.08.2018)
12. **Тархов Д. А.** Нейросетевые модели и алгоритмы. Справочник. М.: Радиотехника, 2014. 352 с.

N. T. Abdullayev, Ph. D., Associate Professor, e-mail: a.namik46@mail.ru,
Azerbaijan Technical University,

K. Sh. Ismayilova, Ph. D., Associate Professor, e-mail: Is_kamalya@yahoo.com,
Azerbaijan State University of Oil and Industry

The Choice of the Iteration Step in the Process of Learning a Neural Network Using Relaxation Algorithms

In information sources, there are a lot of materials devoted to algorithms for optimization and implementation of these algorithms for neural networks. However, the issues of algorithmic support in selecting the iteration step, for learning neural networks are quite relevant. The choice in favor of gradient methods is justified by the fact that, as a rule, in learning problems, the learning criterion can be expressed in the form of a differentiable function of the weights of the neural network. Nevertheless, the uncertainty of the choice of the method of instruction is preserved. In automated systems of neural network programming, one should strive to reduce the uncertainty that is inherent in these technologies. The uncertainty in the choice of the learning algorithm is to some extent eliminated in the proposed adaptive learning method. The algorithm for selecting a step consists of 5 levels (steps).

Step 1. At the beginning of the whole algorithm the initial step is set. Parameters of the algorithm must satisfy the following inequalities $-\gamma_1: \gamma_1 > 1, \gamma_2: 0 < \gamma_2 < 1$.

Step 2. After finding the next direction of motion, the step from the previous iteration is taken as the initial approximation.

Step 3. After that, the value of the error functional $E(w_k + \eta_k - \beta p_k)$ is calculated.

Step 4. If this value is less than the previous one $E(w_k)$, then step umnazhaem to γ_1 and again calculate the value of the error until the error begins to increase. The penultimate minimum point is fixed and a new direction is computed.

Step 5. If $E(w_k + \eta_k - \beta p_k) > E(w_k)$, the step is multiplied by γ_2 and the step decreases until the error becomes less than $E(w_k)$. The resulting step is fixed and a new direction is calculated.

This algorithm is effective in real work and easy to implement. In addition, it is convenient to apply this algorithm together with the heavy ball, Partan and even BFGS algorithms, which are widely used in real problems. The table of methods for selecting a step and their implementation is useful to users when choosing the iteration step in experiments.

Keywords: neural network, relaxation algorithms, iteration step, gradient methods, efficiency, prostate implementation, optimization, learning

DOI: 10.17587/it.25.174-178

References

1. **Ya-xiang Y.** Step-Sizes for the Gradient Method, available at: <ftp://lsec.cc.ac.cn/pub/yx/papers/p0504.pdf> 30.07.2018
2. **Barzilai J., Borwein J. M.** Two point step size gradient methods, *IMA J. Numer. Anal.*, 1988, vol. 8, pp. 141–148.
3. **Petalas Y. G., Vrahatis M. N.** Parallel Tangent Methods with Variable Stepsize, available at: <http://www.math.upatras.gr/~petalas/ijcnn04.pdf> (date of access: 30.07.2018)
4. **Arezki Y., Van Vliet D.** A Full Analytical Implementation of the PARTAN/Frank-Wolfe Algorithm for Equilibrium Assignment, available at: https://www.researchgate.net/publication/244956239_A_Full_Analytical_Implementation_of_the_PARTANFrank-Wolfe_Algorithm_for_Equilibrium_Assignment (date of access: 30.07.2018)
5. **Yuanzhi Li, Yang Y.** Convergence Analysis of Two-layer Neural Networks with ReLU Activation, *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017, 11 p.
6. **Daniel Ch., Taylor J., Nowozin S.** Learning Step Size Controllers for Robust Neural Network Training, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016, pp. 1519–1525.
7. **Castillo E., Guijarro-Berdinas B., Fontenla-Romero O., Alonso-Betanzos A.** A Very Fast Learning Method for Neural Networks Based on Sensitivity Analysis, *Journal of Machine Learning Research*, 2006, iss.7, pp. 1159–1182.
8. **Li K., Jitendra M.** Learning to Optimize Neural Nets, available at: <https://arXiv:1703.00441v2> [cs.LG], 2017, 10 p. (date of access: 05.09.2018).
9. **Hochreiter S., Obermayer K.** Optimal Gradient-Based Learning Using Importance Weights, available at: www.bioinf.jku.at/publications/20051205.pdf (date of access: 05.09.2018).
10. **Borisovich L. V.** *Algoritim i programnaya realizatsiya adaptivnogo metoda obucheniya iskusstvennyh neyronnyh setey* (Algorithm and software implementation of the adaptive method of training artificial neural networks), available at: <https://cyberleninka.ru/article/n/algoritim-i-programmnaya-realizatsiya-adaptivnogo-metoda-obucheniya-iskusstvennyh-neyronnyh-setey>.pdf (date of access: 10.08.2018).
11. **Raznovidnosti algoritmov optimizatsii** (Varieties of optimization algorithms), available at: http://sernam.ru/book_zip.php?id=17 (date of access: 01.08.2018).
12. **Tarhov D. A.** *Neyrosetevye modeli i algoritmi* (Neural network models and algorithms), Moscow, Radiotekhnika, 2014, 352 p. (in Russian).

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В БИМЕДИЦИНСКИХ СИСТЕМАХ

INFORMATION SYSTEMS IN BIOMEDICAL SYSTEMS

УДК 659.3

DOI: 10.17587/it.25.179-184

Е. В. Артюшина, канд. экон. наук, доц., eartushina@mail.ru,

Д. Е. Ершов, канд. экон. наук, erschow@mail.ru,

Национальный исследовательский университет "Высшая школа экономики", г. Нижний Новгород,

Ю. А. Прохода, juprokhoda@gmail.com,

АО "Медиаскоп", г. Москва

Роль интернет-технологий в коммуникационной деятельности медицинской организации

Статья содержит результаты исследования роли интернет-технологий в коммуникационной деятельности медицинских организаций. Изучена практика поиска и выбора потребителями конкретного медицинского центра для обслуживания. Выделены шесть ключевых групп источников информации, используемых потребителями, и 10 групп параметров выбора медицинской организации. Сформирована структура информационно-коммуникационной среды. Результаты исследования могут быть использованы российскими медицинскими организациями.

Ключевые слова: интернет-технологии, коммуникационная деятельность, информационно-коммуникационная среда, медицинские организации

Введение

Перед руководителями медицинских организаций (МО) постоянно встает задача коммуникации с целевой аудиторией. Эффективно организованная коммуникационная деятельность позволяет повысить узнаваемость организации, способствует выстраиванию устойчивых взаимоотношений с целевым рынком, а в долгосрочной перспективе является основой будущего процветания организации, ее конкурентоспособности. Авторы российской и зарубежной литературы отмечают, что коммуникационная деятельность медицинских организаций в большой степени ориентирована на вовлеченность потребителя [1–4]. При этом в качестве основной задачи коммуникационной деятельности ставится задача повышения узнаваемости медицинской организации и, как следствие, увеличение клиентской базы.

Усиление конкуренции и развитие частного сектора оказания медицинских услуг в России привели к росту числа коммуникационных инструментов, используемых компаниями, работающими в медицинской сфере. Развитие информационных технологий и появление новых

коммуникационных каналов оказали влияние и на принятие потребителями решений о выборе места оказания медицинских услуг [2]. Сегодня пациенты полагаются не только на указания и рекомендации своего врача, они могут свободно найти информацию о диагнозах и их лечении, а также обменяться опытом в социальных сетях. Повышение осведомленности потребителей в медицинской сфере приводит к тому, что потребители постепенно перестают полагаться исключительно на мнение лечащего врача [3]. Ярким примером этого является использование нативной информации в социальных сетях, когда потребитель в поисках информации об определенном заболевании находит статью, содержащую подробное описание проблемы с указанием медицинского центра, где возможно провести профессиональное обследование [5].

Исследования, посвященные практике использования интернет-технологий в коммуникационной деятельности МО, в современной литературе представлены недостаточно. Несомненный интерес представляет структурирование источников информации, используемых потребителями при поиске МО для обслуживания. Важно понимать, какие параметры яв-

ляются для потребителя определяющими при выборе МО.

В условиях высокой информационной вовлеченности и осведомленности потребителя научный и практический интерес представляет проведение исследования, целью которого будет формирование структуры информационно-коммуникационной среды МО. Данный вывод подтверждают результаты практических исследований, отражающие повышение интереса компаний, работающих в медицинской сфере, к поиску и выбору информационно-коммуникационных каналов [2]. Под информационно-коммуникационной средой мы понимаем совокупность каналов, позволяющих организации реализовать обмен информацией с целевой аудиторией для достижения результата.

Методы

В целях формирования структуры информационно-коммуникационной среды авторами в период с 15.02.18 по 25.06.18 было проведено исследование конечных потребителей услуг МО, структурно состоящее из трех этапов (табл. 1).

На первом этапе было проведено качественное исследование с использованием метода глубинных интервью. Было проинтервьюировано 96 респондентов, являющихся клиентами МО из разных городов России. На втором этапе для подтверждения полученных результатов качественного исследования было запущено количественное исследование. Сформированная нулевая гипотеза (*H₀*) — "сеть Интернет используется в качестве основного информационного

канала при поиске медицинского центра для обслуживания". Количественное исследование было проведено в форме телефонного опроса, а также личного интервьюирования с использованием случайной выборки. Размер выборки определен исходя из требований, предъявляемых к точности и достоверности результатов исследования. Репрезентативный размер выборки составил 349 респондентов исходя из требований к уровню доверительного интервала в 95 % и принятой точности исследования в 5 %. В процессе исследования было опрошено 387 человек. Принято к анализу 354 анкеты. Третий этап исследования был основан на обобщении результатов анализа практик осуществления коммуникационной деятельности МО в интернет-среде.

Результаты

Полученные результаты исследования представлены в виде таблиц и диаграмм.

Результаты глубинных интервью, проводимых в ходе качественного исследования, представлены в табл. 2. Результаты обобщены в группы по направлениям "источники информации, которые потребители используют при поиске МО" (выделено шесть ключевых групп) и "параметры, являющиеся определяющими при выборе МО" (выделено 10 групп).

В ходе проведения качественного исследования интернет-ресурс выделен в качестве ключевого источника информации. В качестве основного параметра, определяющего выбор той или иной МО, выделена "рекомендация". Полученные результаты качественного исследе-

Таблица 1

Этапы проведенного исследования

Этап	Цель	Метод	Результат
1	Формирование базы для проведения количественного исследования	Качественное исследование. Глубинные личные интервью. Глубинные <i>on-line</i> интервью	Выделены шесть ключевых групп источников информации, 10 групп параметров выбора, используемых потребителями. Результаты требуют количественного подтверждения
2	Подтвердить либо опровергнуть нулевую гипотезу исследования	Ad hoc количественное исследование. Телефонный опрос, личное интервьюирование 387 респондентов	Искомый параметр — процентное соотношение респондентов, использующих сеть Интернет как основной канал поиска информации, — 61 %. Нулевая гипотеза подтверждена. Сформирована структура информационно-коммуникационной среды МО
3	Внедрение результатов исследования в практическую деятельность МО	Кабинетное исследование; анализ полученных результатов, экспертная оценка	Предложены рекомендации по осуществлению коммуникационной деятельности МО в интернет-среде

Таблица 2

Результаты качественного исследования потребителей услуг медицинских центров

Вопросы в ходе проведения глубинных интервью	Выделенные группы на основе анализа ответов респондентов
Ключевые источники информации при поиске МО	<ul style="list-style-type: none"> ● рекомендации существующих пользователей, знакомых (полученные в личной беседе, по телефону); ● рекомендации существующих пользователей, знакомых (получены на форумах, обсуждениях в социальных сетях); ● рекомендации врачей; ● сайт в сети Интернет; ● реклама (ТВ, радио, газеты, журналы); ● проживают рядом
Ключевые параметры при выборе МО	<ul style="list-style-type: none"> ● рекомендации уже существующих пользователей МО, знакомых, друзей; ● рекомендации врачей; ● репутация МО; ● удобное расположение; ● высокий уровень обслуживания; ● цена на услуги; ● конкретный специалист; ● квалификация врачей; ● уникальность предоставляемых услуг; ● реклама

дования легли в основу проведенного количественного исследования (результаты представлены на рис. 1 (см. третью сторону обложки) и рис. 2).

Полученные результаты в рамках количественного этапа исследования отражают основной информационный канал, используемый потребителями, и выделяют интернет-технологии как основу коммуникационной деятельности МО. При выборе медицинского центра для обслуживания 61 % потребителей использует интернет-ресурсы. Из них 46 % потребителей руководствуются рекомендациями уже существующих пользователей или знако-



Рис. 2. Количественное распределение наиболее важных параметров выбора потребителями медицинской организации для обслуживания

Таблица 3

Достоверность результатов исследования

Параметр	Процент потребителей, использующих сеть Интернет в качестве основного канала поиска информации о МО
Искомый параметр	61 %
Гарантированная точность	± 5 %
Доверительный интервал	(61 ± 5) %

мых, полученными через информационные каналы сети Интернет (социальные сети, форумы) и 15 % потребителей при выборе медицинского центра обращаются к поиску сайта организации в сети Интернет. Выявленная последовательность действий потенциального потребителя МО выглядит следующим образом: потребитель вбивает в поисковой системе нужную ему услугу, заходит на первые по результатам поиска несколько сайтов компаний, предоставляющих необходимую ему услугу, изучает их описание и цены, далее продолжает изучение в удобной для него социальной сети, знакомится с отзывами, обсуждениями и затем формирует свой выбор. Таким образом, нулевая гипотеза исследования подтверждена. Для характеристики точности и достоверности полученных данных определим доверительный интервал (табл. 3). Искомый нами параметр находится в диапазоне от 56 до 66 %.

Анализ данных, отраженных на рис. 2, определяет основную идею коммуникаций МО. При выборе медицинского центра для обслуживания основным параметром, который является определяющим и практически в два раза более весомым, чем остальные параметры, является рекомендация уже существующих пользователей. Именно идею "рекомендации" и важно транслировать в процессе коммуникационной активности через сформированную информационно-коммуникационную среду, структура которой схематично представлена на рис. 3.

Структура информационно-коммуникационной среды, предложенная авторами, является упрощенной, описывает основные информационные каналы и может быть использована МО для донесения идеи коммуникации до целевого рынка. Важность формиро-

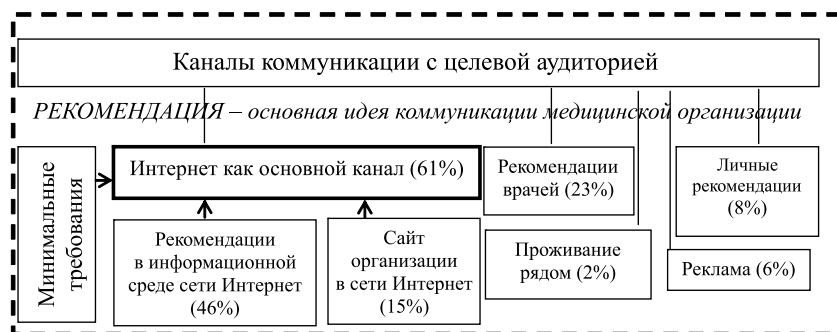


Рис. 3. Структура информационно-коммуникационной среды медицинской организации

вания среды обусловлена значимостью канала передачи сообщения от источника к получателю. В коммуникационной среде современного "электронного общества" более важным элементом становится не само сообщение (*message*), а средство его передачи (*medium*). Ошибочно выбранные каналы будут блокировать получение сообщения целевой аудиторией.

Выделение сети Интернет как базового канала информационно-коммуникационной среды обусловило необходимость изучения коммуникационной деятельности МО, осуществляемой с помощью данной технологии, и формирования практических рекомендаций. Авторами сформированы следующие минимальные требования присутствия МО в сети Интернет (стандартные рекомендации по кастомизации сайта являются общедоступными в литературе и не приводятся в рамках данной статьи):

1. Нахождение сайта организации в топе поисковых систем:

- адаптация под мобильные устройства. По данным исследования Calltouch 2016 г. в сфере медицинских услуг наблюдается существенный рост запросов с мобильных устройств. В 2016 г. 43 % запросов идут с мобильных устройств [4]. С 2017 г. по результатам исследований различных компаний статистика просмотра сайтов с мобильных устройств превысила статистику просмотров с других типов устройств, а на сегодня этот показатель доходит до 84 %;
- размещение на сайте раздела "Отзывы". По данным агентства Websitebuilder, пользователи доверяют отзывам в сети Интернет в 12 раз больше, чем описаниям услуг, которые предоставляются самими организациями [8]. Также агентство выделило пять главных индустрий, на которые отзывы оказывают влияние. В их числе "кабинет врача" и "больницы". Исследования агентства Vendast так-

же говорят о важности отзывов. По результатам их исследований, 92 % пользователей читают отзывы в сети Интернет и 88 % пользователей им доверяют [7]. Российский сайт отзывов yell.ru составил рейтинг ТОП-10 категорий "О чем читают отзывы в России?". По результатам исследований, наиболее популярной категорией оказалась медицина;

- наличие кнопки целевого действия.

2. Присутствие организации в социальных сетях с акцентом внимания на блоки "Обсуждения", "Отзывы", "Комментарии", "Рекомендации специалистов", "Форум". В России проникновение соцсетей оценивается в 47 %, аккаунты в них имеют 67,8 млн россиян. По данным Statista, активнее всего в РФ используют YouTube (63 % опрошенных), второе место занимает ВКонтакте — 61 %. Глобальный лидер Facebook лишь на четвертой строчке с показателем в 35 %. Среди мессенджеров доминируют Skype и WhatsApp (по 38 %) [8].

Сформированная авторами схема информационно-коммуникационной среды была протестирована в процессе коммуникационной деятельности медицинского центра города Набережные Челны. В процессе тестирования была изменена идея коммуникации с "Ваш семейный доктор" (существующая до начала исследования) на "Нас рекомендуют" и ее транслирование через выделенные каналы информационно-коммуникационной среды. В дополнение к этому на интернет-сайт организации добавлена кнопка "Запись на прием", организующая процесс *on-line* записи. В течение месяца через кнопку целевого действия в организацию обратилось 34 новых клиента (что составило 4,25 % от среднемесячного числа обращений новых клиентов).

Дальнейшие направления исследования

Ввиду многогранности проводимого исследования в данной статье есть ряд аспектов, которые представляют собой перспективное поле для дальнейших исследований.

В эмпирической части исследования при выделении групп источников информации не проведено разделение, какой из существующих каналов интернет-среды является наиболее результативным. В связи с этим интерес представляет анализ эффективности каж-

дого канала интернет-коммуникации в отдельности.

Несомненный интерес представляет сравнительный анализ практик применения интернет-технологий в коммуникационной деятельности российских и зарубежных МО.

Также ценным направлением последующих исследований является изучение возможности прогнозирования того, каким образом интенсивность использования основных информационных каналов влияет на результат экономической деятельности МО.

Выводы

Коммуникационная деятельность способствует росту узнаваемости организации, привлечению потребителей, выстраиванию устойчивых взаимоотношений с целевым рынком. Применение интернет-технологий является в современных рыночных условиях основой эффективной коммуникации. Современный человек всегда on-line, устройство для выхода в сеть Интернет у него всегда с собой [9].

До настоящего момента роль интернет-технологий в коммуникационной деятельности российских МО была мало изучена. Наше исследование вносит теоретический и практический вклад в изучение вопросов коммуникационной деятельности. Интерес авторов к теме исследования обусловлен требованиями сегодняшнего общества к информационному обеспечению деятельности современных организаций. Поведение современного потребителя медицинских услуг характеризуется высокой самостоятельностью в условиях информационной перегруженности.

Вопрос выбора информационного канала для донесения коммуникационного сообщения до целевой аудитории возникает перед каждой

медицинской организацией. В ходе исследования авторами выделено шесть ключевых групп источников информации, используемых потребителями при поиске МО для обслуживания, 10 групп параметров, являющихся для потребителей определяющими при выборе медицинского центра. Авторами сформирована структура информационно-коммуникационной среды МО, определяющая основополагающую роль сети Интернет в ее коммуникационной деятельности. Предложены рекомендации по присутствию МО в интернет-среде. Результаты исследования протестированы при планировании и реализации коммуникационной деятельности медицинского центра города Набережные Челны и могут быть использованы российскими МО.

Список литературы

1. **Reilly M. J., Markenson D. S.** Health care emergency management: Principles and practice. Jones & Bartlett Publishers, 2010.
2. **Moser H. R., Freeman Jr G. L.** An empirical analysis of the public's attitudes toward advertising hospital services: A comparative cross-sectional study // Health marketing quarterly. 2014. Vol. 31, N. 1. С. 13–30.
3. **Smith K. T.** Hospital Marketing and Communications Via Social Media // Services Marketing Quarterly. 2017. Vol. 38, N. 3. С. 187–201.
4. **Исследование** в сфере медицинских услуг // Calltouch.ru. URL: <https://blog.calltouch.ru/issledovanie-v-sfere-meditsinskih-uslug> (дата обращения 12.04.2018)
5. **Nanda S., Telang A., Bhatt G.** Hospital advertising: A literature review // International Journal of Healthcare Management. 2012. Vol. 5, N. 1. С. 28–31.
6. **How** Online Reviews Can Either Make or Break Your Business? // Website builder. URL: <https://websitebuilder.org/resources/online-reviews-infographic> (дата обращения 13.04.2018)
7. **50 Important** Stats You Need to Know About Online Reviews [Infographic] // Vendasta Blog. URL: <https://www.vendasta.com/blog/50-stats-you-need-to-know-about-online-reviews> (дата обращения 13.04.2018)
8. **Социальные** сети в 2018 году: глобальное исследование // URL: https://vk.com/@academ_biz-socialnye-seti-v-2018-godu-globalnoe-issledovanie (дата обращения 15.08.18)
9. **Война** за внимание // РБК. 2018. № 7–8.

E. V. Artyushina, Candidate of Economics, Associate Professor, e-mail: eartushina@mail.ru,

D. E. Erschov, Candidate of Economics, e-mail: erschow@mail.ru,

National Research University Higher School of Economics (HSE), Nizhniy Novgorod, Russian Federation,

Yu. A. Prokhoda, e-mail: juprokhoda@gmail.com,

АО "Mediaskop", Moscow, Russian Federation

The Role of Internet Technologies in the Communication Activity of Medical Organization

The article contains key results of a consumer study that identifies the role of modern Internet technologies in the communication process for medical organizations. The results are obtained during the market research, which consisted of three parts and aimed at the formation of the structure of communication environment for medical organization with the assessment of its use effectiveness. The results of the performed consumers study for medical organization presents model that describes

the way of searching and selecting a medical centre by a customer. The model highlights 6 key groups of sources of information that are important for costumers. These are recommendations of existing users received in a personal conversation, by phone; recommendations of existing users received on forums, discussions in social networks; recommendations of doctors; website on the Internet; advertising (TV, radio, Newspapers, magazines), accommodation nearby. Also the model highlights 10 groups of parameters that influence the final choice of the medical organization. These are recommendations of existing users of the medical center, friends; recommendations of doctors; reputation of the medical center; convenient location; high level of service; price for services; specific specialist; qualification of doctors; uniqueness of the services provided; advertising. Taking into account the results of the study, the full map of the communication environment of the medical organization is given. The results of the study can be implemented by Russian medical organizations.

Keywords: internet technologies, communication activity, information, communication environment, medical organization

DOI: 10.17587/it.25.179-184

References

1. **Reilly M. J., Markenson D. S.** Health care emergency management: Principles and practice, Jones & Bartlett Publishers, 2010.
2. **Moser H. R., Freeman Jr G. L.** An empirical analysis of the public's attitudes toward advertising hospital services: A comparative cross-sectional study, *Health marketing quarterly*, 2014, vol. 31, no. 1, pp. 13–30.
3. **Smith K. T.** Hospital Marketing and Communications Via Social Media, *Services Marketing Quarterly*, 2017, vol. 38, no. 3, pp. 187–201.
4. **Issledovanie v sfere medicinskih uslug**, Calltouch.ru (Research in the field of medical services, Calltouch.ru.), available at: <https://blog.calltouch.ru/issledovanie-v-sfere-meditsinskih-uslug> (date of access 12.04.2018) (in Russian).
5. **Nanda S., Telang A., Bhatt G.** Hospital advertising: A literature review, *International Journal of Healthcare Management*, 2012, vol. 5, no. 1, pp. 28–31.
6. **How Online Reviews Can Either Make or Break Your Business?** Website builder, available at: <https://websitebuilder.org/resources/online-reviews-infographic> (date of access 13.04.2018)
7. **50 Important Stats You Need to Know About Online Reviews** [Infographic], Vendasta Blog, available at: <https://www.vendasta.com/blog/50-stats-you-need-to-know-about-online-reviews> (date of access 13.04.2018)
8. **Social'nye seti v 2018 godu: global'noe issledovanie** (Social media in 2018: a global study), available at: https://vk.com/@academ_biz-socialnye-seti-v-2018-godu-globalnoe-issledovanie (date of access 15.08.2018) (in Russian).
9. **Vojna za vnimanie** (The war for attention), *RBK*, 2018, no. 7–8 (in Russian).

ГЛАВНОЕ СОБЫТИЕ В ОБЛАСТИ ПРИБОРОСТРОЕНИЯ, ТОЧНЫХ ИЗМЕРЕНИЙ, МЕТРОЛОГИИ И ИСПЫТАНИЙ МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ ИННОВАЦИОННЫЙ ФОРУМ ТОЧНЫЕ ИЗМЕРЕНИЯ – ОСНОВА КАЧЕСТВА И БЕЗОПАСНОСТИ

Москва, 15-17 мая 2019 года
ВДНХ, Павильон №75

ОРГАНИЗАТОРЫ:



ВЫСТАВОЧНЫЕ РАЗДЕЛЫ

- METROEXPO**
МЕТРОЛОГИЯ, ИЗМЕРЕНИЯ И ИСПЫТАНИЯ
- CONTROL&DIAGNOSTIC**
КОНТРОЛЬ И ДИАГНОСТИКА
- LABTEST**
ЛАБОРАТОРНЫЕ ИСПЫТАНИЯ
- PROMAUTOMATIC**
ПРОМЫШЛЕННАЯ АВТОМАТИЗАЦИЯ
- RESMETERING**
УЧЁТ ЭНЕРГОРЕСУРСОВ
- WEIGHT SALON**
ВЕСОВОЙ САЛОН

СПЕЦИАЛИЗИРОВАННЫЕ
РАЗДЕЛЫ ВЫСТАВКИ

MetroExpo
Control&Diagnostic
ResMetering
LabTest
PromAutomatic

ВСЕРОССИЙСКИЙ
СЪЕЗД МЕТРОЛОГОВ
И ПРИБОРОСТРОИТЕЛЕЙ

ЦИФРЫ И ФАКТЫ 2018 года:

Участники – 296 компаний из 24 стран мира
Посетители – 5046 специалистов из 63 регионов России
Площадь экспозиции – 6870 м²

ДИРЕКЦИЯ ФОРУМА

Тел./Факс: +7 (495) 937-40-23
E-mail: metrol@expoprom.ru

Спешите забронировать стенд www.metrol.expoprom.ru

Ю. Д. Гвоздкова¹, аспирант, e-mail: gina94@yandex.ru,

И. А. Гвоздкова², канд. физ.-мат. наук, доц., зав. каф., проф., e-mail: gvozdkova@yandex.ru,

А. В. Курочкин², канд. физ.-мат. наук, ст. науч. сотр., доц., e-mail: avkur2@yandex.ru,

А. В. Черняев¹, д-р техн. наук, проф., e-mail: alex_v_ch@bk.ru,

¹ФГБОУ ВО "Московский авиационный институт

(национальный исследовательский университет)", г. Москва

²ОУП ВО "Академия труда и социальных отношений", г. Москва

Информационная система оценки экологической безопасности авиационных материалов и технологий методом анализа иерархий

Рассматриваются принципы разработки, результаты апробации и рекомендации по развитию автоматизированной информационной системы (АИС) многокритериального анализа экологических преимуществ авиационных технологий и материалов на основе метода анализа иерархий (МАИ). Обосновываются использование наиболее простого алгоритма МАИ для решения поставленной оптимизационной задачи и выбор соответствующего программного обеспечения для его реализации. Излагаются результаты вычислительного эксперимента, выполненного с помощью разработанной АИС в целях изучения возможностей замещения традиционного авиационного топлива альтернативными источниками энергии.

Ключевые слова: информационная система, экологическая и техносферная безопасность, авиационные технологии и материалы, метод анализа иерархий, программное обеспечение, многокритериальная задача, источники энергии

Введение

Одной из актуальных проблем в сфере развития современных авиационных технологий и материалов является обеспечение их экологической безопасности. Для оценки уровня обеспечения социо-эколого-экономической и техносферной безопасности необходима высокоэффективная система показателей устойчивого эколого-технологического, эколого-экономического и социально-экологического развития, позволяющая осуществлять комплексное сравнение изучаемых объектов и процессов [1, 2]. Реализация подобной системы требует развития соответствующих алгоритмов и информационных технологий. В работах [3–5] была предложена методика многокритериального сравнительного анализа эколого-экономических и эколого-технических преимуществ источников энергии для авиатранспорта и материалов для авиационных энергоустановок на основе одного из основных методов математической оптимизации — метода анализа иерархий (МАИ).

При использовании МАИ в задачах принятия решений с большим числом критериев оценки нахождение оптимального варианта становится трудоемким, поскольку требует громоздких вычислений [6–8]. Поэтому в указанных случаях следует применять компьютерное моделирование поиска наилучшего решения [9–11]. В данной статье излагаются основные принципы разработки, результаты апробации и рекомендации по развитию автоматизированной информационной системы многокритериального сравнительного анализа экологических преимуществ авиационных технологий и материалов на основе метода анализа иерархий.

1. Постановка многокритериальной задачи о безопасности авиационных технологий и материалов

Для поиска оптимальных решений разрабатываются и используются многофункциональные автоматизированные информационные

системы (АИС) сбора, передачи и переработки информации об изучаемых процессах и объектах на основе математических методов [11, 12]. Изучение традиционных и перспективных материалов и технологий для летательных аппаратов (ЛА) показало, что выбор наиболее безопасных из них может быть осуществлен с помощью специальной АИС, включающей блок данных, фор-

мируемый путем сбора сведений об исследуемых объектах; алгоритм их сравнительного анализа на основе выбранных критериев оценки; математическую модель нахождения оптимального решения соответствующей многокритериальной задачи; программное обеспечение (ПО) реализации указанной модели; блок рекомендаций, формулируемых на основе анализа результатов,

Таблица 1

Критерии оценки эколого-технических и эколого-экономических характеристик авиационных материалов и технологий *

Категория	Наименование критериев
Экологические критерии	<p><i>Технологии:</i> возобновляемость ресурсов для технологического процесса (ТП); безопасность разработки и использования технологии (Т); возможность утилизации и обезвреживания отходов в ТП; возможность замещения традиционных ископаемых энергоносителей при использовании Т; объемы выбросов загрязняющих веществ при использовании Т; сокращение выбросов парниковых газов и других загрязнителей окружающей среды (ОС) при использовании Т; уровень шума двигателя ЛА; расход пресной воды в ТП; ресурсосбережение и возможность осуществлять рациональное природопользование; удовлетворение Т требованиям экологических стандартов; снижение ущерба, наносимого ОС, при эксплуатации Т; улучшение здоровья людей и климата при использовании Т и т. д.</p> <p><i>Материалы:</i> возобновляемость ресурсов для ТП получения материала (М); безопасность получения и использования М; возможность утилизации и обезвреживания отходов в ТП получения М; объемы выбросов загрязняющих веществ при получении М; сокращение выбросов парниковых газов и других загрязнителей ОС при использовании М; расход пресной воды в ТП получения и использования М; ресурсосбережение; удовлетворение М требованиям экологических стандартов; снижение ущерба, наносимого ОС, при использовании М; улучшение здоровья людей и климата при использовании М и т. д.</p>
Технические критерии	<p><i>Технологии:</i> обеспеченность ТП сырьем; разнообразие сырьевой базы для ТП; качество сырья для ТП; возможность бесперебойного снабжения ТП сырьем; удобство транспортировки сырья для ТП; степень освоенности Т и ее доступность; наличие серийного производства оборудования для ТП; возможность полной комплектации оборудования для ТП на территории страны или региона; КПД ТП; КПД двигателя ЛА, использующего ТП; возможность использования ТП в авиационной технике большой мощности; возможность использования ТП на различных стадиях полета; разнообразие ЛА, в которых можно использовать ТП; масса и габаритные размеры двигателя, использующего ТП; необходимость использования больших площадей и объемов для реализации ТП; необходимость использования большого количества вспомогательной техники; расход энергии в ТП; срок службы двигателя ЛА, использующего ТП; зависимость эффективности работы двигателя ЛА от погодных условий и климата; возможность обеспечения высокой надежности работы двигателя и систем бортового питания; способность обеспечивать непрерывное энергоснабжение двигателя и систем бортового питания; возможность прогнозировать и планировать переработку энергии в ТП; воспламеняемость ТП; наличие научно-технических разработок для развития Т; наличие научно-технологического сотрудничества с зарубежными партнерами; удовлетворение Т требованиям технологических стандартов и т. д.</p> <p><i>Материалы:</i> обеспеченность ТП получения М сырьем; затраты энергии и степень освоенности и доступности Т получения М; долговечность М; возможность использования М в различных авиационных энергоустановках; масса и габаритные размеры энергоустановок, изготовленных из М; возможность обеспечения высокой надежности работы энергоустановки, использующей М; воспламеняемость М; наличие научно-технических разработок для развития Т получения М; удовлетворение М требованиям технологических стандартов и т. д.</p>
Экономические критерии	<p><i>Технологии:</i> стоимость сырья и оборудования для ТП; стоимость двигателя, использующего ТП; себестоимость ТП; экономичность ТП; наличие инфраструктуры для использования ТП в ЛА; наличие спроса на технологических рынках топлива и энергии; отсутствие зависимости ТП от импорта сырья и/или оборудования; увеличение экспорта традиционных энергоносителей при использовании ТП; снижение зависимости от импорта традиционных энергоресурсов при использовании Т; независимость ТП от цен на топливо и его поставок; укрепление позиций на мировых энергетических и технологических рынках и выход на новые рынки при использовании ТП; наличие законодательных и финансовых механизмов поддержки использования Т; увеличение ВВП и налоговых поступлений в бюджеты различных уровней при использовании Т и т. д.</p> <p><i>Материалы:</i> стоимость М и установок на его основе; наличие инфраструктуры для производства и эксплуатации установок на основе М; наличие спроса на рынках материалов, оборудования и технологий; укрепление позиций на мировых энергетических и технологических рынках при реализации ТП получения М и производства установок на его основе; наличие законодательных и финансовых механизмов поддержки ТП получения и использования М и установок на его основе; увеличение ВВП и налоговых поступлений в бюджеты различных уровней при реализации ТП получения и использования М и т. д.</p>
* Источники: [3, 5], разработка авторов.	

полученных в ходе компьютерного моделирования процесса принятия наилучшего решения.

Методика комплексного многофакторного анализа экологических преимуществ и недостатков рассматриваемых технологий и материалов может быть разработана на основе различных математических подходов к принятию оптимальных решений, в том числе — и на основе их синтеза. Одним из ключевых элементов такой методики должна быть совокупность используемых критериев оценки. Поскольку помимо экологических характеристик производителей и потребителей интересуют и другие особенности сравниваемых объектов, указанную совокупность целесообразно разделить на три категории — экологические, технические и экономические критерии [3–5]. В табл. 1 перечислены критерии, которые предлагается использовать в задаче об эколого-технической и эколого-экономической безопасности технологий и материалов для ЛА.

Рейтинг значимости предлагаемых критериев определяется различными аспектами: технологическими, региональными, особенностями предпочтений лиц, принимающих решения (ЛПР) и т. д.

Процесс сравнительного анализа эколого-технических и эколого-экономических характеристик материалов и технологий, используемых в авиации, рекомендуется осуществлять в следующей последовательности: 1) присвоить критериям оценки рейтинг значимости и отобрать для дальнейшего анализа критерии с наиболее высоким рейтингом; 2) провести оценку изучаемых процессов и/или объектов с помощью выбранных критериев; 3) сравнить их с учетом приоритетности критериев на основе одного или нескольких оптимизационных математических методов.

2. Методы решения многокритериальной задачи о безопасности авиационных технологий и материалов

Для решения сформулированной задачи могут быть применены методы функционально-стоимостного анализа (ФСА, Activity Based Costing, ABC-анализ) [13–15], балльной оценки [16], анализ Парето [2, 17] и МАИ [2, 6, 7, 18]. Указанные оптимизационные подходы рекомендуется реализовывать в виде специальных компьютерных программ в целях повышения эффективности вычислительных процедур в разработанной ИС.

МАИ основан на структуризации проблемы выбора в виде совокупности иерархических

уровней (например, уровня цели (верхний уровень), уровня экспертов, уровня критериев оценки, уровня альтернативных решений (нижний уровень)).

Для выбора наиболее оптимальных авиационных технологий и материалов может быть использован вариант МАИ, включающий наряду с уровнем цели два других иерархических уровня — уровень альтернативных решений и уровень критериев их оценки [2, 4, 5]. Тогда решение задачи об определении степени их экологической безопасности с учетом требований к экономической и технологической эффективности рассматриваемых объектов исследования будет включать следующие этапы [2, 4]: 1) присвоение критериям оценки рейтинга значимости и отбор критериев с наиболее высоким приоритетом с учетом региональных, национальных и/или технологических предпочтений и ограничений; 2) сравнение технологий и материалов на основе выбранных экологических, технических и экономических критериев путем вычисления их безразмерных нормированных показателей оценки по каждому критерию; 3) вычисление итоговых (глобальных) весовых коэффициентов изучаемых альтернатив путем свертки их нормированных показателей оценки по критериям и индексов значимости критериев, измеряемых по шкале от 0 до 1; 4) выбор наиболее оптимального из альтернативных решений (с самым высоким итоговым весовым коэффициентом).

В соответствии с типовым алгоритмом МАИ для выбора решения из P альтернативных вариантов на основе N критериев вначале определяют абсолютные значения показателей оценки решений по всем критериям $Q(i, k)$ ($i = 1, 2, \dots, N$, $k = 1, 2, \dots, P$). Затем ЛПР с учетом своих предпочтений вводит абсолютный (ненормированный) рейтинг значимости каждого критерия оценки $R(i)$. Далее вычисляют нормированные значения $R(i)$ и $Q(i, k)$, измеряемые по шкале от 0 до 1 и обозначаемые соответственно $r(i)$ и $q(i, k)$:

$$r(i) = \frac{R(i)}{\sum_{i=1}^N R(i)}; \quad (1)$$

$$q(i, k) = \frac{Q^m(i, k)}{\sum_{k=1}^P Q^m(i, k)}, \quad i = 1, 2, \dots, N, \quad k = 1, 2, \dots, P, \quad (2)$$

где m — параметр, равный 1 или -1 и определяющий, какое значение показателя оценки решения по критерию является наиболее ценным: минимальное или максимальное (например, для выбросов в ОС $m = -1$, а для КПД $m = 1$).

Глобальные (суммарные, или комбинированные) весовые коэффициенты решений $V(k)$ находят по формуле

$$V(k) = \sum_{i=1}^N r(i)q(i, k), k = 1, 2, \dots, P. \quad (3)$$

Значения суммарных весовых коэффициентов $V(k)$ принадлежат интервалу от 0 до 1, а решение с наивысшим $V(k)$ — оптимальное.

Для решения задач с использованием МАИ разработаны различные программные продукты, требующие применения подходов матричной алгебры и достаточно высокой квалификации пользователей (в некоторых вариантах программной реализации рассматриваемого метода предусмотрена возможность ввода мнений экспертов с помощью устройств удаленного введения и обработки оценок). Выбор оптимальных решений по данному методу может быть осуществлен, например, с помощью программы "MPRIORITY 1.0" [9] и других электронных ресурсов. В некоторых работах предлагается реализация МАИ с использованием только библиотечных функций MS Excel, однако такой подход не обеспечивает полной автоматизации вычислений и не позволяет проводить анализ устойчивости полученных результатов [10]. Если в оптимизационных задачах используются однозначные количественные и/или качественные критерии оценки, то матрицы сравнения элементов иерархии можно не вводить [11, 12]. Тогда вычислительные процедуры значительно упрощаются.

В рассматриваемой АИС используется компьютерная программа, предназначенная для реализации наиболее простого алгоритма МАИ, основанного на расчетах весовых коэффициентов по формулам (1)–(3), адекватного поставленным задачам и не требующего применения методики парных сравнений для определения рейтинга значимости элементов иерархической структуры. Программа написана на языке программирования VBA (Visual Basic for Application) для приложения MS Excel 2010.

3. Вычислительный эксперимент

Апробация разработанной ИС была осуществлена для источников энергии, которые в настоящее время либо активно используют в различных авиатранспортных средствах, либо рассматривают возможности замещения ими традиционного авиационного топлива. Комплексный сравнительный анализ их преимуществ и недостатков в сфере обеспечения техносферной и экологической безопасности

позволил сформировать базу данных о ключевых критериях оценки изучаемых объектов с указанием рейтинга значимости каждого из них и о показателях оценки источников энергии по критериям.

На рис. 1 (см. третью сторону обложки) изображено рабочее окно компьютерной программы АИС, в котором приведены характеристики ИЭ по критериям, показатели абсолютной значимости критериев по шкале от 0 до 10 с учетом задач по обеспечению экологической и техносферной безопасности и результаты расчетов итоговых весовых коэффициентов оцениваемых объектов методом анализа иерархий.

Перед началом работы с программой следует заполнить соответствующие столбцы таблицы наименованиями критериев оценки источников энергии, указав для каждого критерия параметр m и весовой коэффициент, определяющий его абсолютную значимость. Далее следует ввести абсолютные значения показателей оценки источников энергии (ИЭ) по каждому из критериев, порядковый номер критериев определяется автоматически. В случае некорректного ввода информации в ходе работы программы будут появляться предупреждающие сообщения. Следует отметить, что если для критерия $m = -1$, то абсолютные значения показателей оценки решений по нему $Q(i, k)$ согласно формуле (2) не должны быть равны 0. В программе предусмотрена защита вычислений в случае некорректного ввода нулевых значений $Q(i, k)$ при $m = -1$: при этом вместо 0 автоматически вводится очень маленькое положительное число.

В пользовательском интерфейсе предусмотрены пять кнопок управления программным продуктом. После нажатия на кнопку "Пуск" появляются диалоговые окна, в которых необходимо задать число источников энергии и число критериев оценки (техническое ограничение для их числа в данном варианте программы — 100). Далее в автоматическом режиме проводятся необходимые вычисления, и рассчитанные суммарные весовые коэффициенты ИЭ отображаются в правых верхних ячейках таблицы. Рядом с кнопкой "Пуск" имеются вспомогательные кнопки, предназначенные для создания резервной копии данных на отдельном листе Excel ("Резерв"), для восстановления данных ("Из резерва") и две кнопки очистки (очистка только последних введенных значений ("Очист. знач.") и полная очистка таблицы ("Очист. все"). Основные отличия разработанной программы от имеющихся аналогов заключаются в реализации в ней полной автоматизации процесса вычислений и в отсутствии требований к наличию у поль-

<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">ПУСК</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">В резерв</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Из резерва</div> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">RC</div> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">AC</div> </div>							0,438	0,562
№	Критерии оценки источников энергии	Параметр <i>m</i> (если лучший выбор — "минимум для абс. значения", то -1, иначе 1)	Весовой коэф. критерия (от до 10)	Норм. весовой коэф. критерия	Виды источников энергии			
					Керосин		Солнечные батареи	
					Абс. знач.	Вес. коэф.	Абс. знач.	Вес. коэф.
1	Критерий 1	1	10	0,0375	0,1	0,0099	10	0,9901
2	Критерий 2	1	5	0,0187	10	0,6667	5	0,3333
3	Критерий 3	1	10	0,0375	10	0,5000	10	0,5000
4	Критерий 4	1	5	0,0187	2	0,1818	9	0,8182
5	Критерий 5	1	7	0,0262	0	0,0000	10	1,0000
6	Критерий 6	1	5	0,0187	10	0,9091	1	0,0909
7	Критерий 7	1	5	0,0187	10	0,7692	3	0,2308
8	Критерий 8	1	10	0,0375	2,5	0,6250	1,5	0,3750
9	Критерий 9	1	10	0,0375	10	0,9995	0,0005	0,0005
10	Критерий 10	1	10	0,0375	10	0,6667	5	0,3333
11	Критерий 11	1	10	0,0375	10	0,6667	5	0,3333
12	Критерий 12	1	10	0,0375	10	0,6667	5	0,3333
13	Критерий 13	-1	7	0,0262	1	0,8333	5	0,1667
14	Критерий 14	-1	7	0,0262	1	0,8333	5	0,1667
15	Критерий 15	1	8	0,0300	2,5	0,4545	3	0,5455
16	Критерий 16	-1	10	0,0375	1	0,9091	10	0,0909
17	Критерий 17	1	10	0,0375	5	0,3333	10	0,6667
18	Критерий 18	1	10	0,0375	8	0,6154	5	0,3846
19	Критерий 19	1	10	0,0375	8	0,6154	5	0,3846
20	Критерий 20	-1	8	0,0300	5	0,1667	1	0,8333
21	Критерий 21	1	5	0,0187	10	0,6667	5	0,3333
22	Критерий 22	1	10	0,0375	0	0	10	1
23	Критерий 23	1	10	0,0375	1	0,1	9	0,9
24	Критерий 24	-1	10	0,0375	10	1E-04	0,001	0,9999
25	Критерий 25	1	10	0,0375	0	0	10	1
26	Критерий 26	-1	5	0,0187	8	0,1111	1	0,8889
27	Критерий 27	1	5	0,0187	1	0,1	9	0,9
28	Критерий 28	1	10	0,0375	5	0,3333	10	0,6667
29	Критерий 29	-1	10	0,0375	1	0,8333	5	0,1667
30	Критерий 30	1	5	0,0187	0	0	10	1
31	Критерий 31	1	5	0,0187	0	0	10	1
32	Критерий 32	1	5	0,0187	10	0,9091	1	0,0909
33	Критерий 33	1	10	0,0375	0	0	10	1

Рис. 2. Результаты расчетов итоговых весовых коэффициентов керосина (0,438) и солнечных батарей (0,562) с помощью АИС, предназначенной для оценки эколого-экономических и эколого-технических преимуществ источников энергии для ЛА по МАИ

зователей знаний основ линейной алгебры и особенностей работы со встроенными функциями табличного процессора Excel.

На рис. 2 представлены результаты сравнительной оценки эколого-экономических и эколого-технических преимуществ использования керосина и солнечных батарей в небольших турбовинтовых самолетах, полученные с помощью разработанной АИС. Для анализа были использованы наиболее значимые для РФ критерии оценки, перечень которых указан в табл. 2.

Для МАИ, как и для других оптимизационных методов, важным критерием оценки найденного решения является его устойчивость при изменениях вводимых параметров, которые могут быть неточно заданы или подвержены влиянию субъективных факторов. Поэтому для проверки устойчивости полученных результатов было проведено 20 разыгрываний исходных данных модели на основе метода статистических испытаний, реализованного с помощью встроенного в VBA генератора случайных чисел [19–21]. Все вводимые параметры варьировались случайным образом в пределах 30 % от исход-

Таблица 2

Критерии оценки эколого-экономических и эколого-технических преимуществ использования керосина и фотоэлементов в небольших турбовинтовых самолетах*

№ критерия	Наименование
1	Обеспеченность сырьем для производства источника энергии
2	Отсутствие зависимости производства ИЭ от импорта сырья и/или оборудования
3	Возможность бесперебойного снабжения производства ИЭ сырьем
4	Удобство транспортировки источника энергии
5	Возможность замещения традиционных ископаемых энергоносителей при использовании источника энергии
6	Степень освоенности технологии производства и/или преобразования энергии источника и ее доступность
7	Наличие серийного производства оборудования для преобразования энергии источника
8	КПД процесса переработки энергии источника
9	Возможность использования ИЭ в технике большой мощности
10	Возможность использования источника энергии для энергоснабжения двигателя ЛА и систем бортового питания
11	Возможность использования ИЭ на различных стадиях полета
12	Разнообразии ЛА, в которых можно использовать источник энергии
13	Необходимость использования больших площадей и объемов для преобразования энергии источника
14	Необходимость использования большого количества вспомогательной техники при эксплуатации ИЭ
15	Срок службы двигателя ЛА, использующего источник энергии
16	Зависимость эффективности работы двигателя ЛА от погодных условий и климата
17	Возможность обеспечения высокой надежности работы двигателя и систем бортового питания

№ критерия	Наименование
18	Способность обеспечивать непрерывное энергоснабжение двигателя и систем бортового питания
19	Возможность прогнозировать и планировать переработку энергии источника
20	Воспламеняемость
21	Наличие научно-технических разработок для развития технологии преобразования энергии источника
22	Возобновляемость ресурсов для получения и/или эксплуатации ИЭ
23	Безопасность производства и использования источника энергии
24	Объемы выбросов загрязняющих веществ при использовании ИЭ
25	Сокращение выбросов парниковых газов и других загрязнителей ОС при использовании источника энергии
26	Уровень шума двигателя, использующего источник энергии
27	Ресурсосбережение и возможность осуществлять рациональное природопользование
28	Доступность источника энергии
29	Стоимость сырья и оборудования для производства источника энергии
30	Рост экспорта традиционных энергоресурсов при использовании ИЭ
31	Независимость от цен на топливо и его поставок
32	Наличие законодательных и финансовых механизмов поддержки производства и использования источника энергии
33	Снижение ущерба, наносимого ОС

* Источник: разработано авторами.

ного значения (в предположении равномерного распределения их возможных значений и при верхнем ограничении значений, равном 10). В результате указанной обработки были получены следующие результаты для разброса значений итоговых весовых коэффициентов керосина и солнечных батарей: $0,435 \pm 0,012$ и $0,565 \pm 0,012$ (при разыгрывании только весовых коэффициентов критериев); $0,422 \pm 0,022$ и $0,578 \pm 0,022$ (при разыгрывании не только рейтинга значимости критериев, но и абсолютных значений показателей оценки ИЭ по критериям). Погрешность результатов оценивалась по выборочным стандартным отклонениям. Полученные распределения результатов, как и следовало ожидать, оказались близкими к нормальным. Таким образом, было установлено, что первоначально полученные значения суммарных весовых коэффициентов сравниваемых ИЭ, равные 0,438 и 0,562, обладают достаточно высокой статистической достоверностью и устойчивостью.

Заключение

Особенность методики, используемой в рассмотренной АИС, заключается в сочетании в ней универсальности и предметной ориентированности. Суммарные весовые коэффициенты учитывают достоинства и недостатки авиационных технологий и материалов, и в каждом конкретном случае можно взять для расчетов наиболее важные с точки зрения экологической безопасности критерии, чтобы принять

научно обоснованное решение о том, как оптимально использовать рассматриваемые объекты исследования с учетом их конкурентных эколого-экономических и эколого-технических преимуществ. Методика позволяет осуществлять комплексную оценку технологий и материалов для авиатранспортных средств и рекомендуется для широкого практического применения.

Усовершенствование разработанной АИС можно осуществлять как в направлении увеличения числа иерархических уровней в используемом МАИ (например, целесообразно ввести в математическую модель уровень экспертов и модернизировать соответствующее ПО для ее реализации), так и в направлении расширения совокупности методов решения многокритериальной задачи о безопасности технологий и материалов для ЛА.

Список литературы

1. Сосунова И. А. Методология и методы современной социальной экологии. М.: МНЭПУ, 2010.
2. Гвоздкова И. А. Математические задачи социо-эколого-ориентированной экономики // Труд и социальные отношения. 2017. № 6. С. 73–84.
3. Гвоздкова И. А., Гвоздкова Ю. Д. Методика оценки социо-эколого-экономических преимуществ источников энергии для авиатранспорта // Экономика в условиях социально-техногенного развития мира. Матер. II Междунар. междисциплинарной науч. конф. по фундаментальным и прикладным проблемам современного социально-экономического и экономико-экологического развития (5 октября 2017 г., г. Брянск). Брянск: БГТУ, 2017. Т. 2. С. 130–134.
4. Гвоздкова Ю. Д. Многокритериальная оценка перспективных источников энергии для авиатранспорта методом анализа иерархий // 16-я Междунар. конф. "Авиация и космонавтика — 2017". 20–24 ноября 2017 г. М., 2017. С. 373–374.

5. **Гвоздкова Ю. Д.** Комплексная оценка эколого-технических преимуществ материалов для авиационных энергоустановок на основе альтернативных источников энергии // Гагаринские чтения — 2018. Сб. тезисов докл. XLIV Междунар. молодежной науч. конф. М.: Изд. МАИ (НИУ), 2018. Т. 3. С. 87—88.
6. **Саати Т. Л.** Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1993. 316 с.
7. **Саати Т. Л.** Принятие решений при зависимостях и обратных связях: Аналитические сети. М.: Издательство ЛКИ, 2008. 360 с.
8. **Гвоздкова И. А.** Эколого-ориентированные математические модели управления рисками и обеспечением безопасности // Вестник университета. М.: ГУУ. 2016. № 9. С. 200—207.
9. **Абакаров А. Ш., Сушков Ю. А.** Программная система поддержки принятия решений "MPRIORITY 1.0" // Электронный научный журнал "Исследовано в России", 2005. URL: <http://www.elibrary.lt/resursai/Uzsienio%20leidiniai/MFTI/2005/207.pdf> (дата обращения: 17.02.2018).
10. **Шевченко Д. В.** Метод анализа иерархий. Материалы лекции. URL: www.ieml-math.narod.ru/lect/MPUR_MAI.pdf (дата обращения: 25.05.2018).
11. **Гвоздкова И. А., Курочкин А. В.** Программная реализация математических методов в задачах оптимизации выбора безопасных продуктов питания // Экономика в условиях социально-технологического развития мира. Матер. II Междунар. междисциплинарной науч. конф. по фундаментальным и прикладным проблемам современного социально-экономического и экономико-экологического развития (5 октября 2017 г., г. Брянск). Брянск: Изд. БГТУ, 2017. Т. 1. С. 145—152.
12. **Гвоздкова И. А., Курочкин А. В.** Математическое и информационное обеспечение дистанционной занятости // Труд и социальные отношения. 2017. № 5. С. 32—44.
13. **Fang L., Xianyi G., Xin M.** On reasons of uncontrollable construction cost based on the ABC analysis method and the complete decomposition model // Systems Engineering Procedia. 2012. V. 4. P. 359—365. URL: <https://www.researchgate.net/publication/271609803> (дата обращения: 08.06.2018).
14. **Yu M.-C.** Multi-criteria ABC analysis using artificial-intelligence-based classification techniques // Expert Systems with Applications. 2011. V. 38. Iss. 4. P. 3416—3421. URL: <https://www.sciencedirect.com/science/article/pii/S0957417410009358> (дата обращения: 08.06.2018).
15. **Шатунова Г. А., Кузьмина О. Н.** Историко-логический генезис и периодизация этапов развития функционально-стоимостного анализа // Вестник Самарского государственного экономического университета. 2012. № 4 (90). С. 91—96.
16. **Белов В. М.** Метод балльной оценки показателей коэффициентов весомости // Вестник ФГОУ ВПО МГАУ. 2009. № 4. С. 15—19.
17. **Шубин И. И.** Использование метода Парето для анализа и отбора инвестиционных проектов для финансирования // Матер. Междунар. научно-практ. конф. "Актуальные проблемы управления — 2006". М.: ГУУ, 2006. С. 244—247.
18. **Saaty T. L.** Relative measurement and its generalization in decision making: why pairwise comparisons are central in mathematics for the measurement of intangible factors — the analytic hierarchy/network process // RACSAM (Review of the Royal Spanish Academy of Sciences, Series A, Mathematics). 2008. N. 102 (2). P. 251—318.
19. **Кузьменко В. Г.** VBA. Эффективное использование. М.: БИНОМ, 2015.
20. **Операция** перемешивания значений — VB. URL: <http://www.cyberforum.ru/vb-net/thread359942.html> (дата обращения: 11.02.2018).
21. **L'Ecuyer P.** Random number generation // Springer Handbooks of Computational Statistics. 2007. P. 93—137.

J. D. Gvozdikova¹, Postgraduate Student, e-mail: gina94@yandex.ru,

I. A. Gvozdikova², Head of the Department, Professor, e-mail: gvozdikova@yandex.ru,

A. V. Kurochkin², Assistant Professor, e-mail: avkur2@yandex.ru,

A. V. Chernyaev¹, Professor, e-mail: alex_v_ch@bk.ru,

¹Moscow Aviation Institute (National Research University), Moscow

²Academy of labor and social relations, Moscow

Information System of Environmental Safety Assessment of Aviation Materials and Technologies by the Analytic Hierarchy Process

The paper considers design principles, results of testing and recommendations for the development of the automated information system (AIS) using the multi-criteria analysis of environmental benefits of aircraft technologies and materials based on the analytic hierarchy process (AHP). The use of simple AHP algorithm to solve the optimization problem and the choice of the appropriate software for its implementation is substantiated. The results of a computational experiment carried out with the help of the developed AIS to study the possibility of replacing traditional aviation fuel with alternative energy sources are presented.

Keywords: information system, environmental and technosphere safety, aviation technologies and materials, analytic hierarchy process, software, multi-criteria task, energy sources

DOI: 10.17587/it.25.185-192

References

1. **Sosunova I. A.** *Metodologija i metody sovremennoj social'noj ekologii* (Methodology and methods of modern social ecology), Moscow, MNEPU, 2010 (in Russian).

2. **Gvozdikova I. A.** *Matematicheskiye zadachi socio-ekologo-orientirovannoj ekonomiki* (Mathematical problems of socially and ecologically oriented economy), *Trud i Social'nyje Otnoshenija*. 2017, no. 6, pp. 73—84 (in Russian).

3. **Gvozdikova I. A., Gvozdikova J. D.** *Metodika ocenki socio-ekologo-ekonomicheskikh preimushchestv istochnikov energii dlya aviatransporta* (Methods of assessment of ecological and socially economic benefits of aircraft energy sources), *Ekonomika v usloviyah social'no-tehnogenogo razvitiya mira. Materialy II Mezhdunarodnoj mezhdisciplinarnoj nauchnoj konferencii po fundamental'nym i prikladnym problemam sovremennoogo social'no-ekonomicheskogo i ekonomiko-ekologicheskogo razvitiya* (Economy in terms of socio-technological development of the world. Proceedings of the II In-

ternational interdisciplinary scientific conference on fundamental and applied problems of modern socio-economic and environmental development) (5 oktyabrya 2017 g., g. Bryansk), Bryansk, BGTU, 2017, vol. 2, pp. 130–134 (in Russian).

4. **Gvozdikova J. D.** *Mnogokriterial'naja ocenka perspektivnykh istochnikov ehnergii dlya aviatransporta metodom analiza ijerarhij* (Multi-criteria assessment of promising energy sources for air transport by the analytic hierarchy process), *16-ja Mezhdunarodnaja konferencija "Aviacija i kosmonavtika — 2017"*, 20–24 november 2017, Moscow, Tezisy. Tipografija "Lyuksor", 2017, pp. 373–374 (in Russian).

5. **Gvozdikova J. D.** *Kompleksnaja ocenka ekologo-tehnicheskikh preimushchestv materialov dlya aviacionnykh ehnergoustanovok na osnove al'ternativnykh istochnikov ehnergii* (Complex assessment of ecological and technical advantages of materials for aircraft power plants based on alternative energy sources), *Gagarinskije chtenija — 2018. Sbornik tezisev dokladov XLIV Mezhdunarodnoj molodyozhnoj nauchnoj konferencii, Moscow, Moskovskij aviacionnyj institut (nacional'nyj issledovatel'skij universitet)*, 2018, vol. 3, pp. 87–88 (in Russian).

6. **Saati T. L.** *Prinyatije reshenij. Metod analiza ijerarhij* (Decision — making. Analytic hierarchy process), Moscow, Radio i svyaz', 1993, 316 p. (in Russian).

7. **Saati T. L.** *Prinyatije reshenij pri zavisimostyah i obratnykh svyazyah: Analiticheskie seti* (Decision — making at dependences and feedbacks: Analytical networks), Moscow, Izdatel'stvo LKI, 2008, 360 p. (in Russian).

8. **Gvozdikova I. A.** *Ekologo-orientirovannye matematicheskie modeli upravlenija riskami i obespechenijem bezopasnosti* (Ecological-oriented mathematical models of risk management and security), *Vestnik universiteta, Moscow, GUU*, 2016, no. 9, pp. 200–207 (in Russian).

9. **Abakarov A. Sh., Sushkov Yu. A.** *Programmnaja sistema podderzhki prinyatija reshenij "MPRIORITY 1.0"* (Software decision support system "MPRIORITY 1.0"), *Elektronnyj nauchnyj zhurnal "Issledovano v Rossii"*, 2005, available at: <http://www.elibrary.lt/resursai/Uzsienio%20leidiniai/MFTI/2005/207.pdf> (date of access: 17.02.2018) (in Russian).

10. **Shevchenko D. V.** *Metod analiza ijerarhij. Materialy lek-cii* (Analytic hierarchy process. Lecture materials), available at: www.ieml-math.narod.ru/lect/MPUR_MAI.pdf (date of access: 25.05.2018) (in Russian).

11. **Gvozdikova I. A., Kurochkin A. V.** *Programmnaya realizacija matematicheskikh metodov v zadachah optimizacii vybora bezopasnykh produktov pitanija* (Software implementation of mathematical methods in optimizing the choice of safe food), *Ekonomika v usloviyah social'no-tehnogennogo razvitiya mira. Materialy II Mezhdunarodnoj mezhdisciplinarnoj nauchnoj konferencii po fundamental'nyh i prikladnym problemam sovremennogo social'no-ekonomicheskogo i*

ekonomiko-ekologicheskogo razvitiya (Economy in terms of socio-technological development of the world. Proceedings of the II International interdisciplinary scientific conference on fundamental and applied problems of modern socio-economic and environmental development) (5 oktyabrya 2017 g., g. Bryansk), Bryansk, BGTU, 2017, vol. 1, pp. 145–152 (in Russian).

12. **Gvozdikova I. A., Kurochkin A. V.** *Matematicheskoje i informacionnoje obespechenije distancionnoj zanyatosti* (Mathematical and information means of remote employment), *Trud i social'nyje otnoshenija*. 2017, vol. 5, pp. 32–44 (in Russian).

13. **Fang L., Xianyi G., Xin M.** On reasons of uncontrollable construction cost based on the ABC analysis method and the complete decomposition model, *Systems Engineering Procedia*. 2012, vol. 4, pp. 359–365, available at: <https://www.researchgate.net/publication/271609803> (date of access: 08.06.2018).

14. **Yu M.-C.** Multi-criteria ABC analysis using artificial-intelligence-based classification techniques, *Expert Systems with Applications*. 2011, v. 38, iss. 4, pp. 3416–3421, available at: <https://www.sciencedirect.com/science/article/pii/S0957417410009358> (date of access: 08.06.2018).

15. **Shatunova G. A., Kuz'mina O. N.** *Istoriko-logicheskij genezis i periodizacija etapov razvitiija funkcional'no-stoimostnogo analiza* (Historical and logical genesis and periodization of development stages of functional and cost analysis), *Vestnik Samarskogo gosudarstvennogo ekonomicheskogo universiteta*, 2012, no. 4 (90), pp. 91–96 (in Russian).

16. **Belov V. M.** *Metod ball'noj ocenki pokazatelej koefitsientov vesomosti* (Method of the weight coefficients scoring), *Vestnik FGOU VPO MGAU*, 2009, no. 4, pp. 15–19 (in Russian).

17. **Shubin I. I.** *Ispol'zovanie metoda Pareto dlya analiza i otbora investicionnykh proektov dlya finansirovaniya* (The use of Pareto method for analysis and selection of investment projects for financing), *Materialy mezhdunarodnoj nauchno-prakticheskoy konferencii "Aktual'nyje problemy upravlenija — 2006"*, Moscow, GUU, 2006, pp. 244–247 (in Russian).

18. **Saati T. L.** Relative measurement and its generalization in decision making: why pairwise comparisons are central in mathematics for the measurement of intangible factors — the analytic hierarchy/network process, *RACSAM (Review of the Royal Spanish Academy of Sciences, Series A, Mathematics)*, 2008, vol. 102 (2), pp. 251–318.

19. **Kuz'menko V. G.** *VBA. Effektivnoje ispol'zovanie (VBA. Effective using)*, Moscow, BINOM, 2015 (in Russian).

20. **Operacija** peremeshivaniya znachenij — VB (Value mixing operation — VB), available at: <http://www.cyberforum.ru/vb-net/thread359942.html> (date of access: 11.02.2018) (in Russian).

21. **L'Ecuyer P.** Random number generation, *Springer Handbooks of Computational Statistics*, 2007, pp. 93–137.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 10.01.2019. Подписано в печать 22.02.2019. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ319. 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.

Рисунок к статье Е. В. Артюшиной, Д. Е. Ершова, Ю. А. Прохода
«РОЛЬ ИНТЕРНЕТ-ТЕХНОЛОГИЙ В КОММУНИКАЦИОННОЙ ДЕЯТЕЛЬНОСТИ МЕДИЦИНСКОЙ ОРГАНИЗАЦИИ»

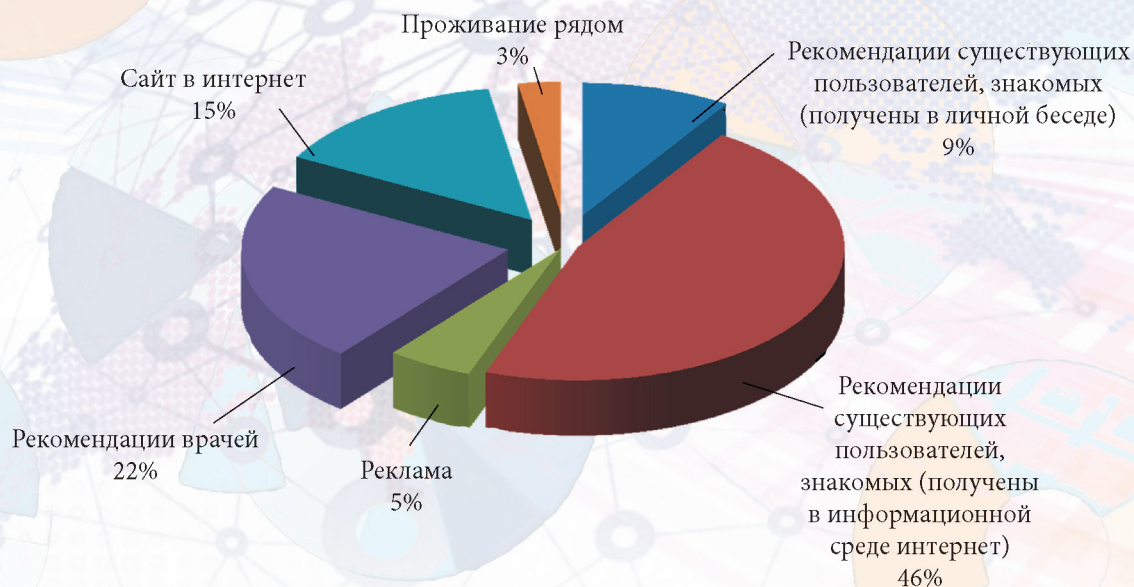


Рис. 1. Количественное распределение респондентов по используемым источникам информации при поиске медицинской организации для обслуживания

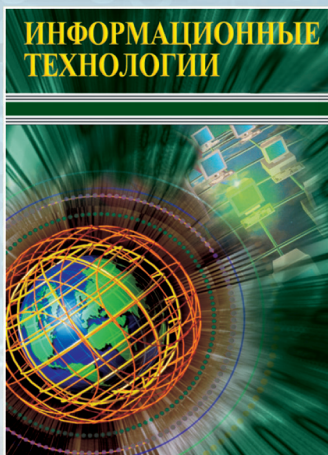
Рисунок к статье Ю. Д. Гвоздковой, И. А. Гвоздковой, А. В. Курочкина, А. В. Черняева

«ИНФОРМАЦИОННАЯ СИСТЕМА ОЦЕНКИ ЭКОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ АВИАЦИОННЫХ МАТЕРИАЛОВ И ТЕХНОЛОГИЙ МЕТОДОМ АНАЛИЗА ИЕРАРХИЙ»

<div style="display: flex; justify-content: space-around; align-items: center;"> ПУСК Резерв Из резерва Очист. знач. Очист. всё </div>					0,282	0,196	0,275	0,247				
№	Критерии оценки иет-в энергии (число от 2 до 100)	Параметр <i>m</i> (если лучший выбор – «минимум для абс. значения», то -1, иначе 1)	Весовой коэф. критерия (от 0 до 10)	Норм. весовой коэф. критерия	ВИДЫ ИС ТОЧНИКОВ ЭНЕРГИИ (число от 2 до 100)							
					Керосин		Водород		Солнечные батареи		Био	
					Абс. знач.	Вес. коэф.	Абс. знач.	Вес. коэф.	Абс. знач.	Вес. коэф.	Абс. знач.	Вес. коэф.
1	Возобновляемость	1	10	0,0885	0	0,0000	5	0,2000	10	0,4000	10	0,4000
2	Доступность	1	10	0,0885	5	0,2941	1	0,0588	10	0,5882	1	0,0588
3	КПД двигателя	1	10	0,0885	5	0,2000	10	0,4000	5	0,2000	5	0,2000
4	Экологичность	1	10	0,0885	0	0,0000	10	0,3333	10	0,3333	10	0,3333
5	Чувствительность к погоде	-1	10	0,0885	1	0,3226	1	0,3226	10	0,0323	1	0,3226
6	Износ двигателя	1	8	0,0708	2	0,1818	3	0,2727	3	0,2727	3	0,2727
7	Независимость от цен на топливо	1	5	0,0442	0	0,0000	0	0,0000	10	1,0000	0	0,0000
8	Стоимость	-1	10	0,0885	1	0,6250	5	0,1250	5	0,1250	5	0,1250
9	Выр-ка энергии	1	10	0,0885	10	0,3333	10	0,3333	0,005	0,0002	10	0,3333
10	Кол-во веп. техники	-1	7	0,0619469	1	0,4167	5	0,0833	5	0,0833	1	0,4167
11	Площадь и объем	-1	7	0,0619469	1	0,4167	5	0,0833	5	0,0833	1	0,4167
12	Эн. расход для пр-ва	-1	8	0,07079646	1	0,7143	10	0,0714	10	0,0714	5	0,142857
13	Воспламенение	-1	8	0,07079646	5	0,1333	10	0,0667	1	0,6667	5	0,133333

Рис. 1. Рабочее окно компьютерной программы АИС с результатами расчетов итоговых весовых коэффициентов ИЭ для авиатранспорта по МАИ

Издательство «НОВЫЕ ТЕХНОЛОГИИ» выпускает научно-технические журналы



Ежемесячный теоретический
и прикладной научно-технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Подписной индекс по Объединенному каталогу
«Пресса России» – 72656



Научно-практический
и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Ежемесячный
междисциплинарный
теоретический и прикладной
научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопросы разработки и внедрения нано микросистем в различные области науки, технологии и производства.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Ежемесячный теоретический
и прикладной
научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автоматического и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

Теоретический
и прикладной
научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

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.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 22765

Адрес редакции журналов для авторов и подписчиков:

107076, Москва, Стромьинский пер., 4. Издательство "НОВЫЕ ТЕХНОЛОГИИ".
Тел.: (499) 269-55-10, 269-53-97. Факс: (499) 269-55-10. E-mail: antonov@novtex.ru