

Программная инженерия

Пр 11
2014
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Михайленко Б.Г., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Липаев В.В., д.т.н., проф.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.С., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана, ОАО "Концерн "Сириус".

СОДЕРЖАНИЕ

Позин Б. А. SEMAT — Software Engineering Method and Theory. О чем, зачем и кому это нужно?	3
Пак Дж. С., Якобсон И., Майбург Б., Джонсон П. SEMAT вчера, сегодня и завтра: перспективы промышленного использования	6
Глоссарий Essence	17
Богоявленская О. Ю. Анализ алгоритма предотвращения перегрузок AIMD в сетях передачи данных	20
Александров Д. Е. Об уменьшении автоматной сложности за счет расширения регулярных языков	26
Жаринов И. О., Жаринов О. О. Исследование свойства равноконтрастности цветовых пространств, применяемых в авионике	35
Антонова Н. Е., Большаков А. А., Лобанов В. В., Киселев В. В., Перова Л. Г. Программный комплекс поддержки принятия решений медицинской диагностики заболеваний с использованием стабилметрической платформы	44

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индексы: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru/pi.html E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования.

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

© Издательство "Новые технологии", "Программная инженерия", 2014

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

№ 11

November

2014

Published since September 2010

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.), Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCENKO A. B., Dr. Sci. (Phys.-Math.), Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad. RAS
MIKHAILENKO B. G., Dr. Sci. (Phys.-Math.), Acad. RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.), Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.), Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B. I.
AFONIN S. A., Cand. Sci. (Phys.-Math)
BURDONOV I. B., Dr. Sci. (Phys.-Math)
BORZOVS JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A. V., Cand. Sci. (Phys.-Math)
GAVRILOV A. V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.), Switzerland
KORNEEV V. V., Dr. Sci. (Tech)
KOSTYUKHIN K. A., Cand. Sci. (Phys.-Math)
LIPAEV V. V., Dr. Sci. (Tech)
MAKHORTOV S. D., Dr. Sci. (Phys.-Math)
MANCIVODA A. V., Dr. Sci. (Phys.-Math)
NAZIROV R. R., Dr. Sci. (Tech)
NECHAEV V. V., Cand. Sci. (Tech)
NOVIKOV B. A., Dr. Sci. (Phys.-Math)
PAVLOV V. L., USA
PAL'CHUNOV D. E., Dr. Sci. (Phys.-Math)
PETRENKO A. K., Dr. Sci. (Phys.-Math)
POZDNEEV B. M., Dr. Sci. (Tech)
POZIN B. A., Dr. Sci. (Tech)
SEREBRJKOV V. A., Dr. Sci. (Phys.-Math)
SOROKIN A. V., Cand. Sci. (Tech)
TEREKHOV A. N., Dr. Sci. (Phys.-Math)
FILIMONOV N. B., Dr. Sci. (Tech)
SHAPCHENKO K. A., Cand. Sci. (Phys.-Math)
SHCHUR L. N., Dr. Sci. (Phys.-Math)
SHUNDEEV A. S., Cand. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A. V., CHUGUNOVA A. V.

CONTENTS

Pozin B. A. SEMAT — Software Engineering Method and Theory. About what, what for and to whom is it addressed?	3
Park J. S., Jacobson I., Myburgh B., Johnson P. SEMAT Yesterday, Today and Tomorrow. An Industrial Perspective	6
Essence Glossary	17
Bogoiavlenskaia O. I. Analysis of Congestion Avoidance Algorithm AIMD in the Data Communication Networks.	20
Aleksandr D. E. Reducing the Automata Complexity of Regular Languages by Means of Language Extension	26
Zharinov I. O., Zharinov O. O. The Research of Uniformchromaticity Property of Color Spaces, used in Avionics Equipment	35
Antonova N. E., Bolshakov A. A., Lobanov V. V., Kiselev V. V., Perova L. G. Program Complex of Support of Decision-Making of Medical Diagnostics of Diseases with Use of the Stabilometry Platform	44

Information about the journal is available online at:
<http://novtex.ru/pi.html>, e-mail: prin@novtex.ru

SEMAT – Software Engineering Method and Theory.

О чем, зачем и кому это нужно?

Статья посвящена краткому анализу основных положений инициативы SEMAT — Software Engineering Method and Theory по воссозданию единой теории программной инженерии. В рамках инициативы разработан стандарт OMG, получивший название Essence, определяющий ядро и язык программной инженерии. Ядро представляет собой минимальное количество сущностей, участвующих в процессе создания программной системы или являющихся его результатом, а также оно вводит отношения между этими сущностями. Разъяснены основные причины эффективности предлагаемой теории.

Ключевые слова: SEMAT, Essence, ядро, альфа, метод, практика, контрольные состояния

14 июня 2014 г. Object Management Group (OMG) объявила об утверждении документа "Essence Kernel and Language for Software Engineering Methods" (далее по тексту — Essence) в качестве стандарта OMG. Этот документ, описывающий ядро и язык программной инженерии, стал признанным международным сообществом результатом деятельности международной инициативы SEMAT (Software Engineering Method and Theory). Эта инициатива является попыткой воссоздания единой теории программной инженерии. Она была начата в 2009 г. под руководством так называемой Тройки (Тройка) в состав которой входили: Ивар Якобсон (Ivar Jacobson), известный программистскому сообществу автор объектно-ориентированной теории проектирования и разработки программ, языка UML и соавтор Rational Unified Process; Берtrand Майер (Bertrand Meyer), автор языка Eiffel и признанный авторитет в области методов программирования, автор нескольких монографий и учебников, изданных во всем мире, и Ричард Соули (Richard Soley), в то время — президент OMG.

Казалось бы, программная инженерия существует несколько десятков лет, изданы монографии по различным методам и технологиям создания программ. Однако методов и технологий много, а "вкусные" пристрастия программистов и влияние моды на применение тех или иных методов приводят к тому, что даже внутри одной компании специалисты по-разному понимают свои задачи. При переходе к новому проекту, к новой проблемной области, при работе в новой компании даже опытный программист должен переучиться, чтобы освоить процесс создания программ, который используется в этой компании. В то же время разные коллективы программистов совершают однотипные, в том числе и системные ошибки. Самой распространенной из них, пожалуй, является

недостаточный учет потребностей заказчика при определении требований к создаваемому программному средству.

Новая теория программной инженерии направлена на то, чтобы выработать общепринятый механизм создания программных систем, имеющий хорошую теоретическую базу и приводящий к существенному повышению качества программных систем как продукции индустрии. Этот механизм должен быть расширяемым и обладать целым рядом свойств, позволяющих использовать его разными по размеру коллективами, которые создают различные по функциональным возможностям и сложности программные системы.

Основой теории является ядро (*kernel*), которое построено как общая модель создания программной системы коллективом специалистов во взаимодействии с теми, кто может/должен выразить функциональные требования, а также с теми, кто должен реализовать их в программной системе. Ядро представляет собой минимальное количество сущностей, участвующих в процессе или являющихся его результатом, а также оно вводит отношения между этими сущностями. Сущности распределены по областям ответственности между тремя следующими группами: область клиента; область продукта или решения; область деятельности команды проекта. Сущности названы альфами (ALPHA — Abstract-Level Progress Health Attribute — атрибут абстрактного уровня прогресса и «здоровья»). В ходе проекта по созданию программной системы каждая альфа изменяет свое состояние, так что изменение состояний всех альф характеризует прогресс проекта на любой его стадии — при формировании потребности в программной системе, при формировании и работе команды проекта.

Однако наличие прогресса вовсе не означает, что он направлен на реализацию заданной цели и все

альфы работают согласованно. Понятие "здоровья" проекта характеризует степень того, насколько на каждом этапе развития проекта достигаются его цели. Для этого введены «правильные» состояния альф и контрольные списки состояний. Эти механизмы в совокупности позволяют управлять программным проектом не по «майстоунам», отражающим в управлении проектами формальный результат, который должен быть достигнут к заданному моменту времени, а по содержательно значимым сведениям о продвижении работ к достижению заданных целей. Контроль состояния проекта, таким образом, может проводиться командой проекта совместно, как по содержательным, так и по формальным результатам работ. Именно этого, как правило, не происходит в классическом управлении проектами. Систематическое использование ядра позволяет самой команде проекта проводить планирование работ, основываясь на реальном состоянии проекта, которое достигнуто к текущему моменту времени.

К числу ключевых в SEMAT/Essence относятся понятия *практики* и *метода*. Под *практикой* понимается по существу методика выполнения какой-либо повторяющейся последовательности работ, направленных на получение определенного результата (чаще промежуточного), при разработке программной системы. Примером практики является, например, Scrum. Комбинацию практик, обеспечивающую создание программной системы — от формирования заинтересованными сторонами возможностей будущей системы до сдачи ее в постоянную эксплуатацию — в SEMAT/Essence называют *методом*. Заметим, что в отечественной литературе для этого обычно используют термин *методология*. Принципиальным в SEMAT/Essence является то, что метод описывает все, что нужно сделать для создания программной системы, при этом учитываются все необходимые сущности и устанавливаемые между ними отношения. Метод вообще является ключевым элементом теории. Именно выбор метода определяет, какие конкретно практики работы являются наилучшими для применения. Например, метод Agile требует применения практики Scrum в сочетании с другими соответствующими практиками, которые могут не применяться, скажем, при каскадном методе ведения работ.

Следует однако отметить, что качество работ определяется не через практику, а через состояния сущностей, альф, что совершенно правильно. Именно такая позиция следует из необходимости не просто создать программную систему выбранным методом, а при этом обеспечить ее высокое качество. Этот факт означает, что каждая сущность должна реализовать все требуемые от нее действия и отношения с другими сущностями. С этой целью в SEMAT/Essence предусмотрено, что в зависимости от типа программной системы альфы могут иметь определенные «правильные»

состояния, к которым следует стремиться в каждом проекте. При этом в такие состояния можно попасть, переходя из одного заранее определенного состояния в другое заранее определенное состояние. В ходе работ по проекту участники могут отвлечься на решение каких-то текущих вопросов. Однако при регулярно проводимом анализе состояния выполнения проекта SEMAT/Essence требует проверки того, соответствует ли состояние альф их «правильным» состояниям. Для этого используются контрольные списки таких состояний. Это означает, что как бы интенсивно ни работала команда проекта, в любой момент можно понять, правильно ли развиваются работы по его выполнению и что необходимо сделать, чтобы направить их в правильное русло. Причем это может сделать сама команда проекта без посторонней помощи и за очень короткое время. Она не только сможет обнаружить свои ошибки, но и понять, в каком направлении она недорабатывает, что делает не так.

Конечно, метод в Essence описывает проект на довольно высоком уровне, описывает главное в проекте, оставляя возможность инженерам-технологам (инженерам по процессам) добавить в методы, используемые каждой компанией, необходимые практики, наиболее успешно ею применяемые. Например, компании дополняют метод практиками в области тестирования и другими. По утверждению авторов Essence, командам программистов не потребуется разрабатывать документы по процессам, которые делают в крупных компаниях. Конечно, небольшие команды смогут так работать. Однако при внедрении Essence в крупных программистских коллективах, для выполнения крупных проектов потребуется создавать описание методов, возможно, более компактное, чем современные аналоги, например, RUP.

Представляется, что появление стандарта Essence является важным событием, поскольку закладывает основу для организации промышленной разработки программных систем и для систематического образования специалистов в области программной инженерии. Конечно, это только начало пути к систематической теории программной инженерии. Потребуется еще много работы для широкого распространения SEMAT. Это путь, который должно пройти мировое профессиональное сообщество. Оно уже ступило на этот путь. Отрадно, что уже больше года существует Российское отделение SEMAT. Наше отделение проводит просветительскую работу (семинары, конференции «Актуальные проблемы системной и программной инженерии», SECR), постановку учебных курсов в МФТИ, МЭСИ, МИЭМ ВШЭ и других вузах, разработку теоретических вопросов создания и развития SEMAT. Российское отделение проводит работу по расширению SEMAT на область системной инженерии (по предложению и силами российского отделения международной организации International

Council of Systems Engineering — INCOSE), а также по расширению ядра на процессы всего жизненного цикла программных систем (силами группы «ЕС-лизинг»).

Узнать о текущих мероприятиях по развитию SEMAT можно на сайте www.semat.org или написав в Российское отделение SEMAT по адресу semat@es-leasing.ru.

В данном номере журнала публикуются (с согласия авторов) перевод статьи Дж. С. Парка (June Sung Park), И. Якобсона (Ivar Jacobson), Б. Майбурга (Barry Myburgh) и П. Джонсона (Pontus Johnson) «SEMAT вчера, сегодня и завтра: перспективы промышленного использования» и разработанный специалистами Российского отделения SEMAT глоссарий основных понятий.

B. A. Pozin, Professor, MIEM of National Research University Higher School of Economics, Chairman, SEMAT Russian Chapter, e-mail: bpozin@es-leasing.ru

SEMAT – Software Engineering Method and Theory. About what, what for and to whom is it addressed?

Article contains brief analysis of fundamentals of initiative SEMAT— Software Engineering Method and Theory with the aim of re-founding software engineering as a rigorous discipline. OMG standard named ESSENCE was developed and defined Kernel and Language for Software Engineering Methods. Kernel defines minimal number of key elements and relations between them in the process of creation of software system. Main reasons of effectiveness of the theory are discussing.

Keywords: SEMAT, Essence, kernel, alpha, method, practice, control states

ИНФОРМАЦИЯ



IX Международная научная конференция "Параллельные вычислительные технологии" (ПаВТ2015)

**30 марта – 3 апреля 2015 г.
Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина (г. Екатеринбург)**

"Параллельные вычислительные технологии (ПаВТ) 2015" — международная научная конференция, девятая в серии ежегодных конференций, посвященных развитию и применению параллельных вычислительных технологий в различных областях науки и техники. **Главная цель конференции** — предоставить возможность для обсуждения перспектив развития параллельных вычислительных технологий и представления результатов, полученных ведущими научными группами в использовании суперкомпьютерных технологий для решения задач науки и техники в странах СНГ и всего Мира.

Организаторы конференции:

- Российская академия наук
- Суперкомпьютерный консорциум университетов России

Тематика конференции покрывает все аспекты применения высокопроизводительных вычислений в науке и технике, включая приложения, аппаратное и программное обеспечение, специализированные языки и пакеты.

В первый день работы конференции будет объявлена **22-я редакция списка Top50** самых мощных компьютеров СНГ.

Во все дни работы конференции будет действовать суперкомпьютерная выставка, на которой ведущие производители аппаратного и программного обеспечения представят свои новейшие разработки в области высокопроизводительных вычислений.

Языки конференции: русский, английский.

Официальный сайт конференции: <http://agora.guru.ru/pavt2015/>

Дж. С. Парк, проф., Корейский институт современной науки и технологии, г. Тэджон, Республика Корея, президент, SEMAT Inc., Корея,
И. Якобсон, проф., президент Ivar Jacobson International, Швейцария,
Б. Майбург, Йоханнесбургский центр программной инженерии, Витватерсрандский университет, г. Йоханнесбург, Южная Африка,
П. Джонсон, проф., Королевский технологический институт, г. Стокгольм, Швеция

SEMAT вчера, сегодня и завтра: перспективы промышленного использования

Данная статья — это исторический экскурс к истокам SEMAT и рассказ о прошлых свершениях, сегодняшних приоритетах и планах на будущее. Статья ориентирована прежде всего на представителей промышленности. Следует особо отметить, что научная дискуссия о теоретических аспектах программной инженерии не входила в замысел авторов статьи, поэтому собственно теория программной инженерии затрагивается в ней лишь поверхностно. Этой теме планируется посвятить следующие публикации.

Ключевые слова: SEMAT, Essence, ядро, альфа, метод, практика

1. Начало SEMAT

В 2009 г. Ивар Якобсон (*Ivar Jacobson*), Бертран Мейер (*Bertrand Meyer*) и Ричард Соли (*Richard Soley*) выступили с инициативой, получившей название SEMAT — *Software Engineering Method and Theory* (Методология и теория программной инженерии). Ее заявленной целью было воссоздание программной инженерии как дисциплины с четкой методологией. Начальным этапом развития SEMAT стал так называемый «Призыв к действию» (*Call for Action*) [1], опубликованный в сентябре 2009 г. и четко определивший основные вызовы и задачи, стоящие перед адептами программной инженерии. В числе таких вызовов:

- ✓ избыточное количество разрозненных методов, которые трудно сравнивать один с другим вследствие отсутствия общей теоретической базы программной инженерии;
- ✓ нехватка экспериментальных оценок и валидации;
- ✓ разрыв между научными изысканиями и их практическим промышленным применением.

«Призыв к действию» предлагал приступить к воссозданию программной инженерии на новой методологической основе. Ее должны были составить убедительные теоретические подходы, подтвержденные практикой принципы и самый передовой опыт, которые способны обеспечить:

- создание ядра, состоящего из общепринятых элементов, с возможностью его расширения под конкретные нужды;
- учет одновременно технологических вопросов и человеческого фактора;

- получение поддержки со стороны представителей промышленности, научного сообщества, исследователей и пользователей;

- поддержку расширения состава методов и средств программной инженерии в условиях изменяющихся требований и перманентного развития технологий.

Очень скоро тысячи людей по всему миру, более десятка известных компаний, таких как IBM, Ericsson, ABB, и примерно столько же исследовательских институтов, в числе которых Пекинский университет (*Peking University*), Королевский технологический институт Стокгольма (*KTH Royal Institute of Technology*) и Флоридский Атлантический университет (*Florida Atlantic University*), поддержали инициативу SEMAT.

Таким образом, SEMAT выявил необходимость переосмысления и закладки нового, надежного научного фундамента с использованием всего опыта в области теории и практики программной инженерии, накопленного за последние десятилетия.

В январе 2010 г. предлагаемый подход к решению поставленных задач был представлен на суд общественности в документе, озаглавленном «Общая концепция» (*Vision Statement*) [2]. По сравнению с «Призывом к действию» «Общая концепция» содержала более подробное разъяснение рамок предполагаемой работы, поэтому по форме и содержанию являлась своеобразным «перечнем требований» и «дорожной картой» для решения сложной задачи, направленной на воссоздание программной инженерии на новой методологической основе. В документе представлены общее видение и объем будущей работы, поставлены

задачи, а также определены принципы и ключевые контрольные точки на первый год.

Концепция [2] предусматривала создание ядра для описания существующих и будущих так называемых **практик** (методик работы и способов их реализации), а также **методов** (методологий проектирования и разработки) в контексте формирования общей платформы, в рамках которой появилась бы возможность:

- проводить **сравнение** результатов применения практик и методов;

- проводить **оценивание** их эффективности, определять возможность их совместного использования (**сочетаемость**);

- осуществлять **моделирование** результатов;

- выявлять их **область применения**;

- понимать способы и эффективность **обучения** персонала, принимающего участие в разработке и сопровождении программного обеспечения на всех этапах его жизненного цикла;

- проводить поисковые **исследования** в области программной инженерии.

Работы по созданию ядра предусматривали создание универсальных элементов и языка ядра. По замыслу авторов концепции ядро должно быть платформой для определения практик и методов, при этом метод представляет собой композицию практик.

В январе 2011 г. работа над ядром была перенесена в рабочую группу OMG (*Object Management Group*) в соответствии с действующей в OMG обычной процедурой принятия стандарта. Это было сделано для обеспечения открытости проекта и реализации более честного (конкурентного) процесса отбора решений, предлагаемых в рамках его выполнения, а также для того чтобы все лица, заинтересованные в программной инженерии, могли пользоваться результатами проделанной работы. В июне 2011 г. был выпущен документ Request for Proposals — RFP («Запрос на предложение») [3].

Следующий раздел статьи посвящен историческому обзору прошедшего этапа развития SEMAT и поставленным на этот период задачам.

2. SEMAT вчера

Становление профессионального сообщества программистов заняло более 60 лет. Исторический анализ показывает, что за редким исключением каждая команда разработчиков обычно использует свой собственный, отличный от других команд метод. По этой причине вполне закономерно будет предположить, что сейчас в мире существует около 100 000 методов, большинство из которых никогда не были описаны. Впрочем, число подходов, принципиально отличных друг от друга, значительно меньше. Как следствие, в программной инженерии существует всего пара сотен четко определенных практик.

В прошлом, различные методы (методологии проектирования и разработки) изначально описывались как изолированные системы, своего рода концепту-

альные острова, не сообщающиеся между собой. Каждый метод представлял собой единственный в своем роде феномен, описываемый своим собственным языком и собственными терминами, без опоры на некий широко применяемый общий базис.

Воссоздание программной инженерии — это беспрецедентный процесс, в основе которого лежит формирование «ядра, состоящего из широко признанных элементов» (*the widely-agreed kernel*), как общей платформы для всей программной инженерии. Независимо от того, какой код будет написан, какая программная система будет разработана, какое решение будет реализовано, какие методологии и технологии работы будут выбраны, а также независимо от особенностей участвующих в проекте организаций, должна оставаться неизменной основа, т. е. ядро основных всепроникающих элементов (например, понятия и качества, присущие любой целенаправленной деятельности по созданию программного продукта). Такое ядро основных элементов — «Сущность программной инженерии» (*Essence of Software Engineering*) — описано в одноименной книге [4].

В начале 2011 г. содержание «Общей концепции» было обновлено и значительно расширено, результатом чего явилась публикация «SEMAT — трехлетняя концепция» (*SEMAT — Three Year Vision*) [5], описывающая основные вехи развития SEMAT на перспективу до начала 2014 г. В данном документе более подробно было рассказано о концепции ядра и требованиях к нему (реализованных в первой версии ядра), которые к тому времени уже были разработаны. Публикация также описывала прогнозируемые результаты развития SEMAT на трехлетнюю перспективу и открывала обсуждение ключевых факторов успеха и контрольных показателей для SEMAT и заинтересованных сторон. Двумя основными направлениями трехлетнего плана развития стали:

- определение ядра из широко признанных элементов;

- определение убедительной теоретической базы.

Создание ядра, состоящего из широко признанных элементов, включало в себя несколько следующих векторов деятельности:

- составление ядра и языка под эгидой OMG;

- разработку инструментальных средств ядра (включая открытый исходный код — *open source*) для удовлетворения потребностей практиков по авторским разработкам, сравнению, составлению, использованию и оценке практик и методов;

- открытие центра обращения практик (*practice marketplace*), предназначенного для опубликования, каталогизации и обращения практик, описанных в ядре;

- разработку учебного курса по использованию ядра для целей программной инженерии;

- написание книг и прочих материалов для действия учебному и личностному развитию.

Основная часть настоящей статьи посвящена более подробно обсуждению некоторых из перечис-

ленных векторов деятельности. Наилучшим подспорьем в решении задачи по определению убедительной теоретической базы могло бы стать исследование по разработке «Общей теории программной инженерии» (*General Theory of Software Engineering (GTSE)*).

Данная статья, однако, посвящена в основном ядру. Такая установка была взята специально, с желанием удостовериться, что ядро определено настолько четко (не допускается двойного толкования и т. д.), насколько это нужно для сравнения различных способов его использования.

Не вызывает сомнений, что наличие убедительных теоретических основ критически важно для научного сообщества, но есть основания полагать, что они также важны и для промышленности, поскольку именно промышленность определяет способность уверенно прогнозировать последствия принятия тех или иных проектных решений, а следовательно, снижает риск провала, который всегда присутствует в любом проекте по созданию программного продукта.

При разработке ядра были использованы следующие принципы, предложенные в «SEMAT — трехлетняя концепция».

Принцип расширяемости предусматривает создание единой базы, используемой для любых видов деятельности по созданию программных продуктов (от маленьких до больших проектов, от упрощенных до формальных моделей жизненного цикла, от обобщенных до детализированных методов, от негласных до четко определенных практик, от простых до сложных программных систем, для различных областей применения и т. д.). Расширение ядра должно выполняться просто, без изменения или усложнения самого ядра. Таким образом, ключевым принципом SEMAT является способность обеспечивать выполнение всей деятельности и, следуя принципам программной инженерии, без необходимости исключать то, что является или будет являться важным для одной или нескольких заинтересованных групп. Реализация данного принципа стала возможна благодаря отделению элементной базы программной инженерии (выраженной в ядре) от методологической базы (выраженной в практиках и методах). Именно поэтому при создании ядра применяется принцип «разделения ответственности» (*Separation of Concerns*). Даже если основной акцент делается на основах программной инженерии, участники могут добавлять какие-то детали. Даже если определения являются обобщенными, они могут быть проработаны и сделаны более узкими. Так, например, ключевой целевой группой являются специалисты-практики, однако благодаря возможности расширения, ядро также может использоваться инженерами по процессам (*process engineer*). Другой пример: основной предметной областью является программная инженерия, но за счет расширяемости ядра возможно также обеспечить поддержку методологии системной инженерии.

Принцип применяемости обеспечивает активную поддержку практиков (аналитиков, разработчиков,

тестировщиков, руководителей проектов и т. п.) в выполнении ими ежедневной работы с помощью легких в использовании карточек (*easy-to-use cards*). В кармане у каждого участника проекта находится небольшая колода карточек-подсказок. Если возникает необходимость обсудить состояние проекта на текущей стадии его выполнения, то участники достают колоды и используют их для совместной работы. Эти же карточки могут использоваться для поиска путей улучшения проекта. В таком подходе кроется фундаментальное отличие от традиционных подходов, которые, как правило, уделяют чрезмерное внимание описанию методов, забывая об эффективности их использования. Чрезвычайно важно, что акцент смещен от инженера по процессам в сторону практика. Размер проектов бывает разным, и огромное число разработчиков программного обеспечения работают в малых и средних компаниях, штатные расписания которых вообще не предусматривают должностного инженера по процессам. Без привлечения широкого круга практиков данная инициатива останется лишь интересной, но "гимнастикой для ума". Расширяемость, основанная на разделении ответственности, позволяет использовать ядро как практикам, так и инженерам по процессам.

Принцип компонентности позволяет проектной команде быстро создавать метод, комбинируя практики, исходя из потребностей, опыта и поставленных перед ней задач. Использование единого ядра при описании всех практик позволяет команде выявлять пробелы, перекрытия, конфликты и дополнения в рамках каждого набора практик, после чего команда легко может их объединить в четко структурированный метод.

Реализация **принципа адаптируемости** позволит обеспечить доработку и изменение практик и методов в процессе реализации проекта, если исходя из полученного на каком-то этапе опыта стало понятно, что это следует сделать, или если это вызвано изменением исходных требований. Традиционно метод определяется с прицелом на его будущую практическую реализацию, и действия, созданные на основании этого определения, выполняются специалистами-практиками в соответствии с заранее установленной их последовательностью. Итогом выполнения действия должен стать результат, заложенный в определении. Подход типа «команда — это компьютер, процесс — это программа» не подходит для такой креативной и требующей высокой степени гибкости работы, как программная инженерия, результат которой рождается из проб, ошибок и тесного сотрудничества. Удачные методы могут появиться только в результате адаптивного, гибкого, легкого и рационального подхода к их разработке. До некоторой степени это означает более эффективную организацию работы по компоновке и адаптации методов. Команды, которые раньше не имели дела с проектным методом, могут начать с более широкого и менее сложного его определения, чтобы впоследствии по мере продвижения проекта и повышения уровня компетентности и знаний команды

перейти к более строгому и регламентированному оп-
ределению.

Принцип масштабируемости означает возможность использования удачного метода при переходе от од-
ного продукта к нескольким, от одной команды к не-
скольким, от одного метода к нескольким. При орга-
низации процесса разработки программного обеспе-
чения применяемый набор практик может отличаться в
зависимости от предпочтений той или иной коман-
ды-разработчика. Описав все практики и методы, ис-
пользовавшиеся ранее при разработке программного
обеспечения на основе единого ядра, организация
приобретает возможность создавать и поддерживать в
актуализированном состоянии архитектуру принятого
метода. Подобная, основанная на ядре, корпоратив-
ная архитектура метода помогает выявлять общие для
нескольких команд практики. Она позволяет коман-
дам обмениваться информацией и совместно улуч-
шать практики, изучать новые и успешные практики
других команд, обучать новых сотрудников пользо-
ваться этими практиками в объеме, который соответ-
ствует занимаемой сотрудником должности. Корпора-
тивная архитектура метода способствует более быст-
рой адаптации работника, переведенного в другую
команду, к новой для него, обладающей определен-
ной спецификой командной практике. Таким обра-
зом, можно полностью обеспечить выполнение внут-
ри организации функции обучения. Организация
процесса разработки программного обеспечения, ис-
пользующая архитектуру метода, основанную на ядре,
предоставляет возможность осуществлять контроль и
мониторинг текущего состояния и прогресса разра-
ботки по каждому текущему процессу, используя
стандартные критерии. Кроме того, предоставляется
возможность гораздо проще управлять большим про-
ектом, в котором задействованы несколько команд,
каждая из которых использует свой набор практик.
В рамках такого проекта на основе ядра можно объ-
единять совершенно разные практики в единый мет-
од, применимый для всего проекта. Такой подход
позволяет отследить динамику продвижения проекта
к заданной цели на основании единых для всего про-
екта критериев оценки.

Подводя итог изложенному можно сказать, что це-
лью SEMAT было создание ядра программной инже-
нерии, обладающего свойствами расширяемости,
применимости, компонуемости, адаптируемости и
масштабируемости с тем, чтобы предоставить возмож-
ность специалистам-практикам и инженерам по про-
цессам использовать ядро для сравнения, оценки,
приспособления, составления, моделирования, ис-
пользования, адаптации, контроля изменений и со-
вершенствования практик и методов.

Поиск элементов ядра на начальных этапах проек-
та был жизненно важной задачей. В ходе исследова-
ний и обсуждений сообществом SEMAT такие уни-
версальные и значимые элементы были найдены.
Главным при реализации поиска был принцип: «иде-
альная система — не та, в которую нечего добавить, а

та, из которой убрано все лишнее». Сообщество
SEMAT сочло также нужным убедиться, что принци-
пы, оставшиеся после исключения «всего лишнего»,
являлись широко признанными, понимая, что успех
инициативы напрямую зависит от такого признания.
По этой причине разработка ядра SEMAT не пред-
ставляла собой решение чисто технической задачи.
Не менее важно было добиться соглашения об основ-
ных элементах ядра, их возможных состояниях и кри-
териях перехода из одного состояния в другое.

Ядро было разработано несколькими ключевыми
членами SEMAT в сотрудничестве с двадцаткой ак-
тивных участников выполнения этого проекта. Ре-
зультатом работы стало появление документа, кото-
рый в настоящее время называется *Essence*. В июне
2013 г., после публичного обсуждения и проверки в
OMG, документ *Essence* был принят и получил статус
бета-версии стандарта OMG «*Essence* — ядро и язык
для методов программной инженерии» (*Essence —
Kernel and Language for Software Engineering Methods*)¹.

Essence описывает три типа элементов ядра: альфы,
пространства действий и компетенции.

Альфы (Альфа — Abstract-Level Progress Health
Attribute — Атрибут абстрактного уровня прогресса и
«здоровья») — это элементы программно-инженер-
ной деятельности, развитие которых в процессе вы-
полнения проекта участники этой деятельности долж-
ны понять, отследить, направить и проконтролиро-
вать. Альфам, как видам деятельности, назначаются
достижимые состояния, а для мониторинга прогресса
и «здоровья» процесса к альфам привязываются кон-
трольные состояния, оценив которые можно при не-
обходимости скорректировать деятельность таким об-
разом, чтобы добиться успешного результата. Конт-
рольные списки таких состояний — это не обычные
списки вида «поставьте-тут-крестик-если-сделано». Они
позволяют измерить реальные, значимые для ка-
чества проекта результаты, достижение каждого из
которых определяет возможность перевода альфы из
одного состояния в другое. В стандарте *Essence* в ядро
включены семь альф: **заинтересованные стороны; воз-
можность; требования; команда; работа; технология
работы; программная система.**

Пространства действий — это наборы действий для
продвижения альф от состояния к состоянию. Набо-
ры действий не одинаковы для разных видов програм-
мно-инженерной деятельности или групп разработчи-
ков. В ядре не предусматривается конкретных действий,
в нем только выделены «места» пространств, в которых
должны выполняться действия, необходимые для ре-
зультативного завершения конкретной деятельности².
Не существует обязательного к соблюдению порядка

¹ В июне 2014 г. *Essence* получил статус стандарта OMG
(прим. ред.).

² В исходном документе *Essence* используется термин
Endeavor, означающий целенаправленную деятельность группы
или организации по реализации конкретной программной сис-
темы. Точный перевод этого термина приведен в официальном
гlossарии (прим. ред.).

выполнения пространства действий. Иерархия действий составляется отдельно для каждой деятельности с учетом выбранного командой метода³. Например, при гибкой реализации (эджайл, *agile implementation*) могут появляться более высокие уровни параллельно выполняемых операций, а при каскадной (водопадной) реализации может возникать больше промежуточных фаз.

Компетенции представляют собой характеристики заинтересованной стороны или члена команды, которые отражают их способность выполнять определенную работу. Компетенция описывает способность выполнить определенную работу и оценивается по шкале чередования уровней компетенции от минимального до максимального. Рекомендуется установить нулевой уровень для тех, кто только «помогает», и пятый — для «инноваторов».

Для понимания SEMAT крайне важно упомянуть о том, чем ядро не является. Ядро, описанное в *Essence*, не является ни новой унифицированной методологией, ни метамоделью процесса создания программного продукта, ни комплексом знаний, ни новым языком моделирования. Оно создавалось не для того чтобы люди построили на нем новые инструментальные средства или купили как можно больше таких средств. Оно — не что иное, как карта того, что в индустрии и программной инженерии уже присутствует (например, команды и проекты), что уже делается (например, определяется и реализуется), какие навыки специалистам надо иметь (например, навыки разработки и тестирования) и того, что индустрия уже производит (например, программные системы). Акцент всегда ставится на программной инженерии, независимо от того, что специалисты делают, пишут ли документацию, и даже от того, хороший или плохой результат в итоге получают команды разработчиков.

3. SEMAT сегодня

Данный раздел посвящен сегодняшнему состоянию проекта SEMAT и тому, чем в настоящее время занимается численно растущее сообщество SEMAT. По актуальным данным современное сообщество SEMAT включает почти 2000 сторонников (www.semat.org) и около 100 активных участников. На рис. 1 представлена организационная структура SEMAT. Числа в скобках указывают на долю от общей активной поддержки, оказываемой регулярными участниками заседаний и проектов, приходящуюся на данную структурную единицу. Некоторые участники поддерживают сразу несколько единиц, поэтому следует понимать, что реальное число активных участников не равно сумме чисел на рисунке.

На настоящее время самой важной вехой стало принятие ядра *Essence* в качестве стандарта OMG.

³ Состав действий может отличаться, например, для похожих видов деятельности, но разных классов программных систем.

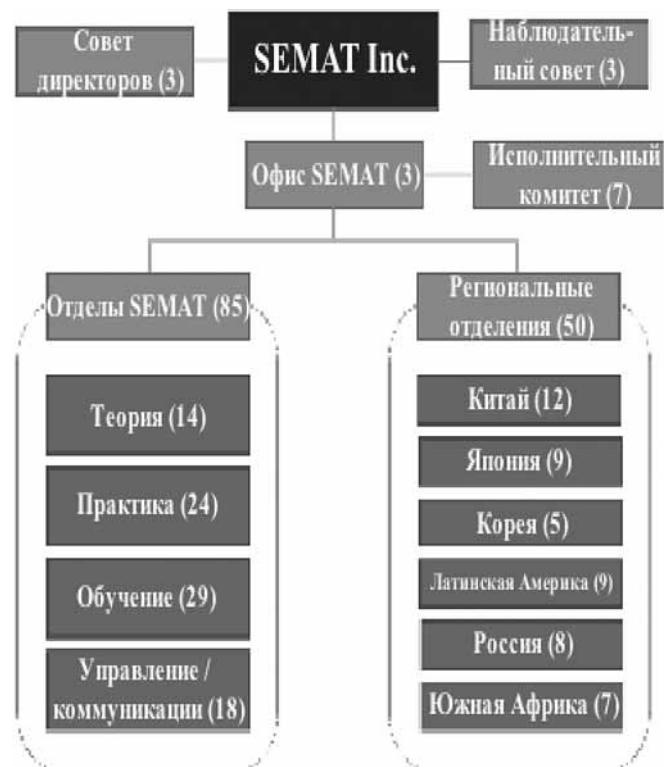


Рис. 1. Организационная структура SEMAT

SEMAT-практика

Отдел практики SEMAT (рис. 1) разработал спецификацию *Essence*, а сейчас занимается составлением «Руководства пользователя *Essence*» (*Essence User's Guide*) и нескольких руководств для проведения программно-технических работ на базе *Essence*. Эти работы важны, поскольку от их результатов зависит, насколько быстро и эффективно будут проходить обучение в соответствии с этими руководствами, аттестация участников, тренинги и сертификация *Essence* в промышленности.

Активные члены практического отдела написали несколько книг и статей с целью объяснить специалистам-практикам, как они могут использовать *Essence* в повседневной работе и тем самым улучшить результат выполнения проекта. В ряде работ показано, как альфы *Essence*, состояние альф и контрольные списки состояний могут использоваться для оценки прогресса и «здоровья» проекта независимо от применяемых методов и практик программной инженерии. Например, работа [4] содержит примеры практического применения *Essence* в реальных промышленных условиях. Сотрудники компаний, не привыкшие "слепло" принимать все новшества, делятся опытом о реальной пользе практического использования состояний альф для проверки выполнения проекта, а также рассказывают о преимуществах определения набора практик, которое становится возможным после пере-

хода на ядро *Essence*. Среди опрошенных компаний — ведущий японский производитель электроники для розничного рынка *Fujitsu Services UK*, а также компании *KPN*, *Munich Re*, ведущий государственный департамент Великобритании, крупная китайская телевизионная продюсерская компания и китайская Интернет-медиа-компания. Детальные отчеты о ситуационных исследованиях и статьи о впечатлениях пионеров практического использования *Essence*, призванные помочь новым пользователям, в настоящий момент находятся на стадии оформления.

SEMAT-обучение

В промышленности. Готовое руководство по *Essence* было представлено на конференции SEMAT, которая проходила в мае 2013 г. параллельно с Международной конференцией по программной инженерии (*International Conference on Software Engineering, ICSE*). На однодневном семинаре по *Essence*, прошедшем в июне 2013 г. в рамках Технического совещания OMG (*OMG Technical Meeting*), были представлены руководства по *Essence*, примеры его использования и учебные примеры, а также несколько демо-версий инструментальных средств.

Южноафриканское региональное отделение организовало семинар «Использование SEMAT *Essence* для проверки «здоровья» проекта и планирования мероприятий» (*Using SEMAT Essence for Project Health Check and Action Planning*). Задачей семинара было знакомство делегатов с *Essence* и представление ситуационного исследования об использовании ядра *Essence* при разработке программного обеспечения.

В настоящее время компания *Ivar Jacobson International (IJI)* предлагает несколько курсов по *Essence*, включая двухдневный курс «SEMAT для гибкой разработки и бережливой разработки» (*Agile and Lean with SEMAT*). В компании *IJI* также разработали программу сертификации, которая призвана способствовать увеличению числа инструкторов, готовых обучать *Essence*, и инженеров, использующих *Essence*.

В университетах. Семь проектных команд в Университете Карнеги-Меллона (*Carnegie Mellon University*) в Кремниевой долине приняли на вооружение набор практик и используют состояния альфы *Essence* для планирования и оценки проектов. Все они заявляют о пользе *Essence* для обеспечения постоянного контроля «здоровья» проекта, независимо от используемых ими практик.

Шведский Королевский технологический институт также использует *Essence* в рамках двух курсов по программированию. С помощью контрольных списков состояний альфы ядра *Essence* студенты оценивают сильные и слабые стороны методов, использованных ими в проектах.

В Университете Витватерсранда (*University of the Witwatersrand*), г. Йоханнесбург (ЮАР) SEMAT используется в качестве инструмента для перевода опи-

санных в классических учебниках практик в современный контекст.

Отдел SEMAT, отвечающий за обучение, разрабатывает учебные материалы по *Essence*, включая тезисы для лекций, ситуационные задания, проекты для практиков и другие средства для преподавания курса.

В ближайшее время специалистам-практикам потребуются инструментальные средства, с помощью которых они могли бы описывать практики в ядре *Essence*, с легкостью компоновать их в проектный метод и применять полученный метод в повседневной работе. Для этой цели компания *IJI* предлагает использовать инструментальные средства *Practice Workbench* и *EssWork*. Помимо этого, *IJI* выпустила мобильное приложение, доступное в *Apple App Store*, которое позволяет использовать виртуальные карточки состояний альфы для планирования проекта и отслеживания состояния его выполнения. Фирма *KnowGravity* разработала прототип программы для корпоративного моделирования *KnowEnterprise*, включающий альфы *Essence*.

SEMAT-теория

Общая теория программной инженерии. Теоретический отдел SEMAT призван сделать важнейшую работу: опубликовать «Общую теорию программной инженерии». В основе теории лежат язык и ядро *Essence*. Она призвана продемонстрировать преимущества, которые дают хорошие теоретические основы специалистам-практикам программной инженерии, в том числе в части методов принятия решений, ориентированных на достижение целей. Такие заявления основываются на возможности теоретического предсказания результата выполнения работ.

Информирование и развитие сообщества. Для полного раскрытия теоретического потенциала SEMAT необходимо повысить уровень информированности сообщества специалистов по программной инженерии о важности теории.

Серия мероприятий под общим названием «Семинары SEMAT по общей теории программной инженерии» (*SEMAT Workshop on a General Theory of Software Engineering, GTSE*) является ключевым компонентом процесса повышения осведомленности профессионального сообщества.

Для увеличения числа членов сообщества SEMAT важно стимулировать расширение спектра точек зрения на эту концепцию. Попытки насильственно навязать исследователям и практикам потерявшие актуальность директивные принципы и положения могут привести к тому, что они дистанцируются от сообщества SEMAT. По этой причине SEMAT должен оставаться площадкой для свободного обмена мнениями об общих принципах и положениях, которые определяют как теоретические, так и практические его аспекты. Данная статья обращена прежде всего к представителям промышленности, поэтому в ней не обсуждаются вопросы теории SEMAT.

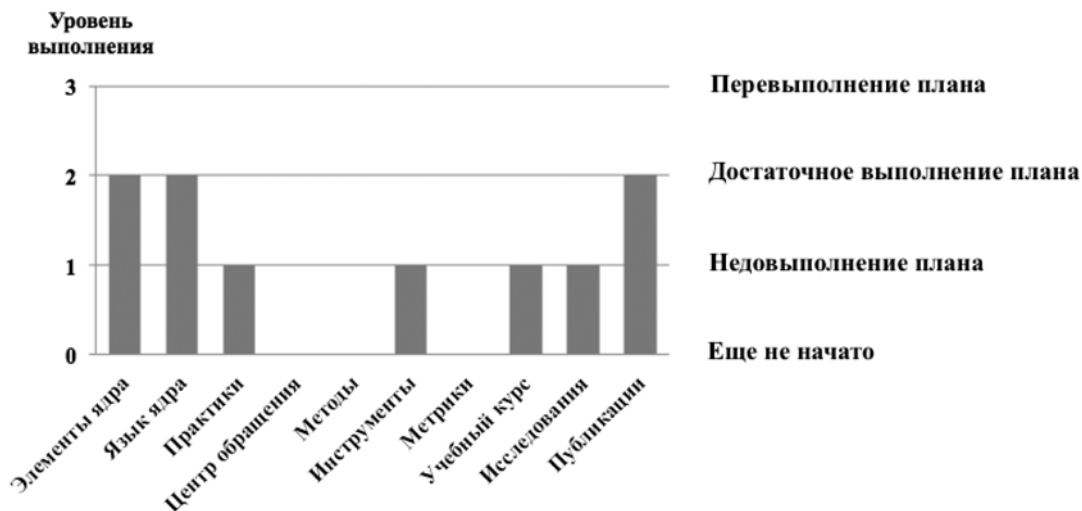


Рис. 2. Достижения SEMAT по состоянию на 2013 г.

Региональные отделения SEMAT

Как показано на рис. 1, существует всего шесть региональных отделений SEMAT. В июне 2014 г. создано также отделение SEMAT в Индии.

Китай. Ассоциация программно-инженерной промышленности Китая (*China Software Industry Association, CSIA*) привлекла китайское отделение SEMAT для проведения внедрения *Essence* в качестве основного компонента Национального плана качества на 2012—2020 гг. (*National Quality Plan, Q Plan*) для программно-инженерной отрасли. Данный план нацелен на повышение качества и международной конкурентоспособности китайских программных продуктов. *Essence* будет использоваться для подготовки разработчиков, улучшения практик развития и измерения качества и зрелости программно-инженерной отрасли на трех следующих уровнях: уровень компании, уровень продукта и уровень инженера.

Корейское отделение получило грант от государства для проведения научно-исследовательских изысканий по созданию практик, основанных на *Essence*.

Отделения других географических регионов — **Латинской Америки, России и Южной Африки** — провели несколько конференций и семинаров для продвижения *Essence*. Мероприятия получили положительные отзывы участников. Сведения о мероприятиях региональных отделений регулярно публикуются на сайте www.semat.org.

Замысел SEMAT заключался в создании расширяемой *Essence*, которая подходила бы разным видам программной инженерии. В настоящее время **российское** отделение SEMAT работает как раз в этом направлении, изучая возможность расширения ядра *Essence* под нужды системной инженерии, имея в виду разработку встроенного программного обеспечения⁴.

Краткий вывод о текущем положении дел

На рис. 2 продемонстрирована некоторая средняя оценка состояния развития SEMAT на настоящее время с учетом целей, объема работ, принципов и планов, содержащихся в опубликованных ранее концепциях [2, 5].

4. SEMAT завтра

Оценка состояния дел, представленная в разд. 3, позволяет констатировать, что по состоянию на настоящее время значительный объем работы уже выполнен. В проекте задействованы около 100 участников SEMAT из разных стран. Принимая во внимание тот факт, что все эти усилия направлены на воссоздание программной инженерии, сообщество SEMAT следует оценить положительно. На самом деле речь никогда не шла о декларативном озвучивании новых идей, практик и инструментов. Поставленная задача не ограничивалась только контактом с самыми активными профессионалами в области создания и использования методологий (методов) и средств программной инженерии (практик). Она с самого начала ставилась шире: SEMAT должны в обозримой перспективе принять на вооружение все 20 млн специалистов-практиков в разных частях света. Именно такую оценку числа специалистов в этой области дают результаты последних исследований.

⁴ Авторы не очень точно характеризуют работы по расширению *Essence* для системной инженерии, ограничивая их только встроенным программным обеспечением. На самом деле в этих работах делается попытка существенно расширить область применения *Essence* при создании информационных систем различных классов. См., например, работы А. Левенчука и российского отделения INCOSE.

Инициатива SEMAT открыта для всех желающих. Помимо трех уже действующих векторов развития (практика, обучение и теория) планируется открыть еще несколько новых. Наиболее перспективными представляются перечисленные далее направления.

- Разработка более абстрактного ядра, нежели ядро *Essence*.

- Системная инженерия, основанная на ядре *Essence*. Российское отделение SEMAT уже работает над расширением *Essence* для применения его в системной инженерии⁵. Представляется естественным привлечь в этот процесс и других представителей мирового сообщества. Отметим, что уже сейчас профессор Бад Лоусон (*Bud Lawson*) и Ивар Якобсон редактируют сборник свежих статей о системной инженерии.

- Бизнес-инженерия с использованием *Essence*. В настоящее время один из участников исследований на этом направлении расширил ядро для поддержки бизнес-проектов.

- Программы высшего образования, посвященные разработке программных продуктов с использованием *Essence*. Один из факультетов одного российского вуза рассматривает такую возможность.

- Разработка инструментальных средств для поддержки *Essence* как проекта с открытым исходным кодом (*open source*).

Стандарт для ядра *Essence*

Ядро и язык, стандартизированные OMG, должны оставаться стабильными, но не статичными. Их эволюция продолжится по мере расширения сферы их практического применения и получения офисом SEMAT отзывов от пользователей. Параллельно с результатами на этом направлении будет улучшаться и наше понимание программной инженерии.

Принятие *Essence* в качестве открытого стандарта OMG не представляет собой последний этап его развития. Скорее это первый заметный этап такого развития, первый шаг для SEMAT на пути к признанию его специалистами-практиками и профессиональными программистами.

Essence продолжит создание и продвижение новой экосистемы для создания практик и методов программной инженерии. Чтобы интерес пользователей не угасал, разработчики SEMAT должны активно, можно даже сказать агрессивно реагировать на их отзывы. Именно поэтому в SEMAT первоочередное внимание уделяют открытию и поддержанию каналов оперативной обратной связи для сбора пользовательских отзывов. Расставленные управленческие приоритеты позволяют быстро реагировать на болевые вопросы, высеченные пользователями в отзывах, и учитывать их в новых версиях стандарта *Essence*.

⁵ Российское отделение SEMAT работает над расширением ядра на весь жизненный цикл программной системы (прим. ред.).

SEMAT и общая теория программной инженерии

В расширенной версии SEMAT следует реализовать функцию прогнозирования, т. е. оценки последствий принятия решений, затрагивающих основы программной инженерии. В широком смысле, теория отвечает на достаточно простые вопросы, например:

- ✓ как изменение подхода к работе скажется на успехе проекта?

- ✓ как изменение структуры и характеристик команды скажется на успехе проекта?

- ✓ как изменение содержания и формы требований скажется на успехе проекта?

Целевое практическое исследование. Помимо разработки "Общей теории программной инженерии" (GTSE), исследования в области практического применения должны помочь найти ответы на следующие вопросы:

- ✓ где лучше описан «Скрам» (*scrum*) — в *Essence* или в "Руководстве по скраму" (*Scrum Guide*)?

- ✓ что лучше, двойное программирование (*pair programming*) или конкурентное программирование (*peer programming*)?

- ✓ что лучше, варианты использования (*use cases*) или пользовательские истории (*user stories*)?

Повышение информированности и развитие профессионального сообщества. Основной целью этого поднаправления деятельности остается повышение информированности сообщества о пользе общих теорий.

SEMAT-практика

В "Руководстве пользователя *Essence*" приводятся различные способы применения *Essence*. Руководство также предоставляет базовую информацию, необходимую для более глубокого изучения SEMAT *Essence*.

Готовится к опубликованию статья «Скрам, усиленный *Essence*» (Park J. S., McMahon P. E., Myburgh B. "Scrum Powered by Essence"), посвященная использованию существующей практики скрама на ядре *Essence* и тому, какие это дает преимущества по сравнению со стандартным "Руководством по Скраму". Все больше и больше популярных практик будут описываться с использованием ядра *Essence*, после чего их можно будет сравнивать и комбинировать, чем и должны воспользоваться специалисты-практики.

Библиотека практик. После оформления ядра и языка модельные практики отбираются и добавляются в библиотеку с открытым доступом. Такая библиотека должна обеспечивать функцию обмена практиками, созданными для разных целей и в разных географических регионах, а также их публикацию в каталоге, поддерживаемом SEMAT.

Открытое стандартное ядро и язык являются необходимым условием для опубликования, каталогизации и обращения практик. Пространство для обраще-

ния практик должно таким образом предоставлять разработчикам преимущественный и беспрепятственный доступ к усовершенствованному, гибкому и доступному окружению, в котором разработчикам предоставлена соответствующая свобода выбора наиболее предпочтительных технологий, соответствующих их специфическим условиям. В свободный доступ будут выложены не только проверенные временем практики, но и составляющие им конкуренцию новые, инновационные идеи.

Инструментальные средства. Что касается разработки инструментальных средств, помогающих освоить ядро *Essence*, то ряд таких средств уже был создан в компании *III*. Они находятся в свободном доступе, но работа еще не завершена, и деятельность на этом направлении продолжается. В настоящее время корейская компания *uEngine Solutions*, предоставляющая программные средства с открытым исходным кодом, начала создание основанного на *Essence* инструментального средства для авторской разработки практик и гармоничного комбинирования методов. Результаты работы планируется выставить в открытом доступе. Ожидается, что разработчики инструментальных средств в скором будущем представят инструментальные средства для *Essence*, интегрированные с уже существующими или новыми инструментальными средствами, которые будут ориентированы, прежде всего, на профессионалов программирования, например, инструментарий для управления жизненным циклом (учет требований, проектирование, написание кода, тестирование, интеграция и т. д.), инструментальные средства для конфигурационного контроля и управления версиями, инструменты для управления проектом и т. д. Есть надежда, что инструментальные средства, применяемые на разных стадиях разработки программной системы, станут взаимозаменяемыми как раз благодаря тому, что разработчики инструментальных средств будут использовать в своей работе элементы ядра и язык. В основном подобные инструментальные средства рассчитаны на специалистов-практиков, которые более всего заинтересованы в разработке качественного программного обеспечения, удовлетворяющего всем требованиям заказчиков.

Опрос. Для исследования того, насколько хорошо специалисты-практики понимают назначение *Essence*, был создан специальный опросный лист, который, в частности, содержал следующие вопросы: где и как вы используете *Essence*, какие трудности вы встретили при переходе на *Essence* и т. д. Чтобы держать руку на пульсе и понимать реальные проблемные вопросы и потребности специалистов-практиков, такой опрос будет проводиться регулярно.

SEMAT-обучение

В рамках SEMAT должны быть организованы и продвинуты учебные программы. Только в этом случае будущие пользователи смогут разобраться в этой

концепции и станут применять ее на практике. Такая учебная инициатива должна охватывать несколько тематических блоков. К их числу относятся перечисленные далее.

✓ Введение в программную инженерию — основные понятия о ядре и т. д. Книга «Основа программной инженерии» [4] с этих позиций является образцом для подражания и должна получить продолжение. Особое внимание следует уделить дистанционному обучению на основе ее положений.

✓ Освоение практик на языке ядра — практическое обучение написанию практик с использованием примеров из реального опыта.

✓ Оценка продвижения проекта по созданию программного продукта и его «здоровья» — использование ядра и альфа для такой оценки проекта. Представляется, что существует два разных способа сделать это: внешний — пригласить аудитора/тренера, и внутренний — обеспечить команде возможность самообучаться в процессе работы.

✓ Создание и сравнение методов.

Накопленный опыт даст стимул к появлению новых, более систематизированных учебных материалов по SEMAT, пригодных для использования как в научной сфере, так и в промышленности. Появятся учебные планы по ядру, языку, практикам и методам, которые можно будет использовать для развития компьютерной науки, преподавания программной инженерии в университетах и проведения корпоративных тренингов для специалистов-практиков.

Будут созданы новые учебники и справочная литература, рекомендованные для профессионального обучения. Они же могут использоваться для самообучения работе с ядром и языком ядра. Материалы должны размещаться в свободном доступе. Использование основанных на ядре практик станет темой для большого числа книг, которые должны отличаться по сложности, т. е. удовлетворять потребностям профессионалов разного уровня квалификации, но быть им одинаково полезными и способствовать совершенствованию их навыков.

SEMAT-сообщество

SEMAT Inc. В целях развития коммерческой деятельности отдел SEMAT по управлению/коммуникациям недавно зарегистрировал SEMAT в качестве некоммерческой организации в штате Флорида (США). Проведена регистрация юридического лица *SEMAT Incorporated*. В настоящее время компания занимается разработкой внутренних документов, устанавливающих правила ведения оперативной деятельности. Тот же отдел занимается переделкой сайта SEMAT, который вскоре должен стать удобной платформой для общения, размещения информации и электронной торговли.

Участие всех и каждого. Приобретение статуса стандарта и издание книги по *Essence* (с учетом всех ее переводов) стали важными вехами в развитии проек-

та. Они должны использоваться для поддержания интереса существующих и привлечения новых подписчиков из рядов научного и бизнес-сообществ.

Весьма вероятно, что непрекращающееся создание новых интеллектуальных активов — практик, методов, инструментальных средств, обучающих и практических материалов, исследовательских статей — является залогом будущего роста SEMAT на нынешнем этапе его жизненного цикла. SEMAT постоянно нуждается в новых участниках, которые способны создавать *Essence*-практики, писать руководства для пользователей *Essence*, руководства по созданию практик, глоссарии, практические материалы, вести интересные блоги и т. д. Рост объема интеллектуальных активов должен, словно гравитационное поле, притягивать новых пользователей со всего мира.

В то же время стало очевидно, что высококвалифицированные специалисты готовы уделять SEMAT лишь ограниченное количество своего времени до тех пор пока он развивается на добровольной основе. В связи с этим обстоятельством появилась необходимость в финансовой поддержке для вознаграждения активных участников. Отсутствие минимально необходимых средств приведет проект в "спираль смерти", когда медленный рост активов вызовет снижение скорости прироста новых пользователей и, следовательно, отток специалистов, участвующих сейчас в SEMAT на добровольной основе. Они предпочтут посвятить свое время чему-то другому.

Прибыль можно генерировать, установив, например, определенную абонентскую плату за доступ к таким ценным активам, как описание практик, ситуационные исследования, электронные учебные материалы, инструментальные средства и т. д. Часть прибыли от реализации платного доступа к таким активам могла бы быть передана их создателям и тем, кто ведет сайт. SEMAT также должен обратиться за финансовой помощью и спонсорством к основным компаниям-разработчикам программного обеспечения и ассоциациям пользователей, которые уже применяют методологию *Essence* для производства своих продуктов, предоставления услуг или проведения внутренних операций.

Взаимодействие с региональными отделениями SEMAT. За взаимодействие с существующими отделениями и поддержку их повседневной работы отвечает отдел по управлению и коммуникациям. SEMAT должен помочь активным участникам в других регионах, например, в Европе и США, открыть местные региональные отделения для расширения состава разработчиков и пользователей SEMAT. Следует выйти на неисследованные пока регионы, такие как Южная Азия и Ближний Восток.

Публикации, публикации и еще раз публикации и книги о языке, и еще больше книг о применении *Essence* — все это очень нужно для подпитки сообщества.

Региональные отделения SEMAT осуществляют свою деятельность по следующим направлениям:

- ✓ предоставление ресурсов для различных отделов SEMAT;
- ✓ перевод книги *Essence*;
- ✓ сотрудничество с образовательными учреждениями для организации обучающих программ;
- ✓ поддержка исследователей и разработчиков;
- ✓ помощь в развитии библиотеки практик;
- ✓ повышение информированности о SEMAT в своих регионах путем организации конференций, симпозиумов, семинаров, ассоциаций пользователей и т. д.

* * *

Успех SEMAT можно измерить, оценив его положительное влияние на процесс программной инженерии среди практиков и среди корпораций. Формулой оценки успеха представляется формула «лучше, быстрее и счастливее» (*Better, Faster and Happier — BFH*). В широком смысле, параметр «лучше» указывает на ограничение риска возникновения дефектов, уменьшение объема работы, которую приходится переделывать, и повышение уровня эффективности устранения дефектов по сравнению с сегодняшними нормами. «Быстрее» означает улучшение скорости прохождения цикла разработки программного обеспечения по сравнению с существующими нормами. Термин «счастливее» указывает на повышение степени удовлетворения заказчика и работника за многолетний период, ограниченный сроком владения, который, в свою очередь, определяется совокупной стоимостью владения.

Для специалистов-практиков «лучше» означает более высокую конкурентоспособность — они разрабатывают программное обеспечение лучшего качества и умеют использовать больше практик. «Быстрее» — способность быстрее учиться и быстрее выполнять свою работу. «Счастливее» указывает на большую уверенность в себе и способность легче перемещаться из одной компании в другую. Для топ-менеджеров из промышленности «лучше» — это синоним инноваций, а следовательно, способности создавать высококачественное программное обеспечение и возможности объективно оценивать «здоровье» проекта. За выражением «быстрее» стоит сокращение времени на выдачу готового продукта заказчику. За «счастливее» — сотни удовлетворенных клиентов и десятки довольных работников.

SEMAT был придуман для воссоздания программной инженерии на новой основе. Конечно, прогресс выполнения такой задачи сложно измерить, но тем не менее необходимо это сделать. Совершенно закономерно, что следует оценить вклад продуктов SEMAT в дело улучшения качества создаваемого программного обеспечения. Такая оценка может быть сделана на основании двух дополняющих (а иногда даже пере-

крывающих) друг друга критериев — качественного и количественного. Результаты оценки по этим критериям будут представлены в следующих статьях, так как пока объем исходных данных для подобной оценки еще недостаточен.

В завершение следует отметить, что в настоящее время SEMAT претворяет в жизнь глобальную концепцию — процесс, начатый самим сообществом. Путь, лежащий перед растущим сообществом, полон трудностей. Однако трудности — это всего лишь завуалированные возможности. Успех SEMAT будет означать качественное повышение эффективности работы сторонников программной инженерии. SEMAT будет представлять собой площадку для создания более эффективных, гармоничных и мощных программных продуктов, разработанных под девизом «лучше, быстрее и счастливее». Возможно, именно Уоттс Хамфри (*Watts Humphrey*) на первой встрече SEMAT в марте 2010 г. в Цюрихе вернее других оценил реальный потенциал SEMAT: «по исторической важности нашу сегодняшнюю встречу можно сравнить с конфе-

ренцией НАТО 1968 г. в Гармише». Результаты SEMAT достигнуты профессиональным сообществом на благо самого сообщества. Вместе мы воссоздаем программную инженерию.

Список литературы

1. **Jacobson I., Meyer B., Soley R.** The SEMAT Initiative: A Call for Action // *Dr. Dobb's Journal*. 10 December 2009. URL: <http://www.drdoobs.com/architecture-and-design/the-semat-initiative-a-call-for-action/222001342>
2. **Jacobson I., Meyer B., Soley R.** Software engineering method and theory — a vision statement. URL: <http://blog.paluno.uni-due.de/semat.org/wp-content/uploads/2012/03/SEMAT-vision.pdf>
3. **Foundation** for the Agile Creation and Enactment of Software Engineering Methods (FACESEM). OMG, 2011. URL: <http://www.omg.org/cgi-bin/doc?ad/2012-11-01>
4. **Jacobson I., Ng P.-W., McMahon P. E., Spence I., Lidman S.** *The Essence of Software Engineering: Applying the SEMAT Kernel*. Addison Wesley, 2013. 224 p.
5. **Jacobson I., Huang S., Kajko-Mattsson M., McMahon P. E., Seymour E.** SEMAT—Three Year Vision // *Programming and Computer Software*. 2011. Vol. 38, N. 1. P. 1—12.

J. S. Park, Professor, Korea Advanced Institute of Science and Technology, Daejeon, President, SEMAT Inc., South Korea,

I. Jacobson, Professor, Chairman, Ivar Jacobson International, Switzerland,

B. Myburgh, MBL, Joburg Centre for Software Engineering, University of the Witwatersrand, Johannesburg, South Africa,

P. Johnson, Professor, Royal Institute of Technology, Stockholm, Sweden

SEMAT Yesterday, Today and Tomorrow. An Industrial Perspective

SEMAT Yesterday, Today and Tomorrow provides an historical overview of where SEMAT started, achievements of the past, today's priorities and plans for the future. The paper is aimed primarily at readers from Industry. While brief entries have been made about software engineering theory, the paper does little to develop discussion about theory. This will be done in a separate publication.

Keywords: SEMAT, Essence, kernel, alpha, method, practice

References

1. **Jacobson I., Meyer B., Soley R.** The SEMAT Initiative: A Call for Action. *Dr. Dobb's Journal*. 10 December 2009. URL: <http://www.drdoobs.com/architecture-and-design/the-semat-initiative-a-call-for-action/222001342>
2. **Jacobson I., Meyer B., Soley R.** Software engineering method and theory — a vision statement. URL: <http://blog.paluno.uni-due.de/semat.org/wp-content/uploads/2012/03/SEMAT-vision.pdf>
3. **Foundation** for the Agile Creation and Enactment of Software Engineering Methods (FACESEM). OMG, 2011. URL: <http://www.omg.org/cgi-bin/doc?ad/2012-11-01>
4. **Jacobson I., Ng P.-W., McMahon P. E., Spence I., Lidman S.** *The Essence of Software Engineering: Applying the SEMAT Kernel*. Addison Wesley, 2013. 224 p.
5. **Jacobson I., Huang S., Kajko-Mattsson M., McMahon P. E., Seymour E.** SEMAT — Three Year Vision. *Programming and Computer Software*. 2011. Vol. 38, N. 1. P. 1—12.

Глоссарий Essence

Версия 1.0

В данном документе представлены термины и определения Essence и их перевод на русский язык (из раздела 4 первой бета-версии стандарта OMG Essence, часть 4, стр. 4), осуществленный рабочей группой российского отделения SEMAT в составе А. Байды, И. Галахова, Ю. Купринова, Б. Позина, Г. Савельева, А. Сверчкова, А. Чудова, В. Швея.

Термины и определения	Перевод на русский язык
Activity An activity defines one or more kinds of work items and gives guidance on how to perform these	Действие Действие определяет один или более видов единиц работ и дает указания по их выполнению
Activity space A placeholder for something to be done in the software engineering endeavor. A placeholder may consist of zero to many activities	Пространство действий Указание на что-то, что должно быть сделано в рамках деятельности по разработке программного обеспечения. Пространство действий может включать в себя несколько действий или не включать ни одного
Alpha An essential element of the software engineering endeavor that is relevant to an assessment of the progress and health of the endeavor. Alpha is an acronym for an Abstract-Level Progress Health Attribute	Альфа Обязательный элемент программно-инженерной деятельности, относящийся к оценке прогресса и состояния "здоровья" деятельности. Альфа является аббревиатурой английского словосочетания "Атрибут абстрактного уровня прогресса и здоровья" (Abstract-Level Progress Health Attribute)
Alpha association An alpha association defines a relationship between two alphas	Связь альф Связь альф определяет отношения между двумя альфами
Area of concern Elements in kernels or practices may be divided into a collection of main areas of concern that a software engineering endeavor has to pay special attention to. All elements fall into at most one of these	Область интересов Элементы ядер или практик могут быть разделены на ряд основных областей интересов, которым в рамках деятельности по разработке программного обеспечения следует уделять особое внимание. Каждый элемент попадает в одну и только одну область интересов
Check list item A check list item is an item in a check list that needs to be verified in a state	Элемент контрольного списка Элемент в контрольном списке, состояние которого должно быть проверено
Competency A characteristic of a stakeholder or team member that reflects the ability to do work. A competency describes a capability to do a certain job. A competency defines a sequence of competency levels ranging from a minimum level of competency to a maximum level. Typically, the levels range from 0 — assists to 5 — innovates	Компетенция Характеристика представителя заинтересованной стороны или члена команды, которая отражает способность выполнять работу. Компетенция описывает возможность выполнять определенную работу. Компетенция определяет последовательность уровней компетентности — от минимального уровня компетентности до максимального уровня. Обычно, это уровни в диапазоне от 0 — "помогает" до 5 — "улучшает"
Constraints Restrictions, policies or regulatory requirements the team must comply with	Ограничения Условия, правила и нормативные требования, которые команда обязана соблюдать
Enactment The act of applying a method for some particular purpose, typically an endeavor	Исполнение Акт применения метода с каким-либо конкретным замыслом, обычно в рамках деятельности
Endeavor An activity or set of activities directed towards a goal	Деятельность Действие или набор действий, направленных на достижение цели
Kernel A kernel is a set of elements used to form a common ground for describing a software engineering endeavor	Ядро Ядро представляет собой набор элементов, используемых для формирования общей основы для описания деятельности по созданию программного обеспечения

Термины и определения	Перевод на русский язык
<p>Method A method is a composition of practices forming (at the desired level of abstraction) a description of how an endeavor is performed. A team's method acts as a description of the team's way-of-working and provides help and guidance to the team as they perform their task. The running of a development effort is expressed by a used method instance. This instance holds instances of alphas, work products, activities, and the like that are the outcome from the real work performed in the development effort. The used method instance includes a reference to the defined method instance, which is selected as the method to be followed</p>	<p>Метод Метод представляет собой композицию из практик, формирующую (на желаемом уровне абстракции) описание того, как выполняется деятельность. Метод команды описывает способ работы команды, помогает и направляет команду при выполнении ее задач. Ведение деятельности по разработке заключается в использовании экземпляра метода. Этот экземпляр содержит экземпляры альф, рабочих продуктов, действий и т. п., которые являются результатом реальной работы, выполняемой в процессе разработки. Используемый экземпляр метода включает ссылку на определенный экземпляр метода, который выбран в качестве метода, которому необходимо следовать</p>
<p>Opportunity The set of circumstances that makes it appropriate to develop or change a software system</p>	<p>Возможность Стечение обстоятельств, которое делает целесообразным разработку или изменение программной системы</p>
<p>Pattern A pattern is a description of a structure in a practice</p>	<p>Паттерн Паттерн представляет собой описание структуры практики</p>
<p>Practice A repeatable approach to doing something with a specific purpose in mind. A practice provides a systematic and verifiable way of addressing a particular aspect of the work at hand. It has a clear goal expressed in terms of the results its application will achieve. It provides guidance to not only help and guide practitioners in what is to be done to achieve the goal but also to ensure that the goal is understood and to verify that it has been achieved</p>	<p>Практика Повторяемый подход к совершению действий с определенным замыслом. Практика обеспечивает систематический и проверяемый способ выполнения конкретного аспекта работы. Она имеет четкую цель, выраженную в терминах результатов, которые ее применение позволит достичь. Она не только содержит указания, помогающие и направляющие исполнителей к тому, что должно быть сделано для достижения цели, но также и обеспечивает верное понимание цели и проверку того, что она была достигнута</p>
<p>Requirements What the software system must do to address the opportunity and satisfy the stakeholders</p>	<p>Требования То, что программная система должна делать для того, чтобы воспользоваться возможностью и удовлетворить заинтересованные стороны</p>
<p>Role A set of responsibilities</p>	<p>Роль Набор обязанностей</p>
<p>Software system A system made up of software, hardware, and data that provides its primary value by the execution of the software</p>	<p>Программная система Система, состоящая из программного и аппаратного обеспечения и данных, главная ценность которой создается посредством исполнения программного обеспечения</p>
<p>Stakeholders The people, groups or organizations who affect or are affected by a software system</p>	<p>Заинтересованные стороны Люди, группы или организации, тем или иным образом влияющие на программную систему или затрагиваемые ею</p>
<p>State A state expresses a situation where some condition holds</p>	<p>Состояние Состояние выражает ситуацию, отвечающую определенным условиям</p>
<p>State Graph A state graph is a directed graph of states with transitions between these states. It has a start state and may have a collection of end states</p>	<p>Граф состояний Граф состояний представляет собой ориентированный граф с вершинами-состояниями и дугами-переходами между этими состояниями. Он имеет начальное состояние и может иметь набор конечных состояний</p>
<p>Team The group of people actively engaged in the development, maintenance, delivery or support of a specific software system</p>	<p>Команда Группа людей, активно вовлеченных в разработку, обслуживание, поставку, внедрение или поддержку конкретной программной системы</p>
<p>Transition A transition is a directed connection from one state in a state machine to a state in that state machine</p>	<p>Переход Переход — это направленная связь одного состояния конечного автомата с другим его состоянием</p>

Термины и определения	Перевод на русский язык
Way-of-working The tailored set of practices and tools used by a team to guide and support their work	Технология работы Адаптированный набор практик и инструментов, используемых командой для ведения и поддержки ее работы
Work Work is defined as all mental and physical activities performed by the team to produce a software system	Работа Работа определяется как все умственные и физические действия, осуществляемые командой для производства программной системы
Work item A piece of work that should be done to complete the work. It has a concrete result and it leads to either a state change or a confirmation of the current state. Work item may or may not have any related activity	Единица работы Часть работы, которую необходимо сделать, чтобы завершить работу. Имеет конкретный результат, приводящий либо к изменению состояния, либо к подтверждению текущего состояния. Единица работы может иметь или не иметь связанных с ней действий
Work product A work product is an artifact of value and relevance for a software engineering endeavor. A work product may be a document or a piece of software, but also other created entities such as: <ul style="list-style-type: none"> • creation of a test environment; • delivery of a training course. Work products may be created, modified, used or deleted during an endeavor	Рабочий продукт¹ Рабочий продукт — это артефакт, представляющий ценность и значимость для программно-инженерной деятельности. Это может быть документ, часть программного обеспечения, а также другие созданные сущности, например: <ul style="list-style-type: none"> • тестовое окружение; • обучающий курс. Рабочий продукт может быть создан, изменен, использован или уничтожен в ходе деятельности
¹ Это определение отсутствует в разд. 4 спецификации стандарта OMG 1.0 beta 1, здесь приводится определение из подразд. 9.3.3.8 спецификации.	

**Продолжается подписка на журнал
"Программная инженерия" на первое полугодие 2015 г.**

Оформить подписку можно через подписные агентства
или непосредственно в редакции журнала.

Подписные индексы по каталогам:

Роспечать — 22765; Пресса России — 39795

Адрес редакции: 107076, Москва, Стромьинский пер., д. 4,
редакция журнала "Программная инженерия"

Тел.: (499) 269-53-97. Факс: (499) 269-55-10. E-mail: prin@novtex.ru

Анализ алгоритма предотвращения перегрузок AIMD в сетях передачи данных

Приведены результаты анализа алгоритма AIMD протокола TCP версии NewReno, который является действующим стандартом сети Интернет. Построен кусочно-линейный случайный процесс, описывающий поведение размера скользящего окна протокола в условиях, когда события потерь данных образуют процесс восстановления. Доказано существование стационарных распределений размера скользящего окна для вложенной марковской последовательности и для исходного кусочно-линейного процесса и найдены их характеристические функции. Последние позволяют получить полное описание характеристик протокола AIMD. Также построена алгебраическая оценка стационарного математического ожидания размера скользящего окна, которая играет наиболее важную роль при решении задач проектирования и администрирования сетей передачи данных.

Ключевые слова: кусочно-линейный случайный процесс, процесс восстановления, сети передачи данных, протокол TCP

Введение

Появление всепроникающих, интерактивных, персонализированных, сверхскоростных сетей, устройств и систем глобального масштаба способствует развитию разнообразного содержания и широкого спектра сетевых услуг. Однако, несмотря на быстрый рост мощности аппаратных компонентов сетевой инфраструктуры, задача развития технологий организации и управления такими сетями, в частности в целях поддержания и управления производительностью сетевых приложений и вычислений, остается актуальной. Методы решения этой проблемы должны базироваться на результатах анализа и математического моделирования процессов передачи данных.

В настоящей статье приведены данные анализа алгоритма *Additive Increase Multiplicative Decrease* (AIMD) версии NewReno [1] протокола сетей передачи данных *Transmission Control Protocol* (TCP), который является действующим стандартом сети Интернет и контролирует сетевые соединения на уровне точка-точка. Используя сведения о доставке данных, поступающие от получателя, протокол TCP управляет скоростью отправки данных в целях эффективного использования доступных ресурсов сети (каналов связи и маршрутизаторов) и одновременно предотвращения их перегрузки. Именно на уровне точка-точка принимаются решения об отправке данных в сеть, что определяет производительность сетевого маршрута, видимую приложениями. При этом только протокол TCP помимо других функций осуществляет управление скоростью отправки данных, определяя уровень загруженности сетевого маршрута на основе сигналов обратной связи. Алгоритмы TCP, в частности AIMD,

успешно обеспечивали связность сети в условиях экспоненциального роста числа пользователей сети Интернет на протяжении более 20 лет. В то же время эти алгоритмы обладают недостатками, которые тормозят развитие ряда актуальных, в том числе мультимедийных, приложений. Как следствие, результаты анализа и моделирования свойств и характеристик протокола TCP могут стать основой как для эффективного управления существующей инфраструктурой, так и для эффективной модернизации ее базовых протоколов и других программных элементов. Алгоритм AIMD является одним из ключевых алгоритмов протокола TCP и включен в подавляющее большинство его реализаций. Этот алгоритм является исходной точкой при разработке, тестировании и настройке параметров экспериментальных версий протокола TCP (см., например, версии BIC и CUBIC [2]).

В сетях передачи данных протокол TCP выполняет две основные функции. Первая — обеспечение надежной доставки данных от отправителя к получателю. Для этого используется механизм подтверждения доставки. Получив от отправителя очередной сегмент данных, получатель отправляет подтверждение о его получении, содержащее номера последовательности доставленных сегментов. Это либо наибольший номер в непрерывной последовательности номеров успешно доставленных сегментов, либо номера недостающих сегментов в последовательности (механизм *Selective Acknowledgement*, SACK). Время, прошедшее с момента отправки сегмента данных до момента получения отправителем подтверждения о его доставке, называется временем кругового оборота и является случайной величиной.

Для того чтобы обеспечить максимально эффективное использование каналов связи и маршрутизаторов, протокол TCP использует так называемое скользящее окно — объем данных, которые отправитель может отправить в сеть, не дожидаясь подтверждения об их доставке. Очевидно, что пропускная способность протокола TCP будет расти с ростом размера окна, однако неограниченный рост скользящего окна приводит к росту потерь данных на сетевом маршруте, к перегрузке сетевой инфраструктуры в целом и далее к явлению, известному как *congestion collapse* [3]. Эту трудность решает вторая основная функция протокола TCP — управление потоком данных. С этой целью TCP контролирует значение скользящего окна, назначая его в соответствии с рядом специальных алгоритмов, один из которых алгоритм AIMD, исследуемый в настоящей работе. Обзор исследований, посвященных другим алгоритмам протокола, можно найти в работе [3].

Алгоритм AIMD предназначен для предотвращения перегрузки элементов инфраструктуры сетевого маршрута и в то же время он стремится к их наиболее эффективному использованию. Если доставка сегмента данных успешно подтверждена получателем, то AIMD трактует это событие как сигнал о возможности увеличения пропускной способности и увеличивает размер скользящего окна. Если получатель сообщает о потере данных, то отправитель трактует это событие как сигнал о приближающейся перегрузке сети и уменьшает размер скользящего окна в $1/\alpha$ раз, где $0 < \alpha < 1$.

Сегодня сеть Интернет является высоко диверсифицированной системой, использующей широкое множество различных по своим параметрам носителей сигнала. Последние, в свою очередь, определяют свойства сетевых маршрутов, а следовательно, оказывают существенное влияние на характеристики производительности TCP. К числу таких носителей, в частности, относятся традиционные стационарные сети; беспроводные сети, как иерархические (Wi-Fi), так и самоорганизующиеся (MANET или ad-hoc); радиорелейные сети (GPRS, 3G и 4G); высокоскоростные сети (оптоволоконные, MPLS); сети с высокой задержкой (спутниковые каналы), а также сети, объединяющие несколько классов носителей сигнала. Методы управления очередью, используемые маршрутизаторами, такие как Drop Tail, Active Queue Management, Random Early Detection и пр., также могут оказывать влияние на характеристики производительности TCP. Анализ вопросов адаптации TCP к различным сетевым средам является отдельной областью исследований для каждой из технологий, перечисленных выше, см., например, обзор [4].

Физические свойства носителей сигнала и алгоритмы управления очередью определяют свойства случайного потока потерь данных, видимые отправителем, которые, в свою очередь, являются важнейшим параметром аналитических моделей алгоритма AIMD.

В литературе используются два приведенных далее основных способа описания потока потерь данных.

1. Определяется последовательность случайных величин $\{\tau_n\}_n > 0$, где τ_n — это интервал времени между двумя последовательными событиями потерь данных.

2. Определяется последовательность случайных величин $\{s_n\}_n > 0$ где s_n — объем данных, отправленных последовательно без потерь. Например, если предполагается, что потери сегментов происходят независимо с вероятностью p , то распределение s_n определяется по схеме Бернулли.

В литературе обе отмеченные выше последовательности определяются и как дискретные, и как непрерывные, в зависимости от контекста и используемых методов анализа.

Основной интерес при анализе алгоритма AIMD представляют характеристики его пропускной способности и/или размера скользящего окна. В настоящее время в литературе получен ряд оценок для специальных предположений о свойствах потока потерь данных. Здесь наиболее цитируемыми являются следующие работы: работа [5], где оценка среднего размера скользящего окна получена в предположении о детерминированном потоке потерь данных; работа [6], где события потерь данных образуют процесс Бернулли. Представленный в этих работах подход развит в работе [7] для случая со специальным двухпараметрическим потоком потерь, являющимся обобщением процесса Бернулли.

В подавляющем большинстве работ поведение алгоритма AIMD описывается кусочно-линейным случайным процессом, что позволяет проводить анализ для более широких классов процессов потерь данных. В частности, в работе [8] предполагается, что поток $\{\tau_n\}_n > 0$ образует стационарный эргодический точечный процесс. Здесь авторами получены математическое ожидание и дисперсия вложенной случайной последовательности. В работе [9] используется обобщенный процесс Пуассона для определения последовательности $\{s_n\}$. В работе [10] получены преобразования Лапласа переходного распределения размера скользящего окна для случая, когда поток потерь является пуассоновским. Важное направление представляют работы, посвященные изучению асимптотических свойств соответствующих случайных процессов. Заметим также, что к настоящему времени опубликовано большое число работ, посвященных численным, имитационным моделям и экспериментальному изучению поведения AIMD в различных сетевых средах (см., например, работы [11]—[14]), а также работ, посвященных анализу взаимодействия n TCP-соединений в некотором сетевом фрагменте или маршрутизаторе, в частности, работы [15, 16]. Более подробный анализ проблем и результатов моделирования протокола TCP и, в частности, алгоритма AIMD, приведены в обзоре [3].

В настоящей работе построен кусочно-линейный процесс, который описывает эволюцию скользящего окна алгоритма AIMD в условиях, когда события по-

терь данных на последовательности $\{s_n\}_{n \geq 0}$, образуют процесс восстановления. Доказаны теоремы о существовании стационарных распределений кусочно-линейного случайного процесса и вложенной в него марковской последовательности. Получены характеристические функции стационарного распределения квадрата размера скользящего окна для последовательности вложенных моментов и исходного кусочно-линейного процесса соответственно, а также оценка стационарного математического ожидания размера скользящего окна кусочно-линейного процесса.

Далее в работе приведено описание кусочно-линейного процесса, даны основные определения, а также приведены данные анализа эргодических свойств кусочно-линейного процесса, описывающего эволюцию AIMD. Получено стационарное математическое ожидание вложенной последовательности и оценка стационарного математического ожидания кусочно-линейного процесса. В заключении перечислены основные результаты работы.

Основные определения

Следуя подходу, принятому в литературе (например, в работах [8–10] и др.), определим кусочно-линейный случайный процесс $X(t) \in \mathbf{R}^+$ — объем данных, которые источник может отправить в сеть без получения подтверждения от получателя в момент времени t . Пусть на интервалах $[\theta_n, \theta_{n+1})$, $n = 0, 1, \dots$ имеет место $X(t) = X(t_0) + b(t - t_0)$, $\forall [t_0, t] \subset [\theta_n, \theta_{n+1})$, где b^{-1} — математическое ожидание времени кругового оборота сегмента данных. В случайные моменты времени $\{\theta_n\}_{n \geq 0}$ процесс $\{X(t)\}_{t > 0}$ совершает скачок вида $X(\theta_{n+1}) = \alpha X(\theta_n)$, где $0 < \alpha < 1$. При этом последовательность

$$s_n = \int_{\theta_n}^{\theta_{n+1}} X(\tau) d\tau$$

(объем данных, последовательно доставленных получателю без потерь) образует процесс восстановления с

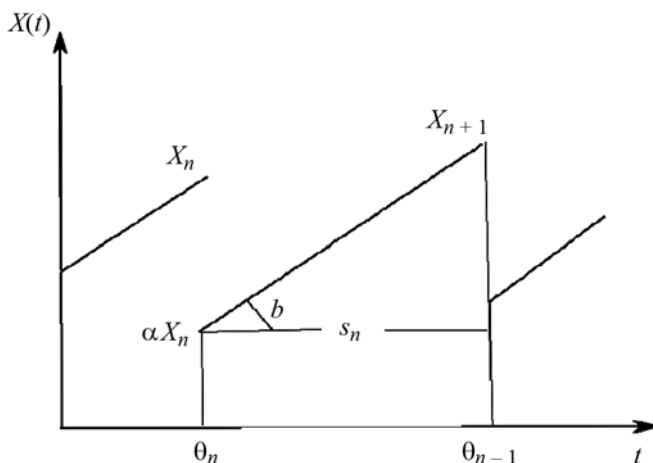


Рис. 1. Пример траектории размера скользящего окна

абсолютно непрерывной функцией восстановления $G(y) = P\{s_n \leq y\}$, имеющей конечные моменты первого и второго порядков, $E[s_n] = \lambda^{-1}$, $\lambda > 0$ и $E[s_n^2] < \infty$. Пример траектории такого процесса представлен на рис. 1.

Определим последовательность $\{X_n = X(\theta_n)\}_{n \geq 0}$. Заметим, что исходя из геометрических соображений, для последней имеет место соотношение

$$X_{n+1}^2 = \alpha^2 X_n^2 + 2bs_n. \quad (1)$$

Обозначим $Z_n = X_n^2$, тогда

$$Z_{n+1} = \alpha^2 Z_n + 2bs_n. \quad (2)$$

Для последовательности Z_n справедлива следующая теорема

Теорема 1. Последовательность $\{Z_n\}_{n > 0}$ имеет стационарное распределение, характеристическая функция которого

$$F(\zeta) = \prod_{k=0}^{\infty} G(2b\alpha^{2k}\zeta), \quad (3)$$

где $G(\zeta)$ — характеристическая функция $G(y)$.

Доказательство. Рассмотрим вероятности перехода последовательности марковских моментов Z_n , а именно условную функцию распределения

$$P\{Z(\theta_n) < x | Z_0 = y\}.$$

Применяя рекуррентно соотношение (2), получим

$$P\{Z_n < x | Z_0 = y_0\} = P\left\{\left(2b \sum_{k=0}^{n-1} \alpha^{2k} s_{n-k-1}\right) + y_0 \alpha^{2n} < x\right\}. \quad (4)$$

Заметим, что характеристическая функция распределения (4) имеет вид

$$F_n(\zeta) = \prod_{k=0}^{n-1} G(2b\alpha^{2k}\zeta) P_0(\alpha^{2n}\zeta), \quad (5)$$

где $P_0(\zeta)$ — характеристическая функция распределения Z_0 . При $n \rightarrow \infty$, правая часть (5) стремится к бесконечному произведению вида

$$F(\zeta) = \prod_{k=0}^{\infty} G(2b\alpha^{2k}\zeta). \quad (6)$$

Правая часть (6) сходится в собственном смысле, так как последовательность функций $\{G(2b\alpha^{2k}\zeta)\}$, $k = 0, 1, \dots$ удовлетворяет критерию, сформулированному в монографии [17, Теорема 3.7.3], и, следовательно, последовательность $\{Z_n\}_{n > 0}$ имеет стационарное распределение с характеристической функцией (3).

Теперь проведем анализ стационарного распределения случайного процесса $\{Z(t)\}_{t > 0} = \{X^2(t)\}_{t > 0}$. Оп-

ределим $B(\zeta)$ — характеристическую функцию распределения

$$Q(y) = \frac{\int_0^y [1 - G(t)] dt}{\int_0^{\infty} [1 - G(t)] dt}.$$

Справедлива следующая теорема.

Теорема 2. *Случайный процесс $\{Z(t)\}_{t>0}$ имеет стационарное распределение, характеристическая функция которого*

$$\Phi(\zeta) = B(2b\zeta) \prod_{k=1}^{\infty} G(2b\alpha^{2k}\zeta). \quad (7)$$

Доказательство. Определим

$$\tau = f(t) = \int_0^t X(\eta) d\eta.$$

Случайная величина $Z(t)$ с использованием соотношения (1) может быть представлена в виде

$$Z(t) = X^2(t) = (\alpha X_n + b(t - \theta_n))^2,$$

где θ_n — момент последнего скачка процесса $\{X(t)\}_{t>0}$ перед моментом времени t . Тогда

$$\begin{aligned} Z(t) &= \alpha^2 X_n^2 + 2X_n b\alpha(t - \theta_n) + b^2(t - \theta_n)^2 = \\ &= \alpha^2 Z_n + 2b\delta_n(t), \end{aligned} \quad (8)$$

где

$$\delta_n(t) = \frac{1}{2} (\alpha X_n + X(t))(t - \theta_n) = \delta(\tau)$$

— размер недоскока [18, § 2.7] процесса восстановления $\{s_n\}$ до уровня τ , где $f(\theta_n) \leq \tau < f(\theta_{n+1})$. Тогда, опираясь на правую часть (8), функция распределения $Z(t)$ может быть определена как сумма сверток распределений следующего вида:

$$\begin{aligned} P\{Z(t) \leq x\} &= \\ &= \sum_{n=0}^{\infty} \left[P\{N_{\tau} = n\} \int_0^x P\left\{Z_n < \frac{x - \xi}{\alpha^2}\right\} dP\left\{\delta(\tau) \leq \frac{\xi}{2b}\right\} \right], \end{aligned}$$

где N_{τ} — число событий восстановления процесса $\{s_n\}$ к моменту τ . Характеристическая функция последнего распределения будет иметь вид

$$\Phi(\zeta, t) = \sum_{n=0}^{\infty} [P\{N_{\tau} = n\} F_n(\alpha^2 \zeta) B_n(2b\zeta, t)], \quad (9)$$

где $B_n(\zeta, t)$ — характеристическая функция распределения $\delta(\tau)$.

Теперь заметим, что $\tau \geq t$, и $\tau \rightarrow \infty$, если $t \rightarrow \infty$, а следовательно, для процесса восстановления $\{s_n\}$ имеет место

$$\lim_{t \rightarrow \infty} P\{N_{\tau} = n\} = 0$$

для любого конечного n . Заметим также, что $F_n(\alpha^2 \zeta) = \frac{F_n(\zeta)}{G(2b\zeta)}$ и, следовательно, согласно теореме 1,

$$\lim_{n \rightarrow \infty} F_n(\alpha^2 \zeta) = \lim_{n \rightarrow \infty} \frac{F_n(\zeta)}{G(2b\zeta)} = \frac{F(\zeta)}{G(2b\zeta)}$$

в собственном смысле. Заметим также, что для процесса восстановления $\{s_n\}$ имеет место предел

$$\lim_{\tau \rightarrow \infty} P\{\delta(\tau) < y\} = \frac{\int_0^y [1 - G(t)] dt}{\int_0^{\infty} [1 - G(t)] dt}$$

согласно [18, § 2.7]. Учитывая перечисленное выше, получим

$$\lim_{t \rightarrow \infty} \Phi(\zeta, t) = B(2b\zeta) \prod_{k=1}^{\infty} G(2b\alpha^{2k}\zeta).$$

Следствие 1. *Если $G(y) = 1 - e^{-\lambda y}$, то $F(\zeta) = \Phi(\zeta)$.*

Для доказательства достаточно заметить, что в этом случае $Q(y) = G(y)$.

Выражение (7) позволяет вычислить все четные стационарные моменты процесса $\{X(t)\}_{t>0}$ для известного распределения $G(y)$ (или его известных моментов), а также, применив стандартные методы обращения интегральных преобразований, в этих случаях можно получить стационарное распределение случайных процессов $\{Z(t)\}_{t>0}$ и $\{X(t)\}_{t>0}$, ввиду того, что

$$P\{Z(t) < z\} = P\{X(t) < \sqrt{z}\}.$$

В частности, в работе [9] такой оригинал найден для показательной функции распределения $G(y)$ с параметром 1. Заметим, что доказательство теоремы 2 является конструктивным, так как формула (9) определяет нестационарное распределение процесса $\{Z(t)\}_{t>0}$ в случае, если известны или могут быть оценены характеристики процесса восстановления $\{s_n\}$.

Математическое ожидание кусочно-линейного процесса

Стационарное математическое ожидание процесса $\{X(t)\}_{t>0}$ вида

$$M_X = \lim_{t \rightarrow \infty} E[X(t)]$$

также может быть получено с помощью оригинала (7). Однако результаты таких вычислений, как правило,

являются весьма громоздкими, и вместе с тем M_X имеет существенное значение для дальнейших исследований и для приложений, связанных с проектированием и администрированием сетей. В настоящее время построен ряд оценок сверху для стационарного математического ожидания последовательности $\{X_n\}_{n \geq 0}$. Результаты, полученные выше, позволяют обобщить эти оценки для случая, когда потери данных описываются процессом восстановления. В настоящем разделе также построена оценка стационарного математического исходного кусочно-линейного процесса M_X .

Заметим, что

$$X_{n+1}^2 = X_n^2 + \gamma_n,$$

где $\gamma_n \in \mathbf{R}$. Для математических ожиданий получим

$$E[X_{n+1}^2] = E[X_n^2] + E[\gamma_n]. \quad (10)$$

Как следует из теоремы 1, левая часть равенства (10) имеет предел при $n \rightarrow \infty$, следовательно, его имеет и правая часть, причем

$$\lim_{n \rightarrow \infty} E[X_{n+1}^2] = \lim_{n \rightarrow \infty} E[X_n^2]$$

и значит, $\lim_{n \rightarrow \infty} E[\gamma_n] = 0$.

Обозначим s'_n величину (объем данных, отправленных без потерь), необходимую процессу $\{X(t)\}_{t > 0}$ для достижения уровня X_n с момента времени θ_n (рис. 2). Тогда

$$X_n^2 = \frac{2b}{1-\alpha} s'_n,$$

и в то же время

$$X_{n+1}^2 = \frac{2\alpha^2 b}{1-\alpha} s'_n + 2b s''_n,$$

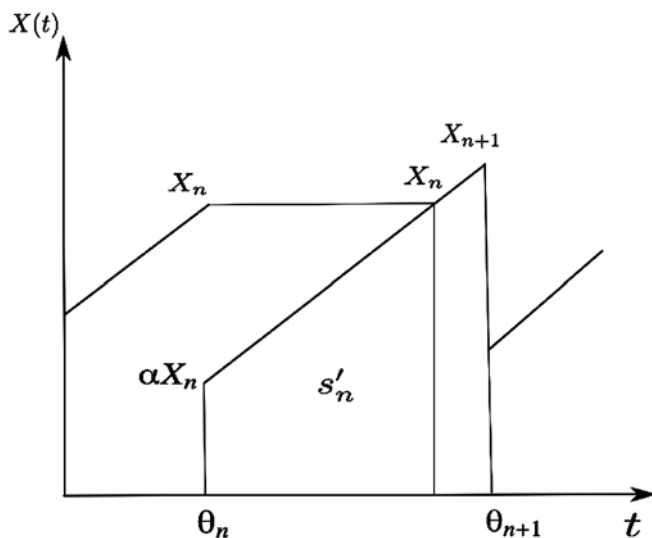


Рис. 2. Соотношение объемов отправленных данных

где $s''_n = s_n - s'_n$, $s''_n \in \mathbf{R}$ и $2b s''_n = \gamma_n$. Тогда

$$\begin{aligned} \lim_{n \rightarrow \infty} E[X_{n+1}] &= \lim_{n \rightarrow \infty} E\left[\frac{2b}{1-\alpha} s'_n + 2b s''_n\right] = \\ &= \lim_{n \rightarrow \infty} E\left[\frac{2b}{1-\alpha} s'_n\right] \end{aligned}$$

в силу того, что $\lim_{n \rightarrow \infty} E[2b s''_n] = \lim_{n \rightarrow \infty} E[\gamma_n] = 0$. В то же время имеет место $\lim_{n \rightarrow \infty} 2b E[s''_n] = \lim_{n \rightarrow \infty} 2b E[s_n - s'_n] = 2b(\lim_{n \rightarrow \infty} E[s_n] - \lim_{n \rightarrow \infty} E[s'_n]) = 0$.

Следовательно, $\lim_{n \rightarrow \infty} E[s'_n] = E[s_n] = \lambda^{-1}$. Окончательно получим неравенство

$$\lim_{n \rightarrow \infty} E[X_n] \leq \sqrt{\frac{2b}{1-\alpha^2} E[s_n]}.$$

Теперь построим оценку M_X . Используем соотношение (8), построенное при доказательстве теоремы 2, $X^2(t) = \alpha^2 X_n^2 + 2b \delta_n(t)$. Обозначим $i^{-1} B'(0) = d$. Тогда в силу теоремы 2 и неравенства Гельдера

$$M_X \leq \sqrt{\frac{2\alpha^2 b}{\lambda(1-\alpha^2)}} + 2db.$$

Заключение

В работе проведен анализ алгоритма AIMD протокола TCP NewReno [1] в условиях, когда объемы данных, доставленные получателю последовательно без потерь, образуют процесс восстановления. Доказаны теоремы о существовании стационарных распределений размера скользящего окна и марковской последовательности, вложенной в соответствующий кусочно-линейный случайный процесс. Найдены характеристические функции, описывающие стационарное распределение марковской последовательности и стационарное распределение кусочно-линейного процесса. Построены оценки стационарного математического ожидания кусочно-линейного процесса, а также стационарного математического ожидания вложенной марковской последовательности.

Список литературы

1. Allman M., Paxson V., Blanton E. TCP Congestion Control, RFC 5681, 2009. URL: <http://tools.ietf.org/html/rfc5681>
2. Ha S., Rhee I., Xu L. CUBIC: A New TCP-Friendly High-Speed TCP Variant // ACM SIGOPS Operating System Review. 2008. N. 42 (5). P. 64–74.
3. Afanasyev A., Tilley N., Reier P., Kleinrock L. Host-to-Host Congestion Control for TCP//IEEE COMMUNICATIONS SURVEYS & TUTORIALS. 2010. Vol. 12, N. 3. P. 304–342.
4. Khan N. I., Ahmed R., Aziz T. A Survey of TCP Reno, New Reno and SACK over mobile ad-hoc Network// International Journal of Distributed and Parallel Systems. 2012. Vol. 3, N. 1. P. 49–63.

5. Floyd S., Fall F. Promoting the use of end-to-end congestion control in the Internet // *IEEE/ACM Transactions on Networking*. 1999. Vol. 7 (4). P. 458–472.

6. Padhey J., Firoiu V., Towsley D., Kurose J. Modeling TCP Throughput: A Simple Model and its Empirical Validation // *IEEE/ACM Transactions on Networking*. 2000. Vol. 8, N. 2. P. 133–145.

7. Parvez N., Mahanti A., Williamson G. An Analytic Throughput Model for TCP NewReno // *IEEE/ACM Transactions on Networking*. 2010. Vol. 18, N. 2. P. 447–461.

8. Altman E., Avrachenkov K., Barakat C. A Stochastic model of TCP/IP with Stationary Random Losses // *Proceedings of ACM SIGCOMM'00 Conference*. Stockholm, 2000. P. 231–242.

9. Guillemin F., Robert P., Zwart B. AIMD Algorithm and exponential functionals // *Annals of Applied Probability*. 2004. N. 14 (1). P. 90–117.

10. Lopker A. H., Leeuwaarden J. S. H. Transient Moments of the TCP window size process // *Journal of Applied Probability*. 2008. N. 45 (1). P. 163–175.

11. Lestas M., Pitsillides A., Ioannou P., Hadjipollas G. A new estimation scheme for the effective number of users in internet

congestion control // *IEEE/ACM Transactions on Networking*. 2011. Vol. 19 (5). P. 1499–1512.

12. Barbera M., Lombardo A., Panarello C., Schembra G. Queue stability analysis and performance evaluation of a TCP-compliant window // *IEEE/ACM Transactions on Networking*. 2010. Vol. 18 (4). P. 1275–1288.

13. Li Y., Leith D., Shorten R. N. Experimental evaluation of TCP protocol for high-speed networks // *IEEE/ACM Transactions on Networking*. 2007. Vol. 15 (5). P. 1109–1122.

14. Eun D. Y., Wang X. Achieving 100 % Throughput in TCP/AQM Under Aggressive Packet Marking With Small Buffer // *IEEE/ACM Transactions on Networking*. 2008. Vol. 16 (4). P. 945–956.

15. Carofiglio M., Muscariello L. On the impact of TCP per-flow scheduling on internet performance // *IEEE/ACM Transactions on Networking*. 2012. Vol. 20 (2). P. 620–633.

16. Bacelli F., Carofiglio G., Foss S. Proxy caching in split TCP: dynamics, stability and tail asymptotics // *Proceedings of IEEE INFOCOM*. Phoenix, Arizona, USA, 2008. P. 131–135.

17. Лукач Е. Характеристические функции. М.: Наука, 1979.

18. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. М.: Радио и связь, 1987.

O. I. Bogoiavlenskaia, Associate Professor, e-mail: olbgvl@cs.karelia.ru, Petrozavodsk State University

Analysis of Congestion Avoidance Algorithm AIMD in the Data Communication Networks

This paper describes probabilistic analysis of NewReno version of TCP protocol AIMD algorithm which plays key role for congestion avoidance in data communication infrastructure. The model is formulated as a piecewise-linear stochastic process which presents the congestion window size and the data loss events are described by the renewal process with absolutely continuous renewal function. Markovian moments sequence embedded in the piecewise-linear process is defined. Its further analysis yields theorems proved on the ergodic properties of the Markovian moments sequence and the piecewise-linear process itself. Characteristic functions of sliding window size distribution for the sequence and the process are obtained as well. Reverse transforms of the characteristic functions yields complete set of AIMD characteristics. Also we expand estimation of the average congestion window size to the case of renewal data loss process.

Keywords: stochastic analysis, renewal process, data communication, AIMD, TCP protocol

References

1. Allman M., Paxson V. TCP Congestion Control, RFC 5681, 2009. URL: <http://tools.ietf.org/html/rfc5681>

2. Ha S., Rhee I., Xu L. CUBIC: A New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating System Review*. 2008. N. 42 (5). P. 64–74.

3. Afanasyev A., Tilley N., Reiher P., Kleinrock L. Host-to-Host Congestion Control for TCP. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*. 2010. Vol. 12, N. 3. P. 304–342.

4. Khan N. I., Ahmed R., Aziz T. A Survey of TCP Reno, New Reno and SACK over mobile ad-hoc Network. *International Journal of Distributed and Parallel Systems*. 2012. Vol. 3, No. 1. P. 49–63.

5. Floyd S., Fall F. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*. 1999. Vol. 7 (4). P. 458–472.

6. Padhey J., Firoiu V., Towsley D., Kurose J. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *IEEE/ACM Transactions on Networking*. 2000. Vol. 8, N. 2. P. 133–145.

7. Parvez N., Mahanti A., Williamson G. An Analytic Throughput Model for TCP NewReno. *IEEE/ACM Transactions on Networking*. 2010. Vol. 18 (2). P. 447–461.

8. Altman E., Avrachenkov K., Barakat C. A Stochastic model of TCP/IP with Stationary Random Losses. *Proceedings of ACM SIGCOMM'00*. Stockholm, 2000. P. 231–242.

9. Guillemin F., Robert P., Zwart B. AIMD Algorithm and exponential functionals. *Annals of Applied Probability*. 2004. N. 14 (1). P. 90–117.

10. Lopker A. H., Leeuwaarden J. S. H. Transient Moments of the TCP window size process. *Journal of Applied Probability*. 2008. N. 45 (1). P. 163–175.

11. Lestas M., Pitsillides A., Ioannou P., Hadjipollas G. A new estimation scheme for the effective number of users in internet congestion control. *IEEE/ACM Transactions on Networking*. 2011. Vol. 19 (5). P. 1499–1512.

12. Barbera M., Lombardo A., Panarello C., Schembra G. Queue stability analysis and performance evaluation of a TCP-compliant window. *IEEE/ACM Transactions on Networking*. 2010. Vol. 18 (4). P. 1275–1288.

13. Li Y., Leith D., Shorten R. N. Experimental evaluation of TCP protocol for high-speed networks. *IEEE/ACM Transactions on Networking*. 2007. Vol. 15 (5). P. 1109–1122.

14. Eun D. Y., Wang X. Achieving 100 % Throughput in TCP/AQM Under Aggressive Packet Marking With Small Buffer. *IEEE/ACM Transactions on Networking*. 2008. Vol. 16 (4). P. 945–956.

15. Carofiglio M., Muscariello L. On the impact of TCP per-flow scheduling on internet performance. *IEEE/ACM Transactions on Networking*. 2012. Vol. 20 (2). P. 620–633.

16. Bacelli F., Carofiglio G., Foss S. Proxy caching in split TCP: dynamics, stability and tail asymptotics, *Proc. IEEE INFOCOM 2008*, Phoenix, Arizona, USA, 2008. P. 131–135.

17. Лукач Е. *Characteristic Functions*. Москва: Наука, 1979.

18. Гнеденко Б., Коваленко И. *Введение в теорию массового обслуживания*. М.: Радио и связь, 1987.

Об уменьшении автоматной сложности за счет расширения регулярных языков

Рассмотрена проблема «экспоненциального взрыва» числа состояний конечного автомата, распознающего множество регулярных языков, задаваемых объединением регулярных выражений вида $*R_1 \cdot R_2^*$, где R_1 и R_2 — произвольные регулярные выражения. Предложен метод уменьшения числа состояний распознающего автомата за счет огрубления распознаваемого множества. Приведены неулучшаемые оценки на число состояний автомата при таком изменении в случае алфавита, состоящего из не менее чем пяти символов. Показано, что относительное уменьшение числа состояний может быть произвольным. Проанализирована практическая эффективность предложенного метода применительно к регулярным выражениям системы Snort.

Ключевые слова: конечные автоматы, регулярные выражения, сетевые системы обнаружения вторжений

Введение

С развитием технологий передачи данных одним из актуальных направлений обеспечения их информационной безопасности стали исследование и фильтрация сетевого трафика. На настоящее время существуют большое число различных сетевых систем обнаружения вторжений, таких как Snort [1], Bro [2], L7-filter [3], а также аппаратные продукты фирмы Cisco [4], базы сигнатур которых представляют собой наборы регулярных выражений. Вердикт о вредности трафика в таких системах выносится на основании соответствия фильтруемых данных хотя бы одному из регулярных выражений базы. Однако с ростом числа выражений возрастает и сложность проверки принадлежности слова заданному набору регулярных множеств.

Существует два основных подхода к оптимизации вышеупомянутой проверки. Первый подход — модификация традиционных детерминированного конечного автомата (ДКА) и недетерминированного конечного автомата (НКА). Так, например, в работе С. Кумара [5], в основе которой лежит алгоритм Ахо—Корасика [6], предложено использовать ДКА с сокращенным определенным образом числом состояний. В работе [7] предложено одновременно использовать два специально сформированных автомата — ДКА и НКА, благодаря которым объем требуемой для автомата памяти меньше, чем в случае одного ДКА. В работах [8, 9] предложено вводить специальные счетчики и битовые флаги, изменяемые в случае определенных переходов между состояниями, что также сокращает необходимый объем памяти.

Второй подход, который будет рассмотрен в данной статье, предполагает изменение исходного набора регулярных выражений таким образом, чтобы сократить сложность реализации в конечных автоматах за

счет расширения определяемого выражениями регулярного языка. Однако методы, предлагаемые в работах по данной теме, например, в статье [10], подразумевают лишь "ручное" переписывание выражений, когда специалист принимает решение об изменении каждого конкретного выражения. Основным результатом, который представлен в настоящей статье, является разработанный автором алгоритм автоматизированной модификации набора выражений, принадлежащих классу выражений вида $*R_1 \cdot R_2^*$ в нотации PCRE (*Perl compatible regular expressions*, Perl-совместимые регулярные выражения), где R_1 и R_2 — произвольные регулярные выражения.

В настоящей статье будет рассмотрена проблема экспоненциального роста числа состояний ДКА относительно числа заданных регулярных выражений и предложен алгоритм модификации таких выражений для сокращения числа состояний. Также будут даны некоторые оценки на число состояний автоматов для исходного и модифицированного набора выражений и представлены результаты применения описанного алгоритма к реально используемым выражениям.

Регулярные языки и выражения

В дополнение к традиционному определению регулярного выражения [11, 12] как задания набора операций объединения, конкатенации и "звезды Клини" над некоторым набором символов, далее будут использоваться следующие операции PCRE-совместимых регулярных выражений:

"." — символ, обозначающий объединение всех символов алфавита;

" $[a_1 a_2 \dots a_n]$ " — обозначение объединения символов a_1, a_2, \dots, a_n ;

" $[\wedge a_1 a_2 \dots a_n]$ " — обозначение объединения всех символов алфавита, кроме символов a_1, a_2, \dots, a_n ;

" $r\{m\}$ " — выражение, обозначающее повтор регулярного подвыражения " r " m раз.

Под регулярным языком над заданным алфавитом Σ далее будем подразумевать подмножество множества Σ^* , описываемое некоторым регулярным выражением. Заметим, что перечисленные выше операции не расширяют определение регулярного выражения, так как выводятся из базового набора операций, а значит классы регулярных языков, определяемых традиционными и расширенными регулярными выражениями, совпадают. Следовательно, будет верна теорема Клини [12] о том, что множество слов является регулярным языком тогда и только тогда, когда существует распознающий его ДКА.

Алгоритм объединения выражений

Основной сложностью, которая возникает при поиске по набору регулярных выражений, является большое число состояний конечных автоматов, реализующих поиск, что, в свою очередь, приводит к большим затратам по памяти.

Пусть есть n регулярных выражений вида $.*\alpha_i.*\beta_i.*$, где α_i и β_i — различные слова длины l из заданного языка Σ^* . Тогда число состояний реализующего ДКА достигает $O(nl2^n)$ [13]. Можно рассмотреть тривиальный пример для случая $n < |\Sigma|$. Возьмем выражения $.*a_i\{l\}.*a_i\{l\}.*$, где $i = \overline{1, n}$, а a_i — различные буквы алфавита Σ (совпадение слов α_i и β_i в рамках одного выражения не влияет на оценку числа состояний). Если рассмотреть приведенные (т. е. не содержащие неотличимых состояний) ДКА для выбранных выражений по отдельности, то у каждого автомата имеются две сильно связанные компоненты, не считая финального состояния. В данном примере при объединении ДКА число сильно связанных компонент перемножается. При этом, если объединены выражения i_1, \dots, i_k , то каждая компонента, пронумерованная элементом из $\{0, 1\}^k$, будет иметь вид, изображенный на рис. 1. Здесь штриховой линией обозначены переходы к другим сильно связным компонентам (если $c_j = 0$, то переход по символу a_{i_j} ведет к начальному состоянию сильно связной компоненты $(c_1, \dots, c_j + 1, \dots, c_k)$) или к

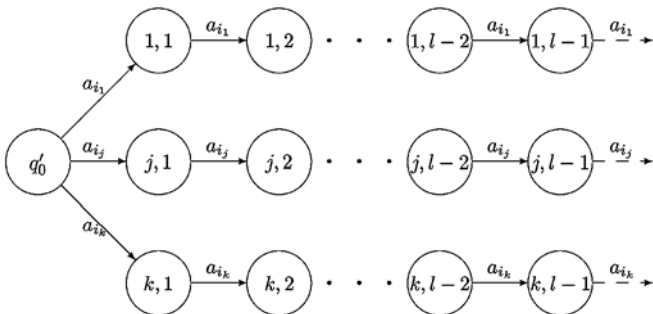


Рис. 1. Общий вид компоненты с меткой (c_1, \dots, c_k) , $c_j \in \{0, 1\}$

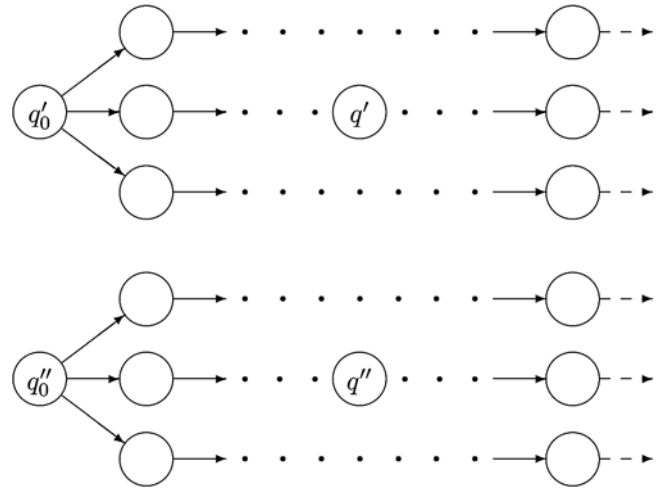


Рис. 2. Две различные компоненты с выбранными состояниями

финальному состоянию (если $c_j = 1$, то переход по символу a_{i_j} ведет в финальное состояние). Неотмеченные переходы с меткой a_{i_j} ведут в состояния q'_{i_j} , а остальные переходы — в начальное состояние компоненты q'_0 .

Сначала докажем неприводимость такого ДКА. Действительно, при такой конструкции состояния одной компоненты связности, очевидно, отличимы. Пусть состояния находятся в разных компонентах (рис. 2).

По построению найдется такое $m \in \overline{1, k}$, что слово $\underbrace{a_m \dots a_m}_l$ переводит состояние q'_0 в финальное, а q''_0 — нет (либо наоборот), т. е. q'_0 и q''_0 отличимы. Очевидно, что тогда q' также переводится этим словом в финальное состояние, а q'' переводится лишь в другую сильно связную компоненту (либо наоборот). Также из построения видно, что данный автомат принимает только слова вида $\alpha_1 \underbrace{a_{i_1} \dots a_{i_1}}_l \alpha_2 \underbrace{a_{i_2} \dots a_{i_2}}_l \alpha_3$, где $\alpha_j \in \Sigma^*$.

Число же состояний приведенного автомата равно $((l-1)k+1)2^k$, а в случае автомата для n выражений — $((l-1)n+1)2^n$ состояний.

Для сокращения числа состояний предлагается следующая замена: пусть $.*R'_1.*R''_1.*$ и $.*R'_2.*R''_2.*$ — два некоторых регулярных выражения из набора выражений вида $.*R'_i.*R''_i.*$, $i = \overline{1, n}$. Заменяем их на одно выражение $.*(R'_1|R''_1).(R'_2|R''_2)*$. В рассмотренном выше случае это приведет к тому, что сильно связанные компоненты вида (c_1, \dots, c_n) , где $c_{i_1} = 0$, $c_{i_2} = 1$ или $c_{i_1} = 1$, $c_{i_2} = 0$, будут отсутствовать, а переходы, которые в исходном случае вели в них, будут вести в компоненту с номером $(c_1, \dots, c_{i_1-1}, 1, c_{i_1+1}, \dots, c_{i_2-1}, 1, c_{i_2+1}, \dots, c_n)$.

Таким образом, число состояний сократится почти в 2 раза — до $((l-1)n+1)2^{n-1}$.

Однако такой подход может привести к увеличению множества слов, принимаемых полученным автоматом. Кроме того, хотя в данном примере модификация позволяет значительно сократить число состояний, в случае произвольных выражений это не всегда так. Далее будет показано, что для произвольных выражений сокращение числа состояний может быть как сколь угодно большим, так и сколь угодно малым.

Оценка числа состояний

Пусть R — регулярное выражение над алфавитом Σ , $L(R)$ — регулярный язык, определяемый выражением R , $V(L(R))$ — приведенный ДКА, принимающий язык $L(R)$, а $|V|$ — число состояний автомата V . Через $L^{pf}(R)$ обозначим такое наибольшее подмножество $L(R)$, что ни одно из слов из $L(R)$ не является нетривиальным префиксом для слов из $L^{pf}(R)$. Под нетривиальным префиксом в данном случае понимаем префикс, не совпадающий со всем словом. В случае если $L(R)$ содержит пустую строку Λ , положим $L^{pf}(R) = \{\Lambda\}$.

Докажем некоторые свойства множества $L^{pf}(R)$.

Лемма 1. Множество $L^{pf}(R)$ является регулярным языком.

Доказательство. Достаточно доказать, что существует ДКА V' , принимающий множество $L^{pf}(R)$. Тогда из теоремы Клини [11] будет следовать, что $L^{pf}(R)$ является регулярным языком.

Пусть $V(L(R))$ — ДКА, принимающий язык $L(R)$, со множеством финальных состояний A . Добавим в автомат нефинальное поглощающее состояние q_s и замкнем в него все состояния из A (рис. 3).

Докажем, что полученный автомат V' будет принимать $L^{pf}(R)$.

Пусть L' — множество, принимаемое автоматом V' . Тогда из построения следует, что $L^{pf}(R) \subseteq L'$. Действительно, пусть слово $\alpha \in L(R)$ такое, что $L(R)$ не содержит его нетривиальных префиксов. Тогда путь из начального состояния $V(L(R))$, соответствующий слову α , заканчивается в одном из финальных состояний, но не проходит ни через одно из них. Из построения V' следует, что состояния и переходы на пути слова α

остались без изменения, а значит путь в V' , соответствующий α , также приведет в финальное состояние. Докажем от противного, что ни одно из слов, принимаемых V' , не имеет нетривиальных префиксов из $L(R)$. Действительно, пусть α — такое слово, принимаемое V' , что $\alpha = \alpha'\alpha''$, где α' — минимальный префикс из $L(R)$, а α'' — непустое слово. Но так как слово α' — минимальный префикс из $L(R)$, то оно не имеет нетривиальных префиксов из $L(R)$, а значит, по доказанному выше, V' принимает α' . Но так как все финальные состояния автомата V' имеют переходы только в нефинальное поглощающее состояние, то, если V' принимает α' , то слово $\alpha'\alpha''$ не может быть принято, если α'' непустое слово. Противоречие.

Таким образом, множество $L^{pf}(R)$ принимается построенным автоматом V' , а следовательно, оно регулярно.

Лемма 2. Множество $L^{pf}(R)$ единственно.

Доказательство. Докажем от противного. Пусть L_1 и L_2 — различные подмножества $L(R)$, удовлетворяющие определению $L^{pf}(R)$. Из этого предположения следует, что множество $L_1 \cup L_2$ не содержит слов с префиксом из $L(R)$, иначе в L_1 или L_2 присутствует слово с префиксом из $L(R)$. Из того факта, что L_1 и L_2 различны, вытекает, что либо $L_1 \subset L_1 \cup L_2$, либо $L_2 \subset L_1 \cup L_2$. Однако тогда одно из множеств не максимально. Противоречие.

Лемма 3. Детерминированный конечный автомат $V(L^{pf}(R))$, где R — произвольное регулярное выражение, имеет всего одно финальное состояние, все переходы которого ведут в единственное нефинальное поглощающее состояние.

Доказательство. По построению из леммы 1 и единственности множества $L^{pf}(R)$ (лемма 2) получаем, что $V(L^{pf}(R))$ имеет нефинальное поглощающее состояние, причем все переходы из финальных состояний ведут в него. Таким образом, финальные состояния неотличимы, а значит, в приведенном ДКА оно всего одно.

Лемма 4. Детерминированный конечный автомат $V(L(. *R. *))$, где R — произвольное регулярное выражение, имеет одно финальное состояние, которое также является поглощающим, и не содержит нефинальных поглощающих состояний.

Доказательство. Заметим, что если слово α принимается автоматом $V(L(. *R. *))$, то $\alpha\beta$, где $\beta \in \Sigma^*$, также принимается автоматом. По этой причине любой путь из любого финального состояния проходит только через финальные состояния. Из этого следует, что финальные состояния неотличимы, т. е. финальное состояние всего одно и является поглощающим.

Докажем теперь, что нефинальных поглощающих состояний нет. Действительно, если такое состояние существует, возьмем слово $\alpha \in \Sigma^*$, переводящее автомат в него, и слово β из $L(R)$. Тогда слово $\alpha\beta$ переводит автомат $V(L(. *R. *))$ из начального состояния в нефинальное поглощающее состояние. Однако $\alpha\beta$ принадлежит языку $L(. *R. *)$, а значит, слово $\alpha\beta$ должно переводить рассматриваемый автомат в финальное состояние. Противоречие.

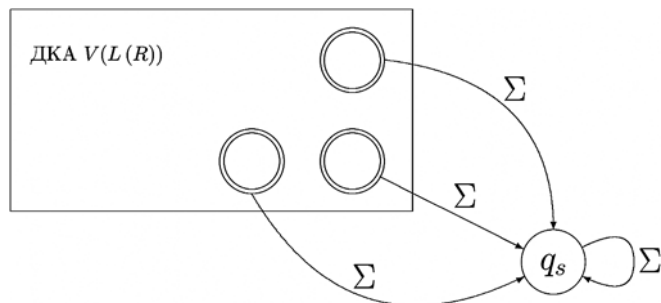


Рис. 3. Модификация автомата

Лемма 5. Приведенный ДКА, полученный из автомата $V(L^{pf}(*R))$ путем устранения нефинального поглощающего состояния и замыкания всех переходов финального состояния на себя, изоморфен автомату $V(L(*R^*))$.

Доказательство. Обозначим $V_1 = V(L(*R^*))$, $V_2 = V(L^{pf}(*R))$.

Пусть $|V_1| + 1 \geq |V_2|$. Исключим из автомата V_2 тупиковое состояние и замкнем финальное состояние. По построению полученный автомат V'_2 будет принимать язык $L^{pf}(*R)\Sigma^*$. Заметим, что $L^{pf}(*R)\Sigma^* = L(*R^*)$. Пусть $\alpha \in L(*R^*)$, тогда по определению L^{pf} существуют $\alpha' \in L^{pf}(*R)$, $\alpha'' \in \Sigma^*$ такие, что $\alpha = \alpha'\alpha''$. Следовательно, $L^{pf}(*R)\Sigma^* \supseteq L(*R^*)$. Очевидно, что $L^{pf}(*R)\Sigma^* \subseteq L(*R^*)$, а значит, $L^{pf}(*R)\Sigma^* = L(*R^*)$. Таким образом, V'_2 и V_1 принимают один и тот же язык. Из минимальности V_1 следует, что $|V_1| \leq |V'_2| = |V_2| - 1$, но по предположению $|V_1| + 1 \geq |V_2|$, а значит $|V_1| = |V_2| - 1 = |V'_2|$. По этой причине, в силу того, что число состояний у данных автоматов, принимающих один язык, совпадает, а также принимая во внимание, что один из автоматов минимален, автоматы изоморфны [11].

Пусть $|V_1| + 1 < |V_2|$. Модифицируем автомат V_1 , добавив тупиковое состояние и заменив переходы замкнутого финального состояния в себя на переходы в тупиковое состояние. Очевидно, что регулярный язык L' полученного автомата V'_1 входит в $L(*R^*)$. Докажем, что $L' = L^{pf}(*R)$. Из построения и того факта, что $L' \subseteq L(*R^*)$, следует, что L' свободен от префиксов и $L' \subseteq L^{pf}(*R)$. Докажем, что $L' \supseteq L^{pf}(*R)$. Действительно, пусть $\alpha \in L^{pf}(*R)$. Тогда из определения L^{pf} следует, что для любого нетривиального разбиения $\alpha = \alpha'\alpha''$: $\alpha' \in L^{pf}(L(*R))$. Как следствие, если рассмотреть путь, соответствующий слову α и начинающийся

в начальном состоянии V_1 , то можно заметить, что путь проходит через финальное состояние только один раз — в самом конце (иначе в $L(*R^*)$ присутствует префикс слова из $L^{pf}(*R)$), а значит, по построению V'_1 получаем, что $\alpha \in L'$. Таким образом, $L' = L^{pf}(*R)$. Из минимальности V_2 следует, что $|V_1| + 1 = |V'_1| \geq |V_2|$, но по предположению $|V_1| + 1 \leq |V_2|$, а значит, $|V_1| = |V_2| - 1 = |V'_1|$. Аналогично предыдущим рассуждениям получаем, что автоматы V'_1 и V_2 изоморфны. В таком случае, если у обоих автоматов убрать нефинальные поглощающие состояния, то получим изоморфные автоматы V_1 и V'_2 .

Следствие 1. Для произвольного регулярного выражения R верно, что

$$|V(L(*R^*))| + 1 = |V(L^{pf}(*R))|.$$

Лемма 6. Пусть заданы два регулярных выражения R_1 и R_2 , тогда

$$|V(L(*R_1 * R_2^*))| = |V(L^{pf}(*R_1))| + |V(L^{pf}(*R_2))| - 3.$$

Доказательство. Обозначим $V_1 = V(L^{pf}(*R_1))$, $V_2 = V(L^{pf}(*R_2))$. По лемме 4 автоматы V_1 и V_2 содержат по одному финальному и поглощающему нефинальному состоянию. Объединим их как показано на рис. 4, т. е. "сошьем" финальное состояние первого автомата q'_f и начальное состояние второго автомата q''_0 , сохранив переходы начального состояния, и замкнем финальное состояние второго автомата q''_f . В качестве начального состояния возьмем начальное состояние автомата V_1 . Очевидно, что язык, принимаемый новым автоматом V' , будет совпадать с $L(*R_1 * R_2^*)$.

Действительно, рассмотрим произвольный путь из начального состояния q'_0 в финальное состояние q''_f .

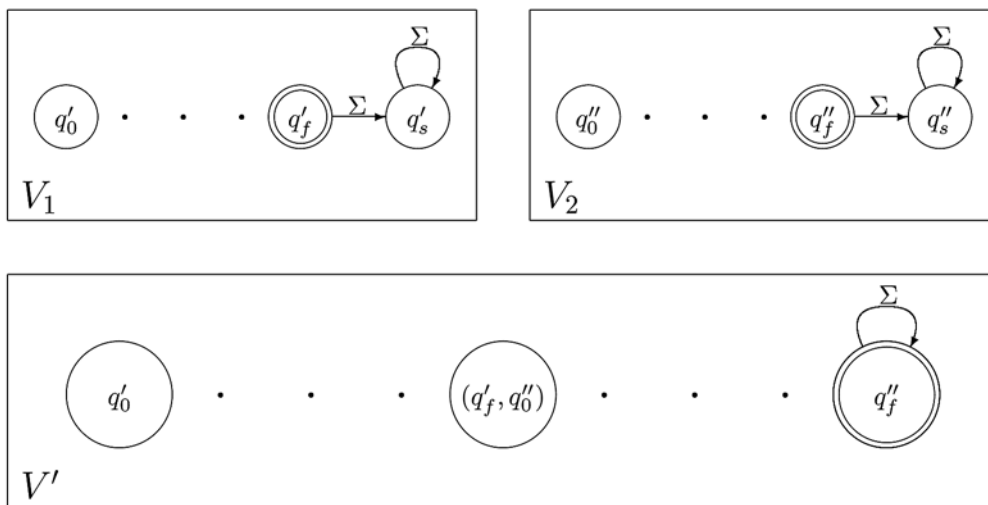


Рис. 4. Объединение двух автоматов

Такой путь обязан пройти через состояние (q'_f, q''_0) . Пусть слово $\alpha \in \Sigma^*$ соответствует части пути от q'_0 до (q'_f, q''_0) , а $\beta \in \Sigma^*$ соответствует оставшейся части пути. Тогда по построению $\alpha \in L^{pf}(. * R_1)$ и по лемме 5 $\beta \in L(. * R_2)$. Таким образом, $\alpha\beta \in L(. * R_1 * R_2)$, т. е. язык, принимаемый автоматом, не выходит за пределы $L(. * R_1 * R_2)$.

Пусть слово $\gamma \in L(. * R_1 * R_2)$. Среди всех разбиений $\gamma = \alpha\beta$ выберем такое, что $\alpha \in L^{pf}(. * R_1) \subseteq L(. * R_1)$, $\beta \in L(. * R_2)$. По построению путь из q'_0 , соответствующий слову α , приводит в состояние (q'_f, q''_0) , а путь из (q'_f, q''_0) , соответствующий β , приводит в финальное состояние q''_f . Таким образом, автомат V' принимает язык, совпадающий с $L(. * R_1 * R_2)$.

Докажем от противного, что полученный автомат является приведенным. Предположим, что существуют два различных неотличимых состояния q_1 и q_2 . Если q_1 и q_2 соответствуют состояниям исходного автомата V_2 , то по построению и по лемме 5 данные состояния не могут быть неотличимы. Пусть q_1 и q_2 соответствуют состояниям исходного автомата V_1 . Из минимальности автомата V_1 (различимости состояний в автомате V_1) следует, что существует такое слово $\alpha \in \Sigma^*$, которое переводит одно из состояний (без ограничения общности можем считать, что это q_1) в состояние с меткой (q'_f, q''_0) , а второе состояние переводит в некоторое отличное от (q'_f, q''_0) состояние, соответствующее нефинальному непоглощающему состоянию из V_1 . Пусть β — слово минимальной длины, принимаемое автоматом V_2 . Тогда утверждается, что слово $\alpha\beta$, переводящее состояние q_1 в финальное, различает состояния q_1 и q_2 . Предположим, что это не так и $\alpha\beta$ переводит q_2 в финальное состояние. Однако, так как по построению любой путь из состояния, соответствующего автомату V_1 , в финальное состояние обязательно проходит через состояние (q'_f, q''_0) , существует такое нетривиальное разбиение $\beta = \beta'\beta''$, что $\alpha\beta'$ переводит q_2 в (q'_f, q''_0) . Вместе с тем β'' переводит состояние (q'_f, q''_0) в финальное состояние (а значит, и принимается автоматом V_2), причем длина β'' меньше длины β , что противоречит выбору слова β . Пусть q_1 и q_2 соответствуют автоматам V_1 и V_2 . Через α обозначим слово минимальной длины, переводящее состояние q_1 в (q'_f, q''_0) , а через β — слово минимальной длины, принимаемое автоматом V_2 , т. е. $\beta \in L^{pf}(. * R_2)$. Из того, что любой путь из состояния, соответствующего автомату V_1 , проходит через (q'_f, q''_0) , следует, что $\alpha\beta$ — слово минимальной длины, переводящее q_1 в финальное состояние. Пусть γ — слово из Σ^* , которое

переводит (q'_f, q''_0) в q_2 . Так как $\beta \in L^{pf}(. * R_2)$, то $\gamma\beta \in L(. * R_2) \subseteq L(. * R_2)$. Тогда, очевидно, $\gamma\beta$ переводит (q'_f, q''_0) в финальное состояние. Поэтому β переводит q_2 в финальное состояние q''_f , но не переводит q_1 в q''_f , так как минимальная длина слова, переводящего q_1 в финальное состояние, равна длине слова $\alpha\beta$, т. е. строго больше длины слова β .

Таким образом, доказано, что построенный автомат минимален и, соответственно, с точностью до изоморфизма совпадает с $V(L(. * R_1 * R_2))$, а число его состояний в точности равно $|V(L^{pf}(. * R_1))| + |V(L^{pf}(. * R_2))| - 3$.

Лемма 7. Пусть R' и R'' — такие регулярные выражения, что автоматы $V(L(R'))$ и $V(L(R''))$ имеют по одному финальному состоянию, причем оба финальных состояния являются поглощающими. Тогда $|V(L(R') \cup L(R''))| \leq (|V(L(R'))| - 1)(|V(L(R''))| - 1) + 1$.

Доказательство. Пусть Q' и Q'' — множества состояний; q'_0 и q''_0 — начальные состояния; δ' и δ'' — функции переходов, а q'_f и q''_f — финальные состояния автоматов $V(L(R'))$ и $V(L(R''))$ соответственно.

Построим автомат V с $|Q'| + |Q''|$ состояниями, где каждому состоянию ставится в соответствие пара состояний $q' \in Q'$, $q'' \in Q''$. Для произвольного символа $a \in \Sigma$ зададим значение функции переходов как $\delta((q', q''), a) = (\delta'(q', a), \delta''(q'', a))$. В качестве начального состояния возьмем состояние (q'_0, q''_0) , а в качестве финальных — все состояния вида (q'_f, q''_f) и (q'_f, q''_f) , где $q' \in Q'$, $q'' \in Q''$. Такой автомат, очевидно, будет принимать язык $L(R') \cup L(R'')$. Так как финальные состояния V' и V'' являются поглощающими, то по построению финальные состояния автомата V также переходят только в финальные состояния, а следовательно, они неотличимы. Тогда в приведенном автомате V будет не больше $(|Q'| - 1)(|Q''| - 1)$ нефинальных состояний и одно финальное поглощающее состояние, т. е.

$$|V(L(R') \cup L(R''))| \leq (|Q'| - 1)(|Q''| - 1) + 1 = (|V(L(R'))| - 1)(|V(L(R''))| - 1) + 1.$$

Теорема 1. Пусть имеются два регулярных выражения $. * R_1 * R_2$ и $. * R_3 * R_4$ такие, что $|V(L^{pf}(. * R_i))| = n_i + 2$, $n_i \geq 1$. Тогда

$$\begin{aligned} |V(L(. * R_1 * R_2))| &= n_1 + n_2 + 1, \\ |V(L(. * R_3 * R_4))| &= n_3 + n_4 + 1, \end{aligned} \quad (1)$$

$$\begin{aligned} |V(L(. * R_1 * R_2) \cup L(. * R_3 * R_4))| &\leq \\ &\leq (n_1 + n_2)(n_3 + n_4) + 1, \end{aligned} \quad (2)$$

$$|V(L(. * (R_1 | R_3) * (R_2 | R_4)))| \leq n_1 n_3 + n_2 n_4 + 1. \quad (3)$$

Причем в случае $|\Sigma| \geq 5$ оценки (2) и (3), вообще говоря, неуплучшаемы.

Если $|\Sigma| \geq 3$, то для любого $C \in (0; 1] \cap \mathbb{Q}$, \mathbb{Q} — множество рациональных чисел, найдутся такие выражения R_1, R_2, R_3 и R_4 , что:

$$C = \frac{|V(L(. * (R_1 | R_3). * (R_2 | R_4). *))|}{|V(L(. * R_1. * R_2. *) \cup L(. * R_3. * R_4. *))|}. \quad (4)$$

Доказательство. Равенство (1) следует из леммы 6:

$$|V(L(. * R_i. * R_j. *))| = |V(L^{pf}(. * R_i))| + |V(L^{pf}(. * R_j))| - 3 = n_i + 2 + n_j + 2 - 3 = n_i + n_j + 1.$$

В силу того, что выражения $. * R_1. * R_2. *$ и $. * R_3. * R_4. *$ удовлетворяют условию леммы 4, к ним может быть применена лемма 7:

$$\begin{aligned} & |V(L(. * R_1. * R_2. *) \cup L(. * R_3. * R_4. *))| \leq \\ & \leq (|V(L(. * R_1. * R_2. *))| - 1)(|V(L(. * R_3. * R_4. *))| - 1) + 1 = \\ & = (n_1 + n_2)(n_3 + n_4) + 1. \end{aligned}$$

Пусть a, b, c, d и e — различные символы алфавита Σ . Тогда в качестве примера достижимости оценки можно рассмотреть следующие выражения: $R_1 = [ab]\{n_1 - 1\}d$, $R_2 = [ab]\{n_2 - 1\}d$, $R_3 = [ac]\{n_3 - 1\}e$, $R_4 = [ac]\{n_4 - 1\}e$. Все пары нефинальных состояний, из построения в доказательстве леммы 7, достижимы. Действительно, возьмем произвольное нефинальное состояние (i, j) .

Пусть $i < n_1, j < n_3$. Если $i \leq j$, то состояние достигается словом $c\{j - i\}a\{i\}$. Если $i > j$, то словом $b\{i - j\}a\{j\}$.

Пусть $i < n_1, j \geq n_3$. Если $i \leq j - n_3$, то состояние достигается словом $c\{n_3 - 1\}ec\{j - n_3 - i\}a\{i\}$. Если $i > j - n_3$, то словом $c\{n_3 - 1\}eb\{i - j + n_3\}a\{j - n_3\}$.

Пусть $i \geq n_1, j < n_3$. Если $i - n_1 \leq j$, то состояние достигается словом $b\{n_1 - 1\}dc\{j - i + n_1\}a\{i - n_1\}$. Если $i - n_1 > j$, то словом $b\{n_1 - 1\}db\{i - n_1 - j\}a\{j\}$.

Пусть $i \geq n_1, j \geq n_3$. Если $i - n_1 \leq j - n_3$, то состояние достигается словом $b\{n_1 - 1\}dc\{n_3 - 1\}ec\{j - n_3 - i + n_1\}a\{i - n_1\}$. Если $i - n_1 > j - n_3$, то словом $b\{n_1 - 1\}dc \times \{n_3 - 1\}eb\{i - n_1 - j + n_3\} \times a\{j - n_3\}$.

Докажем, что все пары нефинальных состояний отличимы. Возьмем две различные пары (i, j) и (i', j') . Пусть $i \neq i'$. Без ограничения общности можно считать, что $i < i'$, тогда если $i' < n_1$, то слово $b\{n_1 - 1 - i'\} \times db\{n_2 - 1\}d$ очевидным образом приводит состояние (i', j') в финальное, в отличие от (i, j) . Если $i' \geq n_1$, то различающим словом является слово $b\{n_1 + n_2 - 1 - i'\}d$. В случае $j \neq j'$, без ограничения общности считая, что

$j < j'$, различающими словами будут $c\{n_3 - 1 - j'\} \times dc\{n_4 - 1\}d$ и $c\{n_3 + n_4 - 1 - j'\}d$ соответственно.

Таким образом, оценка (2) неуплучшаема.

Докажем неравенство (3).

По лемме 6

$$\begin{aligned} & |V(L(. * (R_1 | R_3). * (R_2 | R_4). *))| = \\ & = |V(L^{pf}(. * (R_1 | R_3)))| + |V(L^{pf}(. * (R_2 | R_4)))| - 3. \end{aligned}$$

Докажем, что число состояний конечного автомата $V(L^{pf}(. * R_k | R_l))$ не превышает $n_k n_l + 2$. По лемме 7 число состояний простейшего объединения автоматов $V(L(. * R_k. *))$ и $V(L(. * R_l. *))$, которые имеют по одному финальному поглощающему состоянию, не превышает $(|V(L(. * R_k. *))| - 1)(|V(L(. * R_l. *))| - 1) + 1$, что, в свою очередь, по следствию 1 равно $|V(L^{pf}(. * R_k))| |V(L^{pf}(. * R_l))| + 1 = n_k n_l + 1$. Поэтому

$$\begin{aligned} |V(L^{pf}(. * (R_1 | R_3)))| & = V(L(. * (R_1 | R_3). *)) + 1 \leq n_1 n_3 + 2, \\ |V(L^{pf}(. * (R_2 | R_4)))| & = V(L(. * (R_2 | R_4). *)) + 1 \leq n_2 n_4 + 2. \end{aligned}$$

Причем для выражений $R_k = [ab]\{n_k - 1\}d$ и $R_l = [ac]\{n_l - 1\}e$ верно равенство $V(L(. * (R_k | R_l). *)) = n_k n_l + 1$. Действительно, состояние (i, j) , где $i < n_k$ и $j < n_l$, достижимо словом $b\{j - i\}a\{i\}$, если $i < j$, и словом $c\{i - j\}a\{j\}$ в противном случае.

Докажем отличимость нефинальных состояний (i, j) и (i', j') . Пусть $i < i'$ (случай $i > i'$ аналогичен), тогда слово $b\{n_1 - 1 - i'\}$ приводит (i', j') в финальное состояние, в отличие от (i, j) . В случае $j < j'$ (случай $j > j'$ аналогичен) различающим словом будет $c\{n_3 - 1 - j'\}e$.

Таким образом, получаем:

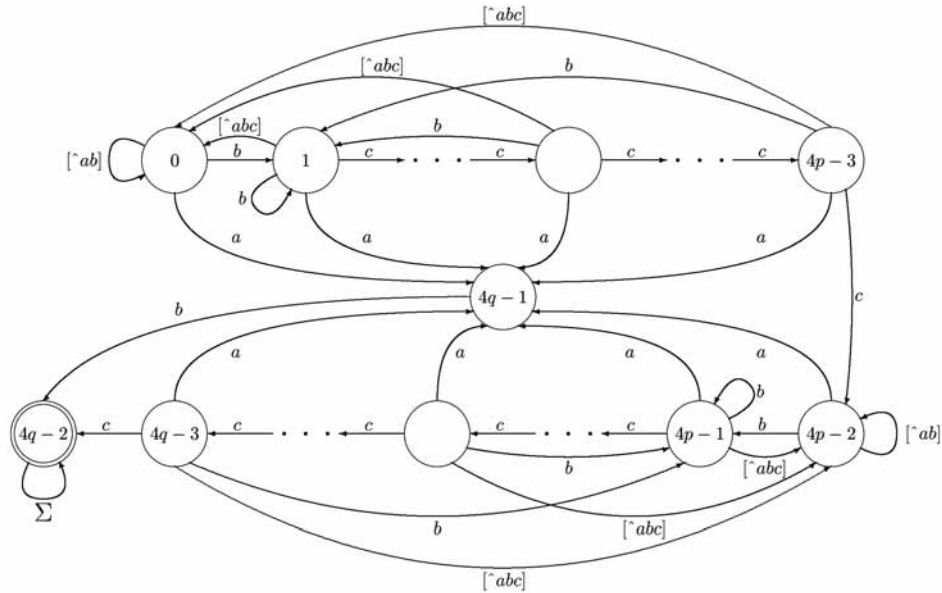
$$\begin{aligned} & |V(L(. * (R_1 | R_3). * (R_2 | R_4). *))| = \\ & = |V(L^{pf}(. * (R_1 | R_3)))| + |V(L^{pf}(. * (R_2 | R_4)))| - 3 \leq \\ & \leq n_1 n_3 + 2 + n_2 n_4 + 2 - 3 = n_1 n_3 + n_2 n_4 + 1. \end{aligned}$$

Причем равенство достигается при $R_1 = [ab]\{n_1 - 1\}d$, $R_2 = [ab]\{n_2 - 1\}d$, $R_3 = [ac]\{n_3 - 1\}e$, $R_4 = [ac]\{n_4 - 1\}e$.

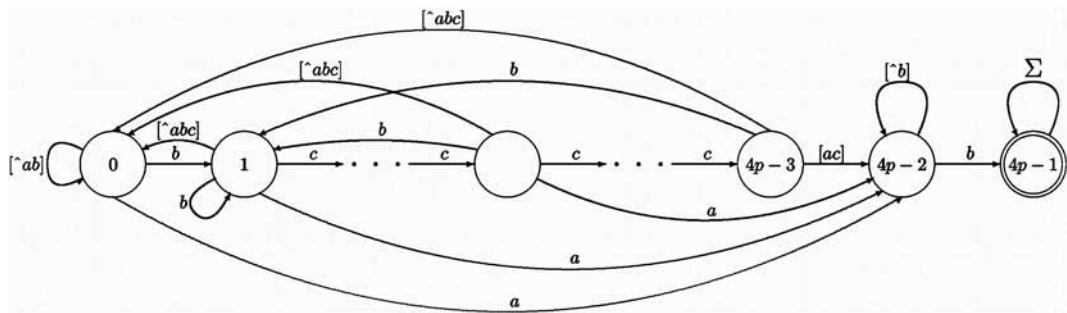
Докажем теперь, что $\forall C \in (0; 1] \cap \mathbb{Q}$ найдутся такие R_1, R_2, R_3 и R_4 , что:

$$\frac{|V(L(. * (R_1 | R_3). * (R_2 | R_4). *))|}{|V(L(. * R_1. * R_2. *) \cup L(. * R_3. * R_4. *))|} = C.$$

Пусть $C = \frac{p}{q}$, где $p, q \in \mathbb{N}$, $p \leq q$. В качестве выражений R_i возьмем следующие: $R_1 = a$, $R_2 = b$, $R_3 =$



a)



б)

Рис. 5. Автоматы V_1 (а) и V_2 (б)

$= bc\{4p - 3\}$, $R_4 = bc\{4(q - p)\}$. Если $p = q$, то $R_2 = R_4$, а значит

$$\begin{aligned} & \frac{|V(L(. * (R_1 | R_3). * (R_2 | R_4). *))|}{|V(L(. * R_1. * R_2. *) \cup L(. * R_3. * R_4. *))|} = \\ & = \frac{|V(L(. * R_1. * (R_2 | R_4). *) \cup L(. * R_3. * (R_2 | R_4). *))|}{|V(L(. * R_1. * R_2. *) \cup L(. * R_3. * R_4. *))|} = \\ & = \frac{|V(L(. * R_1. * R_2. *) \cup L(. * R_3. * R_4. *))|}{|V(L(. * R_1. * R_2. *) \cup L(. * R_3. * R_4. *))|} = 1. \end{aligned}$$

Пусть $p < q$, докажем, что тогда автоматы V_1 и V_2 , изображенные на рис. 5, являются приведенными конечными автоматами, принимающими языки $L(. * R_1. * R_2. * | . * R_3. * R_4. *)$ и $L(. * (R_1 | R_3). * (R_2 | R_4). *)$ соответственно.

Рассмотрим автомат V_1 . Очевидно, что любое слово из $L(. * a. * b. * | . * bc\{4p - 3\}. * bc\{4(q - p)\}. *)$ принимает-

ся автоматом V_1 . Докажем, что любое слово α , принимаемое V_1 , принадлежит этому языку. Пусть путь, соответствующий α , проходит через состояние $4q - 1$. По построению до попадания в $4q - 1$ он обязательно проходит по переходу с меткой a , а после попадания — по переходу с меткой b . Значит α имеет вид $. * a. * b. *$. Пусть путь, соответствующий α , не проходит через $4q - 1$. Тогда, так как в состояние $4q - 2$ ведут только два перехода — из состояния $4q - 3$ и $4q - 1$, в пути содержится переход по символу c из $4q - 3$ в $4q - 2$. В состоянии $4q - 3$ ведет только переход по символу c из $4q - 4$. Продолжая рассуждение до состояния $4p - 1$, получаем, что путь, соответствующий слову α , содержит путь из состояния $4p - 1$ в $4q - 2$, которому соответствует слово $c\{4(q - p)\}$. А так как в состоянии $4p - 1$ ведут только переходы с меткой b , то в пути, соответствующему слову α , найдется такой подпуть, не проходящий через состояния $0, \dots, 4p - 2$, но, возможно начинающийся в $4p - 2$, что соответствующим ему словом является $bc\{4(q - p)\}$. Заметим, что по построению любой путь из состояния 0 в состояние $4q - 2$, не проходящий

через состояние $4q - 1$, обязательно проходит через состояние $4p - 2$ и содержит переход из $4p - 3$ в $4p - 2$. Повторяя рассуждения о единственности входящих переходов, получаем, что в пути, соответствующем слову α , найдется подпуть из состояния 0 или 1 в состояние $4p - 2$, которому соответствует слово $bc\{4p - 3\}$. Причем по построению путь, соответствующий слову α , проходит сначала по второму подпути, а потом по первому. Таким образом, слово α , соответствующее данному пути, имеет вид $.bc\{4p - 3\}.bc\{4(q - p)\}.*$.

Докажем неотличимость состояний автомата V_1 . Пусть i и j — произвольные нефинальные состояния, причем без ограничения общности считаем, что $i < j$. Если $j = 4q - 1$, то различающим словом будет слово b . Если $j \leq 4p - 2$, то различающее слово — $c\{4p - 2 - j\} \times b\{4(q - p)\}$. Если $j > 4p - 2$, то $c\{4q - 2 - j\}$ различает состояния i и j .

Рассмотрим автомат V_2 . Очевидно, что любое слово из $L(.*(a|bc\{4p - 3\}).*(b|bc\{4(q - p)\}).*)$ принимается автоматом V_2 . Докажем, что любое слово α , принимаемое V_2 , принадлежит этому языку. Аналогично предыдущим рассуждениям, в пути, соответствующем слову α , должен присутствовать переход из состояния $4p - 2$ в состояние $4p - 1$ с меткой b . Также по построению путь должен содержать либо переход с меткой a из какого-то состояния в состояние $4p - 2$, либо переход из состояния $4p - 3$ с меткой c в состояние $4p - 2$. В первом случае получаем, что слово α имеет вид $.a.*b.*$, а во втором случае, в силу того, что в состояниях 2, ..., $4p - 3$ ведут лишь переходы с меткой c из состояний с номерами на единицу меньше, получаем, что путь, соответствующий слову α , содержит подпуть из состояния 1 в состояние $4p - 2$, соответствующий слову $c\{4p - 3\}$, а так как в состоянии 1 ведет лишь переход с меткой b , то слово α содержит подслово $bc\{4p - 3\}$ и имеет вид $.bc\{4p - 3\}.*b.*$.

Число состояний автоматов V_1 и V_2 равно $4q$ и $4p$ соответственно. Таким образом, при данном выборе выражений получаем

$$\frac{|V(L(.*(R_1 | R_3).*(R_2 | R_4).*))|}{|V(L(. * R_1 . * R_2 . *) \cup L(. * R_3 . * R_4 . *))|} = \frac{4p}{4q} = C.$$

Применение алгоритма

Для работы с регулярными выражениями и конечными автоматами был разработан программный комплекс [14], в котором также реализована возможность модификации выражений вида $.R_1.*R_2.*$.

Для проверки эффективности модификации регулярных выражений была выбрана база сигнатур сетевой системы обнаружения вторжений Snort [15]. Из нее были взяты 37 выражений вида $R_i = . * R'_i . * R''_j . *$. Затем для каждой пары выражений R_i и R_j ($i \neq j$) были построены автоматы $V(L(R_i | R_j))$ и $V(L(.*(R'_i | R'_j).*(R''_i | R''_j).*))$. Частота появления N определенных значений отноше-

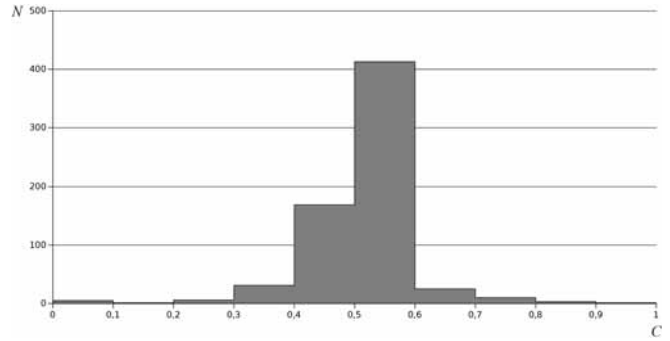


Рис. 6. Гистограмма эффективности алгоритма

ния $C = \frac{|V(L(.*(R'_i | R'_j).*(R''_i | R''_j).*))|}{|V(L(R_i | R_j))|}$, характеризующего эф-

фективность предложенного алгоритма, приведена на рис. 6.

Видно, что применение данного алгоритма изменения выражений для различных пар выражений в большинстве случаев привело к сокращению числа состояний до 40...60 %. При этом в одном из случаев число состояний сократилось примерно до 0,5 %, а в двух случаях, где у выражений совпали подвыражения R'_i или R''_j , число состояний не изменилось. Таким образом, модификация может быть эффективна, но это существенно зависит от выбора пары выражений для изменения.

Заключение

Предложен новый подход к изменению регулярных выражений для сокращения числа состояний детерминированных конечных автоматов, реализующих поиск по выражениям. Доказательно представлены неутраченные оценки на число состояний автомата при таком изменении в случае алфавита, состоящего из не менее чем пяти символов. Доказано также, что относительное уменьшение числа состояний может быть произвольным.

Автор выражает благодарность Галатенко Алексею Владимировичу и Панкратьеву Антону Евгеньевичу за постановку задачи и внимание к работе, Александрову Денису Евгеньевичу за помощь в редактировании статьи.

Список литературы

1. **Документация** системы Snort. URL: <http://www.snort.org/documents>
2. **Документация** системы Bro. URL: <http://www.bro.org/>
3. **Описание** системы L7-filter. URL: <http://l7-filter.sourceforge.net/README>
4. **Страница** аппаратных продуктов компании Cisco. URL: <http://www.cisco.com/c/en/us/products/security/intrusion-prevention-system-ips>
5. **Kumar S., Dharmapurikar S., Yu F., Crowley P., Turner J.** Algorithms to accelerate multiple regular expressions matching for

deep packet inspection // ACM SIGCOMM Computer Communication Review. 2006. Vol. 36, N. 4. P. 339–350.

6. **Aho A. V., Corasick M. J.** Efficient string matching: an aid to bibliographic search // Communications of the ACM. 1975. Vol. 18, N. 6. P. 333–340.

7. **Liu C., Wu J.** Fast Deep Packet Inspection with a Dual Finite Automata // Computers, IEEE Transactions on. 2013. Vol. 62, N. 2. P. 310–321.

8. **Kumar S., Chandrasekaran B., Turner J., Varghese G.** Curing regular expressions matching algorithms from insomnia, amnesia, and acalculia // Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems., 2007. P. 155–164.

9. **Smith R., Estan C., Jha S., Kong S.** Deflating the big bang: fast and scalable deep packet inspection with extended finite automata // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38, N. 4. P. 207–218.

10. **Yu F., Chen Z., Diao Y., Lakshman T. V., Katz R. H.** Fast and memory-efficient regular expression matching for deep packet inspection // Architecture for Networking and Communications systems, 2006. ANCS 2006. ACM/IEEE Symposium on. IEEE, 2006. P. 93–102.

11. **Кудрявцев В. Б., Алешин С. В., Подколзин А. С.** Введение в теорию автоматов. М.: Наука. Гл. ред. физ.-мат. лит., 1985.

12. **Martin J. C.** Introduction to Languages and the Theory of Computation. New York: McGraw-Hill, 2011.

13. **Smith R., Estan C., Jha S.** XFA: Faster signature matching with extended automata // Security and Privacy, 2008. IEEE Symposium on. IEEE, 2008. P. 187–201.

14. **Александров Д. Е.** Программный комплекс RE2FA. Свидетельство о государственной регистрации программы для ЭВМ № 2014614857.

15. **База сигнатур системы Snort.** URL: <http://www.snort.org/snort-rules/>

D. E. Alexandrov, Postgraduate Student, Lomonosov Moscow State University,
e-mail: dalexandrov@intsys.msu.ru

Reducing the Automata Complexity of Regular Languages by Means of Language Extension

This work addresses the problem of the "exponential explosion" of the number of states in a finite automata recognizing a set of regular languages specified by the union of regular expressions of the class $\cdot R_1 \cdot R_2 \cdot$, where R_1 and R_2 are arbitrary regular expressions. A method for reducing the number of states by extending the set to be recognized is proposed. Unimprovable estimates on the number of states of the automata under such modification in the case of an alphabet consisting of at least five symbols are presented. It is shown that the relative gain in the number of states may take arbitrary values. The efficiency of the method suggested is analyzed on regular expressions of Snort.

Keywords: finite automata, regular expressions, network intrusion detection systems

References

1. **Documentation** of Snort: <http://www.snort.org/documents>

2. **Documentation** of Bro: <http://www.bro.org/>

3. **Description** of L7 filter: <http://l7-filter.sourceforge.net/README>

4. **Cisco** products for intrusion prevention <http://www.cisco.com/c/en/us/products/security/intrusion-prevention-system-ips>

5. **Kumar S., Dharmapurikar S., Yu F., Crowley P., Turner J.** Algorithms to accelerate multiple regular expressions matching for deep packet inspection. *ACM SIGCOMM Computer Communication Review*. 2006. Vol. 36, N. 4. P. 339–350.

6. **Aho A. V., Corasick M. J.** Efficient string matching: an aid to bibliographic search. *Communications of the ACM*. 1975. Vol. 18, N. 6. P. 333–340.

7. **Liu C., Wu J.** Fast Deep Packet Inspection with a Dual Finite Automata. *Computers, IEEE Transactions on*. 2013. Vol. 62, N. 2. P. 310–321.

8. **Kumar S., Chandrasekaran B., Turner J., Varghese G.** Curing regular expressions matching algorithms from insomnia, amnesia, and

acalculia. *Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems*. ACM, 2007. P. 155–164.

9. **Smith R., Estan C., Jha S., Kong S.** Deflating the big bang: fast and scalable deep packet inspection with extended finite automata. *ACM SIGCOMM Computer Communication Review*. 2008. Vol. 38, N. 4. P. 207–218.

10. **Yu F., Chen Z., Diao Y., Lakshman T. V., Katz R. H.** Fast and memory-efficient regular expression matching for deep packet inspection. *Architecture for Networking and Communications systems. ACM/IEEE Symposium on*. IEEE, 2006. P. 93–102.

11. **Кудрявцев В. Б., Алыошин С. В., Подколзин А. С.** *Vvedenie v teoriiu avtomatov*. — М.: Nauka. Gl. red. fiz.-mat. lit., 1985.

12. **Martin J. C.** *Introduction to Languages and the Theory of Computation*. New York: McGraw-Hill, 2011.

13. **Smith R., Estan C., Jha S.** XFA: Faster signature matching with extended automata. *Security and Privacy. IEEE Symposium on*. IEEE, 2008. P. 187–201.

14. **Alexandrov D. E.** Programmnyy kompleks RE2FA. Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM N. 2014614857.

15. **Snort** rules. URL: <http://www.snort.org/snort-rules/>

И. О. Жаринов, д-р техн. наук, доц., зав. каф., Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (НИУ ИТМО), руководитель учебно-научного центра, Санкт-Петербургское ОКБ "Электроавтоматика" им. П. А. Ефимова, e-mail: igor_rabota@pisem.net,

О. О. Жаринов, канд. техн. наук, доц., Санкт-Петербургский государственный университет аэрокосмического приборостроения (ГУАП)

Исследование свойства равноконтрастности цветовых пространств, применяемых в авионике

Рассмотрена задача исследования свойства равноконтрастности цветовых пространств, применяемых в авионике. Свойство равноконтрастности цветовых пространств является определяющим для вычисления оценки, характеризующей меру различия цветов — цветовой разницы на XU - и на UV -плоскостях. Приведены гистограммы распределения координат цветности на обеих плоскостях, полученные с использованием методов математического моделирования. Показано, что распределение координат цветности и на XU -, и на UV -плоскостях отлично от равномерного, в связи с чем оба цветовых пространства в общем случае не могут классифицироваться как равноконтрастные и оценка числовой меры различия цветов не будет равнозначной во всех областях цветового графика XU - и UV -плоскости. Определены области неравноконтрастности цветовых графиков на обеих плоскостях.

Ключевые слова: цветовые пространства, системы индикации, авионика, свойство равноконтрастности

Введение

При решении ряда практических задач, связанных с анализом и обработкой значений координат цветности элементов изображения, индицируемых на жидкокристаллическом (ЖК) экране бортовых средств отображения пилотажно-навигационных параметров и геоинформационных данных [1–4], необходимо вычислять значение, характеризующее меру различия цветов и оттенков. К таким задачам, в частности, относится задача подтверждения цветовой палитры, используемой в программном обеспечении авионики, при сертификации систем индикации.

Мера различия цветов и оттенков представляет собой параметр, характеризующий расстояние между цветами и оттенками в выбранном цветовом пространстве. Согласно ГОСТ Р 52324—2005 «Эргономические требования к работе с визуальными дисплеями, основанными на плоских панелях» вычисление расстояния между цветами необходимо проводить в цветовых пространствах, представленных на плоскости в виде равноконтрастных цветовых графиков [5]. Равноконтрастный цветовой график — это двумерный график, система координат которого выбрана таким образом, чтобы равным расстояниям в любой части графика соответствовало равное число цветовых различий. Согласно ГОСТ Р 52324—2005 такой системой координат является прямоугольная UV -система с координатами (u', v') .

Между тем известно, что разработчики авиационной техники при создании функционального программного обеспечения оперируют с кодами RGB (R — Red, G — Green, B — Blue), а средства измерения, позволяющие экспериментально оценить координаты цветности индицируемых элементов изображения, представляют оценки в XU -системе с координатами (x, y) : красный цвет $0,25 < y < 0,33, x > 0,70$; зеленый цвет $x > 0,08, y > 0,70$; синий цвет $x < 0,15, y < 0,10$. При этом двумерный график XU -системы не является равноконтрастным, в связи с чем он не может использоваться для вычисления меры различия цветов.

Таким образом, программный код бортового средства отображения информации формируется в кодах RGB . Измерительное устройство (колориметр), используемое при этом для получения значений координат цветности, дает оценки на XU -плоскости, а вычисление межцветового различия, необходимого для подтверждения цветовой палитры, осуществляется на UV -плоскости.

Положения ГОСТ Р 52870—2007 «Средства отображения информации коллективного пользования» свидетельствуют, что любое средство отображения информации, в том числе и ЖК экран, способно воспроизводить не все доступные для восприятия человеческим глазом и существующие теоретически в UV -системе координат цвета и оттенки, а только часть из них. Эта часть ограничена на плоскости равноконтрас-

тного цветового графика областью, границы которой аппроксимируются треугольником (цветовой охват). Вершины треугольника соответствуют координатам цветности для красного, зеленого и синего цветов.

Опытным путем замечено, что распределение точек (u', v') -координат цветности равноконтрастного цветового графика в пределах геометрического места точек, вписанного в треугольник цветового охвата, неравномерно, т. е. свойство равноконтрастности выполняется не в каждой области графика UV -системы. Таким образом, способ вычисления меры цветового различия, предлагаемый ГОСТ Р 52324—2005, не может использоваться равномерно во всех областях UV -системы координат, а может применяться лишь в отдельных ее зонах.

Целью исследования, результаты которого представлены в статье, является поиск областей, вписанных в треугольник цветового охвата, в пределах которых имеет место существенная неравномерность распределения (u', v') -координат. Такие области могут быть подвергнуты нелинейному преобразованию, линеаризирующему числовую меру цветовых различий неравноконтрастных графиков, например, по методикам, изложенным в работах [6—8].

1. Цветовые пространства, применяемые в авионике

В современной авиационной промышленности при разработке средств бортовой индикации используют следующие четыре цветовых пространства: XYZ , XY , UV , RGB . Пространство XYZ основано на представлении цвета в виде трех координат цвета X , Y , Z , определенных по системе цветового треугольника Максвелла. Пространство XY предполагает способ задания каждого цвета или оттенка цвета в виде пары вещественных (x, y) -координат на XY -плоскости. Пространство UV предполагает способ задания каждого цвета или оттенка цвета в виде пары вещественных (u', v') -координат на UV -плоскости. В пространстве RGB , используемом программистами при разработке функционального программного обеспечения авионики, способ задания цвета или оттенка цвета основан на двоичном (десятичном) представлении кодов основных цветов (красный, зеленый, синий).

Взаимосвязь цветовых пространств подробно рассмотрена авторами в работе [9] и основана на правилах прямого $RGB \rightarrow XYZ$ и обратного $XYZ \rightarrow RGB$ преобразований Грассмана:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (1)$$

где X , Y , Z — компоненты цвета в системе XYZ цветового треугольника Максвелла; X_r , X_g , X_b , Y_r , Y_g , Y_b , Z_r , Z_g , Z_b — компоненты цвета, определенные Международной комиссией по освещению и используемые в качестве эталона для точного стандарта определения

цвета; R , G , B — десятичный код цвета компонентов основных цветов в системе RGB . Компоненты X_r , Y_r , Z_r определяют правило преобразования кода RGB для эталонного значения красного цвета, компоненты X_g , Y_g , Z_g и X_b , Y_b , Z_b — для зеленого цвета и синего цвета соответственно.

Переход от координат цвета цветового пространства XYZ к координатам цветности на XY - и на UV -плоскостях осуществляется по следующим формулам:

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad (2)$$

$$u' = \frac{4X}{X+15Y+3Z}, \quad v' = \frac{9Y}{X+15Y+3Z}. \quad (3)$$

При этом ГОСТ Р 52324—2005 вводит следующую взаимосвязь преобразований:

$$u' = \frac{2x}{6y-x+1,5}, \quad v' = \frac{4,5y}{6y-x+1,5},$$

$$x = \frac{4,5u'}{3u'-8v'+6}, \quad y = \frac{2v'}{3u'-8v'+6}.$$

2. Оценка разрешающей способности преобразования Грассмана на XY -плоскости и на UV -плоскости

Разрешающая способность преобразования Грассмана представляет собой минимальное (по модулю) ненулевое значение приращения (x, y) - или (u', v') -координат, которое возникает при дискретном изменении кода RGB . Оценки разрешающей способности при этом имеют следующий вид:

$$\Delta x = \frac{\det \begin{pmatrix} \det \begin{pmatrix} X_b & Y_b + Z_b \\ X_g & Y_g + Z_g \end{pmatrix} & R_i & R_{i+1} \\ \det \begin{pmatrix} X_r & Y_r + Z_r \\ X_b & Y_b + Z_b \end{pmatrix} & G_i & G_{i+1} \\ \det \begin{pmatrix} X_g & Y_g + Z_g \\ X_r & Y_r + Z_r \end{pmatrix} & B_i & B_{i+1} \end{pmatrix}}{(a_r R_i + a_g G_i + a_b B_i)(a_r R_{i+1} + a_g G_{i+1} + a_b B_{i+1})}, \quad (4)$$

$$\Delta y = \frac{\det \begin{pmatrix} \det \begin{pmatrix} Y_b & X_b + Z_b \\ Y_g & X_g + Z_g \end{pmatrix} & R_i & R_{i+1} \\ \det \begin{pmatrix} Y_r & X_r + Z_r \\ Y_b & X_b + Z_b \end{pmatrix} & G_i & G_{i+1} \\ \det \begin{pmatrix} Y_g & X_g + Z_g \\ Y_r & X_r + Z_r \end{pmatrix} & B_i & B_{i+1} \end{pmatrix}}{(a_r R_i + a_g G_i + a_b B_i)(a_r R_{i+1} + a_g G_{i+1} + a_b B_{i+1})}$$

$$\Delta u' = \frac{\det \begin{pmatrix} \det \begin{pmatrix} X_b & 60Y_b + 12Z_b \\ X_g & 60Y_g + 12Z_g \end{pmatrix} & R_i & R_{i+1} \\ \det \begin{pmatrix} X_r & 60Y_r + 12Z_r \\ X_b & 60Y_b + 12Z_b \end{pmatrix} & G_i & G_{i+1} \\ \det \begin{pmatrix} X_g & 60Y_g + 12Z_g \\ X_r & 60Y_r + 12Z_r \end{pmatrix} & B_i & B_{i+1} \end{pmatrix}}{(a'_r R_i + a'_g G_i + a'_b B_i)(a'_r R_{i+1} + a'_g G_{i+1} + a'_b B_{i+1})}, \quad (5)$$

$$\Delta v' = \frac{\det \begin{pmatrix} \det \begin{pmatrix} Y_b & 9X_b + 27Z_b \\ Y_g & 9X_g + 27Z_g \end{pmatrix} & R_i & R_{i+1} \\ \det \begin{pmatrix} Y_r & 9X_r + 27Z_r \\ Y_b & 9X_b + 27Z_b \end{pmatrix} & G_i & G_{i+1} \\ \det \begin{pmatrix} Y_g & 9X_g + 27Z_g \\ Y_r & 9X_r + 27Z_r \end{pmatrix} & B_i & B_{i+1} \end{pmatrix}}{(a'_r R_i + a'_g G_i + a'_b B_i)(a'_r R_{i+1} + a'_g G_{i+1} + a'_b B_{i+1})}$$

где Δx — разрешающая способность по x -координате; Δy — разрешающая способность по y -координате; $\Delta u'$ — разрешающая способность по u' -координате; $\Delta v'$ — разрешающая способность по v' -координате; $R_i, G_i, B_i, R_{i+1}, G_{i+1}, B_{i+1}$ — десятичные коды для начальной i -й и конечной $(i+1)$ -й точки в системе RGB , вызвавшие приращения $\Delta x, \Delta y$ в системе XY и $\Delta u', \Delta v'$ в системе UV ; $a_r = X_r + Y_r + Z_r, a_g = X_g + Y_g + Z_g, a_b = X_b + Y_b + Z_b$; $a'_r = X_r + 15Y_r + 3Z_r, a'_g = X_g + 15Y_g + 3Z_g, a'_b = X_b + 15Y_b + 3Z_b$.

Оценки разрешающей способности преобразования Грассмана получены теоретически путем вычисления приращений:

$$\Delta x = x_{i+1}|_{R_{i+1}G_{i+1}B_{i+1}} - x_i|_{RG B_i}, \quad \Delta y = y_{i+1}|_{R_{i+1}G_{i+1}B_{i+1}} - y_i|_{RG B_i},$$

$$\Delta u' = u'_{i+1}|_{R_{i+1}G_{i+1}B_{i+1}} - u'_i|_{RG B_i}, \quad \Delta v' = v'_{i+1}|_{R_{i+1}G_{i+1}B_{i+1}} - v'_i|_{RG B_i}$$

для $R_{i+1} = R_i + \Delta R, G_{i+1} = G_i + \Delta G, B_{i+1} = B_i + \Delta B$ с использованием выражений (1)–(3). Методика получения оценок следующая (на примере XY -системы).

В соответствии с формулами (1), (2):

$$\begin{cases} \Delta x = x_{i+1}|_{R_{i+1}G_{i+1}B_{i+1}} - x_i|_{RG B_i} = \\ = \frac{X_{i+1}}{X_{i+1} + Y_{i+1} + Z_{i+1}} - \frac{X_i}{X_i + Y_i + Z_i} \Rightarrow \\ \Delta y = y_{i+1}|_{R_{i+1}G_{i+1}B_{i+1}} - y_i|_{RG B_i} = \\ = \frac{Y_{i+1}}{X_{i+1} + Y_{i+1} + Z_{i+1}} - \frac{Y_i}{X_i + Y_i + Z_i} \\ \Rightarrow \begin{cases} \Delta x = \frac{X_{i+1}Y_i + X_{i+1}Z_i - X_iY_{i+1} - X_iZ_{i+1}}{(X_i + Y_i + Z_i)(X_{i+1} + Y_{i+1} + Z_{i+1})}, \\ \Delta y = \frac{Y_{i+1}X_i + Y_{i+1}Z_i - Y_iX_{i+1} - Y_iZ_{i+1}}{(X_i + Y_i + Z_i)(X_{i+1} + Y_{i+1} + Z_{i+1})}, \end{cases} \end{cases} \quad (6)$$

где

$$\begin{aligned} X_i &= X_r R_i + X_g G_i + X_b B_i, \\ Y_i &= Y_r R_i + Y_g G_i + Y_b B_i, \\ Z_i &= Z_r R_i + Z_g G_i + Z_b B_i, \\ X_{i+1} &= X_r R_{i+1} + X_g G_{i+1} + X_b B_{i+1}, \\ Y_{i+1} &= Y_r R_{i+1} + Y_g G_{i+1} + Y_b B_{i+1}, \\ Z_{i+1} &= Z_r R_{i+1} + Z_g G_{i+1} + Z_b B_{i+1}. \end{aligned}$$

После замены переменных и раскрытия скобок числители первого и второго уравнений системы (6) примут следующий вид:

$$\begin{aligned} X_{i+1}Y_i + X_iZ_i - X_iY_{i+1} - X_iZ_{i+1} &= \\ &= (R_{i+1}G_i - R_iG_{i+1})(X_rY_g + X_rZ_g - X_gY_r - X_gZ_r) + \\ &+ (R_{i+1}B_i - R_iB_{i+1})(X_rY_b + X_rZ_b - X_bY_r - X_bZ_r) + \\ &+ (G_{i+1}B_i - G_iB_{i+1})(X_gY_b + X_gZ_b - X_bY_g - X_bZ_g) = \\ &= \det \begin{pmatrix} \det \begin{pmatrix} X_b & Y_b + Z_b \\ X_g & Y_g + Z_g \end{pmatrix} & R_i & R_{i+1} \\ \det \begin{pmatrix} X_r & Y_r + Z_r \\ X_b & Y_b + Z_b \end{pmatrix} & G_i & G_{i+1} \\ \det \begin{pmatrix} X_g & Y_g + Z_g \\ X_r & Y_r + Z_r \end{pmatrix} & B_i & B_{i+1} \end{pmatrix}, \end{aligned} \quad (7)$$

$$\begin{aligned} Y_{i+1}X_i + Y_iZ_i - Y_iX_{i+1} - Y_iZ_{i+1} &= \\ &= (R_{i+1}G_i - R_iG_{i+1})(Y_rX_g + Y_rZ_g - Y_gX_r - Y_gZ_r) + \\ &+ (R_{i+1}B_i - R_iB_{i+1})(Y_rX_b + Y_rZ_b - Y_bX_r - Y_bZ_r) + \\ &+ (G_{i+1}B_i - G_iB_{i+1})(Y_gX_b + Y_gZ_b - Y_bX_g - Y_bZ_g) = \\ &= \det \begin{pmatrix} \det \begin{pmatrix} Y_b & X_b + Z_b \\ Y_g & X_g + Z_g \end{pmatrix} & R_i & R_{i+1} \\ \det \begin{pmatrix} Y_r & X_r + Z_r \\ Y_b & X_b + Z_b \end{pmatrix} & G_i & G_{i+1} \\ \det \begin{pmatrix} Y_g & X_g + Z_g \\ Y_r & X_r + Z_r \end{pmatrix} & B_i & B_{i+1} \end{pmatrix}. \end{aligned} \quad (8)$$

На заключительном шаге преобразования выражений (7), (8) применено правило разложения определителя по первому столбцу и по минорам [10]. Знаменатель уравнений системы (6) равен

$$\begin{aligned} (X_i + Y_i + Z_i)(X_{i+1} + Y_{i+1} + Z_{i+1}) &= \\ &= (R_i(X_r + Y_r + Z_r) + G_i(X_g + Y_g + Z_g) + \\ &+ B_i(X_b + Y_b + Z_b))(R_{i+1}(X_r + Y_r + Z_r) + \\ &+ G_{i+1}(X_g + Y_g + Z_g) + B_{i+1}(X_b + Y_b + Z_b)) = \\ &= (a_r R_i + a_g G_i + a_b B_i)(a_r R_{i+1} + a_g G_{i+1} + a_b B_{i+1}), \end{aligned}$$

где $a_r = X_r + Y_r + Z_r, a_g = X_g + Y_g + Z_g, a_b = X_b + Y_b + Z_b$.

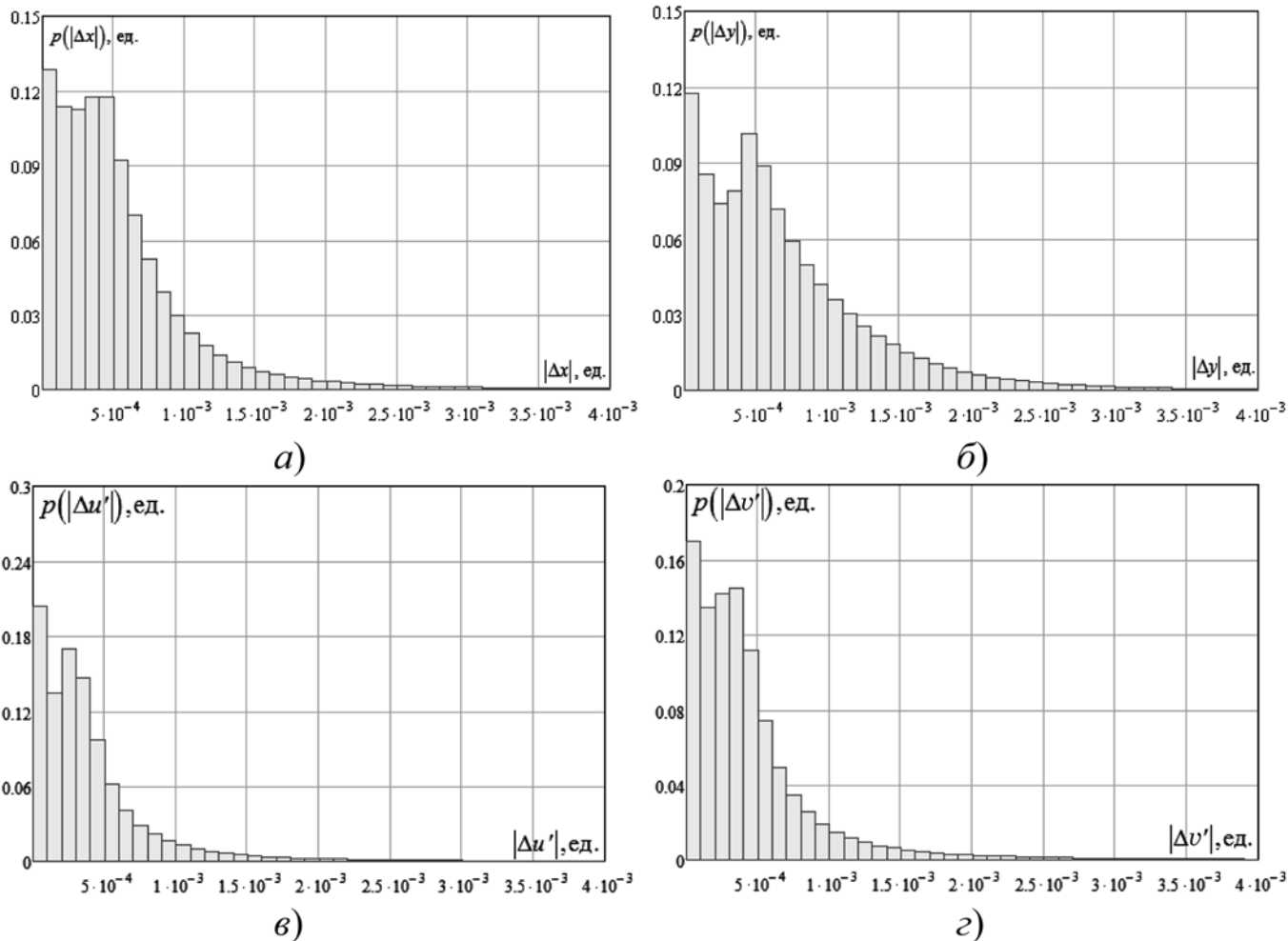


Рис. 1. Гистограммы распределения оценок:
 а – $|\Delta x|$; б – $|\Delta y|$; в – $|\Delta u'|$; г – $|\Delta v'|$

Таким образом, приращения координат $\Delta x = x_{i+1} - x_i$, $\Delta y = y_{i+1} - y_i$ на XY -плоскости, соответствующие дискретному изменению десятичного кода $RGB \in [0, 255]$ от значения $R_i G_i B_i$ до значения $R_{i+1} G_{i+1} B_{i+1}$, вычисляются по формуле (4). Аналогично может быть получена система (5) для UV -плоскости.

Гистограммы $p(|\Delta x|)$, $p(|\Delta y|)$, $p(|\Delta u'|)$, $p(|\Delta v'|)$ распределения оценок $|\Delta x|$, $|\Delta y|$, $|\Delta u'|$, $|\Delta v'|$ для $\Delta = \Delta R = \Delta G = \Delta B = 1$ приведены на рис. 1.

Как видно на рис. 1, в обеих системах координат распределение оценок приращений $|\Delta x|$, $|\Delta y|$, $|\Delta u'|$, $|\Delta v'|$, возникающих при изменении десятичного кода $RGB \in [0, 255]$ от значения $R_i G_i B_i$ до значения $R_{i+1} G_{i+1} B_{i+1}$ с единичным шагом по каждому из основных цветов, не является равномерным, т. е. в пределах треугольника цветового охвата как на XY -плоскости, так и на UV -плоскости существуют дискретные переходы, различные по абсолютной величине.

3. Программная реализация модели вычисления гистограммы распределения координат цветности цветовых пространств

Для исследования свойства равноконтрастности цветовых пространств на XY - и на UV -плоскостях была разработана специализированная компьютерная программа в среде MathCAD 15.0. Код программы приведен на рис. 2.

Исходными данными для программы являются значения матрицы преобразования (1), характеризующие положение треугольника цветового охвата в цветовом пространстве. Значения координат цветности на XY - и на UV -плоскостях вычисляются в соответствии с формулами (2), (3). Переменной h определен шаг вычисления гистограммы по каждой из двух осей.

Элементами двумерных массивов Izo_{xy} , Izo_{uv} являются значения частот попадания координат цвет-

$X_r := 0.478$ $X_g := 0.299$ $X_b := 0.175$
 $Y_r := 0.263$ $Y_g := 0.650$ $Y_b := 0.081$
 $Z_r := 0.020$ $Z_g := 0.160$ $Z_b := 0.908$

$$XYZ(R,G,B) := \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$x(R,G,B) := \frac{XYZ(R,G,B)_0}{XYZ(R,G,B)_0 + XYZ(R,G,B)_1 + XYZ(R,G,B)_2}$$

$$u(R,G,B) := \frac{4 \cdot XYZ(R,G,B)_0}{XYZ(R,G,B)_0 + 15 \cdot XYZ(R,G,B)_1 + 3 \cdot XYZ(R,G,B)_2}$$

$$y(R,G,B) := \frac{XYZ(R,G,B)_1}{XYZ(R,G,B)_0 + XYZ(R,G,B)_1 + XYZ(R,G,B)_2}$$

$$v(R,G,B) := \frac{9 \cdot XYZ(R,G,B)_1}{XYZ(R,G,B)_0 + 15 \cdot XYZ(R,G,B)_1 + 3 \cdot XYZ(R,G,B)_2}$$

$h := 0.001$

```

Izo_xy :=
K ← 1·floor(1/h)
for k1 ∈ 0,1..K-1
  for k2 ∈ 0,1..K-1
    M_k1,k2 ← floor(0)
  for Ri ∈ 1,2..254
    trace("Ri={0}", Ri)
    for Gi ∈ 0,1..254
      for Bi ∈ 0,1..254
        xi ← x(Ri,Gi,Bi)
        yi ← y(Ri,Gi,Bi)
        kx ← floor(K·xi)
        ky ← floor[K·(1-yi)]
        M_ky,kx ← M_ky,kx + 1
      1
    M

```

```

Izo_uv :=
K ← 1·floor(1/h)
for k1 ∈ 0,1..K-1
  for k2 ∈ 0,1..K-1
    M_k1,k2 ← floor(0)
  for Ri ∈ 1,2..254
    trace("Ri={0}", Ri)
    for Gi ∈ 0,1..254
      for Bi ∈ 0,1..254
        xi ← u(Ri,Gi,Bi)
        yi ← v(Ri,Gi,Bi)
        kx ← floor(K·xi)
        ky ← floor[K·(1-yi)]
        M_ky,kx ← M_ky,kx + 1
      1
    M

```

```

Setka :=
K ← 1·floor(1/h)
KΔ ← K/10
for k1 ∈ 0,1..K-1
  for k2 ∈ 0,1..K-1
    Setka_k1,k2 ← if((floor(k1/KΔ) = k1/KΔ) + (floor(k2/KΔ) = k2/KΔ), 30, 0)
  Setka

```

$maxM_{xy} := max(Izo_{xy}) + max(Setka)$

$maxM_{xy} := max(Izo_{xy}) + max(Setka)$

$Izobmp_{xy} := 255 - (Izo_{xy} + Setka) \cdot \frac{255}{maxM_{xy}}$

$Izobmp_{xy} := 255 - (Izo_{xy} + Setka) \cdot \frac{255}{maxM_{xy}}$

$WRITEBMP("xy.bmp") := Izobmp_{xy}$

$WRITEBMP("xy.bmp") := Izobmp_{xy}$

Рис. 2. Код программы расчета гистограммы распределения координат цветности цветковых пространств XY и UV в среде MathCAD 15.0

ности (соответственно x , y и u , v) в подинтервал длиной h . Структура циклов расчета элементов массивов Izo_{xy} , Izo_{uv} идентична. Заполнение массивов Izo_{xy} , Izo_{uv} проводится в цикле при последовательном изменении десятичного кода RGB в диапазоне значений $0...255$ по каждому из компонентов основных цветов. Для определенности начальное значение по компоненте R принято равным единице.

Для удобства отображения гистограммы распределения координат цветности цветowych графиков в программе выполняется расчет координатной сетки. Сетка вычисляется в цикле *Setka*. Сетка совмещается с графиком гистограммы. Частоты попадания координат цветности в подинтервалы нормируются. Пользователю отображается гистограмма распределения координат цветности в обеих плоскостях — XY и UV . Графики гистограмм сохраняются в виде файлов формата .bmp.

Моделирование проводилось на персональном компьютере ASUS K56CB-X0391H со следующими характеристиками: процессор Intel(R) Core(TM) i5-3337U, 4 ядра, тактовая частота 1,8 ГГц, оперативная память 6 Гбайт под управлением операционной системы Windows 8.1. Время вычисления гистограмм распределения координат цветности каждого цветового пространства (XY или UV) составило 2,5 мин.

4. Оценка свойства равноконтрастности цветowych графиков

На рис. 3 приведены распределения координат цветности, сосредоточенных в области треугольников цветового охвата UV - и XY -плоскостей. Распределения получены путем полного перебора кодов RGB с единичным шагом в 8-битной модели представления основных цветов и вычисления координат цветности в соответствии с формулами (1)–(3). Таким образом, в пределах треугольника цветового охвата обеих плоскостей сосредоточены по 2^{24} координат цветности.

Распределения представлены методом теневой пластики в оттенках серого цвета, где наиболее темным областям на графике соответствует повышенная плотность сосредоточения точек цветowych координат, а светлым областям — пониженная плотность сосредоточения точек.

Количественная мера сосредоточения точек при построении рис. 3 вычислялась как число различных координат цветности, приходящееся на единицу площади графика. Единица площади графика при построении рис. 3 для определенности была принята в размерах $0,001 \times 0,001$ ед.²

Анализ распределения координат цветности показывает, что в пределах площади треугольника цветового охвата существуют неоднородные области заливки серого цвета, т. е. число координат цветности, приходящееся на единицу площади графика, в разных областях графика различно.

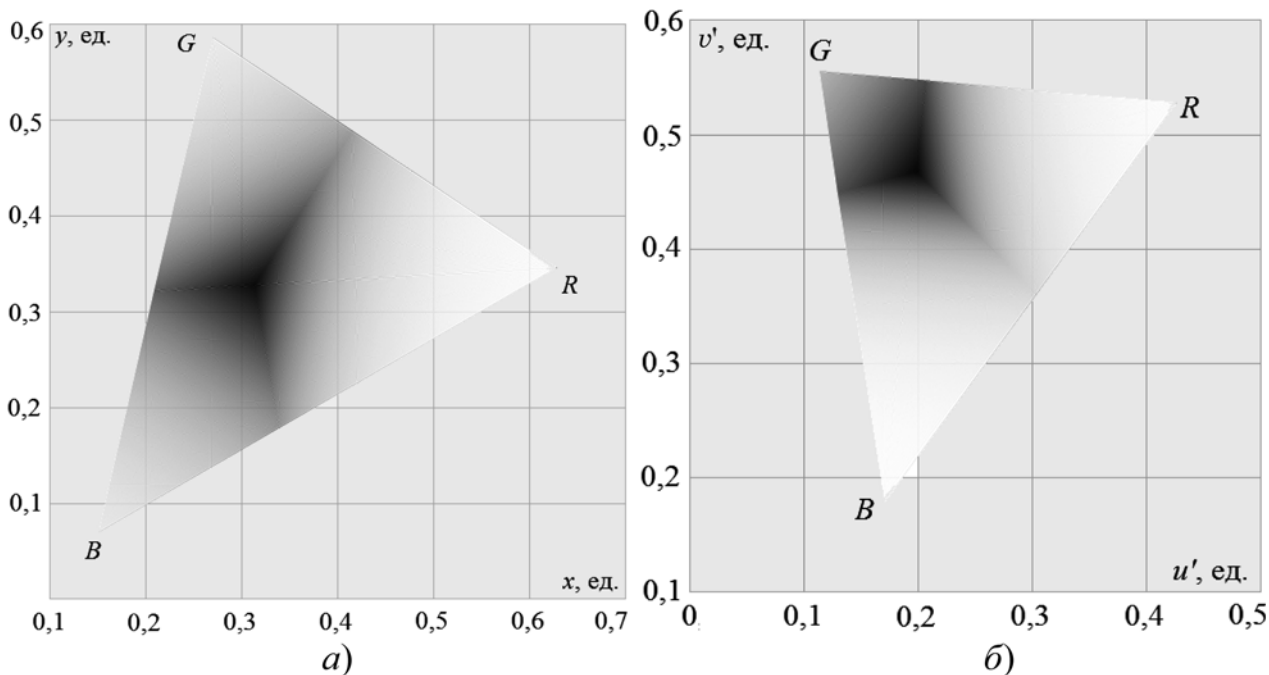


Рис. 3. Распределение координат цветности: a — на XY -плоскости; b — на UV -плоскости

Максимум плотности распределения (x, y) -координат цветности на XU -плоскости (рис. 3, *а*) приходится на зоны белого цвета (в окрестностях точки белого: $x \approx 0,333$, $y \approx 0,333$). Существенная неравномерность распределения наблюдается также вдоль линий, соединяющих координаты цветности точки белого и координаты цветности желтого ($x \approx 0,42$, $y \approx 0,48$), малинового ($x \approx 0,34$, $y \approx 0,17$) и голубого ($x \approx 0,21$, $y \approx 0,33$) цветов. В окрестностях основных цветов (красного, зеленого, синего) — вершины треугольника цветового охвата — распределение координат цветности можно считать равномерным (в сравнении с окрестностью точки белого). Координаты вершин треугольника цветового охвата на XU -плоскости определяются из следующих соотношений:

$$R: \{x_R = X_r/(X_r + Y_r + Z_r), \\ y_R = Y_r/(X_r + Y_r + Z_r)\},$$

$$G: \{x_G = X_g/(X_g + Y_g + Z_g), \\ y_G = Y_g/(X_g + Y_g + Z_g)\},$$

$$B: \{x_B = X_b/(X_b + Y_b + Z_b), \\ y_B = Y_b/(X_b + Y_b + Z_b)\}.$$

Максимум плотности распределения (u', v') -координат цветности на UV -плоскости (рис. 3, *б*) приходится также на зоны белого цвета (в окрестностях точки белого: $u' \approx 0,19$, $v' \approx 0,48$). Неравномерность распределения наблюдается вдоль линий, соединяющих координаты цветности точки белого и координаты цветности желтого ($u' \approx 0,21$, $v' \approx 0,54$), малинового ($u' \approx 0,31$, $v' \approx 0,36$) и голубого ($u' \approx 0,13$, $v' \approx 0,44$) цветов. В окрестностях основных цветов (красного, зеленого, синего) распределение координат цветности равномерно. Координаты вершин треугольника цветового охвата на UV -плоскости определяются из следующих соотношений:

$$R: \{u'_R = 4X_r/(X_r + 15Y_r + 3Z_r), \\ v'_R = 9Y_r/(X_r + 15Y_r + 3Z_r)\},$$

$$G: \{u'_G = 4X_g/(X_g + 15Y_g + 3Z_g), \\ v'_G = 9Y_g/(X_g + 15Y_g + 3Z_g)\},$$

$$B: \{u'_B = 4X_b/(X_b + 15Y_b + 3Z_b), \\ v'_B = 9Y_b/(X_b + 15Y_b + 3Z_b)\}.$$

Следует заметить, что равномерность распределения координат цветности на рис. 3, *б* существенно выше, чем на рис. 3, *а*. В связи с этим UV -плоскость, в

отличие от XU -плоскости, может считаться равноконтрастной плоскостью. Однако свойство равноконтрастности цветового графика нарушается в окрестностях точки белого и вдоль следующих линий цветов: белый—малиновый, белый—желтый, белый—голубой.

На рис. 4 и 5 (см. вторую сторону обложки) приведены графики сечений распределений координат цветности, показанных на рис. 3. На рис. 4, *а* приведено сечение распределения XU -плоскости по y -координате.

По оси ординат отложено число N координат цветности, приходящееся на единицу длины графика, равную 0,001 ед. Сечения получены на следующих уровнях: $y = 0,25$; $y = 0,3$; $y = 0,4$; $y = 0,45$; $y = 0,5$. Анализ данных рис. 4, *а* показывает, что максимум распределения координат цветности приходится на точку белого цвета, в которой на каждые 0,001 ед. приращений $|\Delta x|$ приходится порядка 550 точек. Для всех пяти представленных сечений распределение координат отлично от равномерного, что исключает свойство равноконтрастности для графика XU -плоскости по определению. Рис. 4, *б*, на котором показано сечение распределения XU -плоскости по x -координате, также подтверждает неравномерность распределения координат цветности на уровнях сечений $x = 0,22$; $x = 0,25$; $x = 0,3$; $x = 0,35$; $x = 0,4$.

Анализ данных рис. 5, *а* (см. вторую сторону обложки), на котором приведено сечение UV -плоскости по v' -координате, показывает, что равномерность распределения координат цветности сохраняется в пределах значений $0,25 \leq v' < 0,4$. При увеличении v' до уровня 0,5, равномерность распределения нарушается. Аналогично, на рис. 5, *б* равномерность распределения координат цветности наблюдается при $0,3 \leq u' < 0,4$. Для $u' < 0,3$ равномерность распределения нарушается, разница в плотности распределения точек достигает 10 раз. Таким образом, графики цветовых пространств UV -плоскости не могут считаться строго равноконтрастными в каждой точке. В связи с этим вычисление межцветового пространства на их основе не является равнозначным.

Заключение

Результаты моделирования показывают, что для выбранного профиля ЖК экрана, заданного компонентами $X_r = 0,478$; $X_g = 0,299$; $X_b = 0,175$; $Y_r = 0,263$; $Y_g = 0,650$; $Y_b = 0,081$; $Z_r = 0,020$; $Z_g = 0,160$; $Z_b = 0,908$ матрицы преобразования (1), в окрестностях точки белого цвета наблюдается существенная нелинейность графика цветового UV -пространства. В меньшей степени нелинейность выражена вдоль линий цветовых переходов: белый—малиновый, белый—желтый, белый—голубой.

Эксперименты с профилями ЖК экранов, заданных матрицами преобразования (1) с различными системами оценки баланса белого цвета (D-75, D-65, D-55, D-50 и др.), в пределах каждой из которых компоненты основных цветов $X_r, X_g, X_b, Y_r, Y_g, Y_b, Z_r, Z_g, Z_b$ базового преобразования (1) являются различными, показали, что направления неравномерности цветовых графиков сохраняются. Смещаются только границы и вершины треугольника цветового охвата в соответствии с изменениями значений компонентов $X_r, X_g, X_b, Y_r, Y_g, Y_b, Z_r, Z_g, Z_b$.

В связи с этим при вычислении цветовой разницы по ГОСТ Р 52324—2005 на цветовом графике UV-плоскости для цветов и оттенков, (u', v') -координаты цветности которых расположены в непосредственной близости от точки белого цвета (в пределах окружности радиусом 0,3 ед.) или вдоль линий белый—малиновый, белый—желтый, белый—голубой, необходимо вводить специальные преобразования, линеаризирующие отношение числа координат цветности, приходящееся на единицу изменения координаты. Примеры таких преобразований, выполненных пока еще только на XY-плоскости, изложены в работах [5, 7, 8].

Список литературы

1. Парамонов П. П., Костишин М. О., Жаринов И. О., Нечаев В. А., Сударчиков С. А. Принцип формирования и отобра-

жения массива геоинформационных данных на экран средств бортовой индикации // Научно-технический вестник информационных технологий, механики и оптики. 2013. № 6. С. 136—142.

2. Жаринов И. О., Жаринов О. О. Бортовые средства отображения информации на плоских жидкокристаллических панелях: учеб. пособие. Информационно-управляющие системы. СПб.: ГУАП, 2005. 144 с.

3. Костишин М. О., Жаринов И. О., Жаринов О. О. Исследование визуальных характеристик средств отображения пилотажно-навигационных параметров и геоинформационных данных в авионике // Информационно-управляющие системы. 2014. № 4. С. 61—67.

4. Barber S. et al. US Patent 7,417,641 B1: Aeronautical chart display apparatus and method; Aug. 26. 2008.

5. Крыжевич Л. С. Выбор и преобразование цветового пространства // Ученые записки: электронный научный журнал Курского государственного университета. 2011. № 4. С. 22—28.

6. Ложкин Л. Д. Дифференциальная колориметрия в телевидении: Автореф. дисс. д-ра техн. наук: СПб., 2014. 31 с.

7. Ложкин Л. Д., Неганов В. А. Способ преобразования цветового пространства. Патент 2494461 С2. МПК G06K 9/68; заявл. 08.07.2011, № 011128504/08; опубл. 27.09.2013, Бюл. № 27.

8. Со И. А., Малыхина Г. Ф. Способ и устройство для измерения цветовых различий (варианты). Патент 2477843 С2. МПК G 01J 3/25; заявл. 09.03.2011, № 2011110322/28; опубл. 20.03.2013, Бюл. № 8.

9. Жаринов И. О., Жаринов О. О. Исследование распределения оценки разрешающей способности преобразования Грассмана в системах кодирования цвета, применяемых в авионике // Программная инженерия. 2014. № 8. С. 40—47.

10. Корн Г., Корн Т. Справочник по математике (для научных работников и инженеров). М.: Наука, 1973. 832 с.

I. O. Zharinov, Assistant Professor, Chef of Department, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (University ITMO), Chef of learning-scientists center, SPb Scientific Design Bureau "Electroavtomatika" n. a. P. A. Efimov, e-mail: igor_rabota@pisem.net, O. O. Zharinov, Assistant Professor, Saint-Petersburg State University of Aerospace Equipment

The Research of Uniformchromaticity Property of Color Spaces, used in Avionics Equipment

The problem of the research of uniformchromaticity property of color spaces, used in avionics equipment, is considered. Uniformchromaticity property of color spaces is decisive for calculation of quantitative measure of colors difference. Development of functional software for avionics indication equipment is based on color spaces, with graphical representation on XY- and UV-chromaticity planes. In according with modern standards (manuals R 52324—2005, R 52870—2007) it is recommended to use the coordinate system, based on UV-plane, as uniformchromatical one. Histograms of chromaticity coordinates distribution, obtained for both XY- and UV-planes by the use of mathematical modeling, are presented. There was established that chromaticity coordinates distributions on both planes significantly differ from uniform law. Taking this into account, both of considered color spaces can not be accepted as true uniform colour spaces, and hence, an estimation of colors difference value won't keep constant accuracy within the whole area of XY- and UV- chromaticity planes. Significant nonuniformity of chromaticity coordinates may be observed in every part of defined chromaticity area of XY-plane. For UV-plane, nonuniformity areas tend to be close to white color point and also concentrated along the lines drawn between points of some colors, namely: white-magenta, white-yellow, white-sky blue. Areas of non-equal contrast on UV-chromaticity plane were localized during the research.

Keywords: color spaces, on-board indication equipment, avionics, uniformchromaticity property

References

1. Paramonov P. P., Kostishin M. O., Zharinov I. O., Nechaev V. A., Sudarchikov S. A. Princip formirovaniya i otobrazheniya massiva geoinformacionnykh dannykh na jekran sredstv bortovoj indikacii. *Nauchno-tehnicheskij vestnik informacionnykh tehnologij, mehaniki i optiki*. 2013. N. 6. P. 136—142.
2. Zharinov I. O., Zharinov O. O. *Bortovye sredstva otobrazheniya informacii na ploskih zhidkokristallicheskih paneljah: ucheb. posobie*. Informacionno-upravljajushhie sistemy. SPb.: GUAP, 2005. 144 p.
3. Kostishin M. O., Zharinov I. O., Zharinov O. O. Issledovanie vizual'nykh harakteristik sredstv otobrazheniya pilotazhno-navigacionnykh parametrov i geoinformacionnykh dannykh v avionike. *Informacionno-upravljajushhie sistemy*. 2014. N. 4. P. 61—67.
4. Barber S. et al. US Patent 7,417,641 B1: Aeronautical chart display apparatus and method; Aug. 26. 2008.
5. Kryzhevich L. S. Vybor i preobrazovanie cvetovogo prostranstva. *Uchenye zapiski: jelektronnyj nauchnyj zhurnal Kurskogo gosudarstvennogo universiteta*. 2011. N. 4. P. 22—28.
6. Lozhkin L. D. Differencial'naja kolorimetrija v televideanii: Avtoref. dis. d-ra tehn. nauk: SPb., 2014. 31 p.
7. Lozhkin L. D., Neganov V. A. Sposob preobrazovanija cvetovogo prostranstva. Patent 2494461 S2. MPK G06K 9/68; zajavl. 08.07.2011, № 011128504/08; opubl. 27.09.2013. Bjul. N. 27.
8. So I. A., Malyhina G. F. Sposob i ustrojstvo dlja izmerenija cvetovykh razlichij (varianty). Patent 2477843 S2. MPK G 01J 3/25; zajavl. 09.03.2011, № 2011110322/28; opubl. 20.03.2013, bjul. N. 8.
9. Zharinov I. O., Zharinov O. O. Issledovanie raspredelenija ocenki razreshajushhej sposobnosti preobrazovanija Grassmana v sistemah kodirovanija cveta, primenjaemykh v avionike. *Programmnaja ingeneria*. 2014. N. 8. P. 40—47.
10. Korn G., Korn T. *Spravochnik po matematike (dlja nauchnykh rabotnikov i inzhenerov)*. M.: Nauka, 1973. 832 p.

ИНФОРМАЦИЯ



V Международная научно-техническая конференция
"Открытые семантические технологии проектирования
интеллектуальных систем"
Open Semantic Technologies for Intelligent Systems
OSTIS-2015

19 – 21 февраля 2015 г., Белорусский государственный университет
информатики и радиоэлектроники, Минск, Республика Беларусь

НАПРАВЛЕНИЯ РАБОТЫ КОНФЕРЕНЦИИ

- Принципы, лежащие в основе семантического представления знаний, и их унификация. Типология знаний и особенности семантического представления различного вида знаний и метазнаний. Связи между знаниями и отношения, заданные на множестве знаний. Семантическая структура глобальной базы знаний, интегрирующей различные накапливаемые знания
- Языки программирования, ориентированные на параллельную обработку семантического представления баз знаний
- Модели решения задач, в основе которых лежит обработка знаний, осуществляемая непосредственно на уровне семантического представления обрабатываемых знаний. Семантические модели информационного поиска, интеграции знаний, анализа корректности и качества баз знаний, сборки информационного мусора, оптимизации баз знаний, дедуктивного и индуктивного вывода в базах знаний, правдоподобных рассуждений, распознавания образов, интеллектуального управления. Интеграция различных моделей решения задач
- Семантические модели восприятия информации о внешней среде и отображения этой информации в базу знаний
- Семантические модели мультимодальных пользовательских интерфейсов интеллектуальных систем, в основе которых лежит семантическое представление используемых ими знаний, и унификация этих моделей
- Семантические модели естественно-языковых пользовательских интерфейсов интеллектуальных систем. Структура семантического представления лингвистических баз знаний, описывающих естественные языки и обеспечивающих решение задач понимания естественно-языковых текстов и речевых сообщений, а также задач синтеза естественно-языковых текстов и речевых сообщений, семантически эквивалентных заданным фрагментам баз знаний
- Интегрированные комплексные логико-семантические модели интеллектуальных систем, основанные на семантическом представлении знаний, и их унификация
- Различные технические платформы и варианты реализации интерпретаторов унифицированных логико-семантических моделей интеллектуальных систем, основанных на семантическом представлении знаний
- Средства и методы, основанные на семантическом представлении знаний и ориентированные на проектирование различных типовых компонентов интеллектуальных систем (баз знаний, программ, решателей задач, интерфейсов)
- Средства и методы, основанные на семантическом представлении знаний и ориентированные на комплексное проектирование различных классов интеллектуальных систем (интеллектуальных справочных систем, интеллектуальных обучающих систем, интеллектуальных систем управления, интеллектуальных робототехнических систем, интеллектуальных систем поддержки проектирования и др.)
- Прикладные интеллектуальные системы, основанные на семантическом представлении используемых ими знаний

Рабочие языки конференции: русский, белорусский, английский.

Вся необходимая информация о предстоящей и предыдущих конференциях OSTIS
находится на сайте конференции <http://conf.ostis.net>

Н. Е. Антонова, аспирант, e-mail: myasn.naty@gmail.com,
А. А. Большаков, д-р техн. наук, проф., e-mail: bolshakov@overta.ru,
В. В. Лобанов, канд. техн. наук, доц., зам. зав. каф.,
В. В. Киселев, канд. техн. наук, асс., **Л. Г. Перова**, канд. техн. наук, асс., Саратовский государственный технический университет имени Ю. А. Гагарина

Программный комплекс поддержки принятия решений медицинской диагностики заболеваний с использованием стабилметрической платформы

Рассмотрена программная реализация системы поддержки принятия решений при медицинской диагностике заболеваний опорно-двигательного аппарата с применением стабилметрической платформы. Предложена архитектура интеллектуальной системы диагностики заболеваний с использованием комбинированных интеллектуальных процедур и встроеного модуля поддержки принятия решений, созданных на языке LabVIEW.

Ключевые слова: система поддержки принятия решений, архитектура системы диагностики, программный комплекс, LabVIEW

Введение

Обработка сигналов в медицине — одно из важных направлений развития математических методов и техники. В течение XX века в медицине и биологии продолжали развиваться методы анализа сигналов и широко использовались современные математические методы. К ним относятся такие области исследований, как анализ физиологических сигналов, создание математических моделей физиологических и патологических процессов, поддержка принятия решений при диагнозе больных, анализ и моделирование макромолекул, математическое моделирование лекарственных препаратов, разработка методов нелинейной динамики для описания физиологических и патологических процессов, использование методов искусственного интеллекта.

Вопросам компьютерной диагностики заболеваний посвящены работы ученых Н. А. Корневского, Е. С. Подвального, Б. А. Кобринского, Д. Ферруччи, Э. Козйра и др. [1, 2]. Однако диагностика заболеваний опорно-двигательного аппарата с использованием стабилметрической платформы является одной из областей медицины, которая практически не охвачена интеллектуальными информационными технологиями. Это связано, прежде всего, со сложностью строения опорно-двигательного аппарата и методов его исследования.

В научной литературе представлены многочисленные авторские методики или предложения по обработке данных, получаемых при регистрации статокинезиограмм [3]. Несмотря на большое многообразие

методов математической обработки статокинезиограмм, существует ряд стандартных статистических показателей, а также характеристики спектрального состава стабилограмм, индикаторы, характеризующие параметры статокинезиограмм, и другие специальные показатели.

Стабилография [4] обладает рядом существенных недостатков: неунифицированность методов; отсутствие критериев выбора информативных параметров статокинезиограммы; расплывчатые и неточные представления о границах нормальных значений стабилметрических параметров; отсутствие общепринятого представления о функциональной системе, осуществляющей функцию равновесия и координацию движений; неясность диагностической ценности изменений параметров равновесия при периферических и центральных поражениях нервной системы.

Эффективность стабилметрии в терапии существенно увеличится, если ее дополнить не только возможностью получения общих кривых, а также формированием категорий вероятных диагнозов [4].

Таким образом, совершенствование методов и алгоритмов анализа колебаний центра тяжести человека, как одного из важных диагностических методов, и разработка соответствующего программного обеспечения является востребованной и актуальной задачей.

Характеристика объекта исследования

При проведении стабилметрического исследования получают данные о состоянии двигательного коор-

динационной сферы человека, его трансформации и передают в режиме реального времени в обрабатывающий блок компьютера (рис. 1, см. третью сторону обложки). Эти данные используют для получения объективной информации для диагностики и оценки состояний двигательной-координационной сферы, системы пространственной ориентации, а также для проведения восстановительного лечения, тренировок, либо специальных функциональных проб. При этом текущие данные о колебаниях проекции собственного центра масс человека преобразуются в определенный внешний сигнал для создания биологической обратной связи [5].

Из получаемой записи можно определить требуемое число характеристик и параметров. Наиболее значимыми являются два изображения (статокнезиограмма и стабиллограмма), шесть параметров (X -среднее, площадь, LFS, ANØ2, VFY, коэффициент Ромберга) и пять функций.

На стабиллограмме по оси абсцисс отложено время, а по оси ординат — последовательные положения центра давления. Таким образом, получают два графика — для X (движения вправо-влево) и для Y (движения вперед-назад). Шкала оси ординат имеет для каждого графика значение полной шкалы. Заметим, что следует быть внимательными к изменениям этих шкал, так как обычно они различаются от графика к графику.

Статокнезиограмма (рис. 2) отображает последовательность дискретных точек положения центра давления по отношению к системе координат с точкой отсчета в центре полигона опоры. Только эти положения центра давления точно отражаются на рисунке в соответствии с указанной шкалой. Полученный контур полигона опоры не всегда точно соответствует контуру полигона опоры обследуемого пациента и зависит от применяемого программного обеспечения.

Процедуру диагностики можно представить как задачу определения зависимости между симптомами

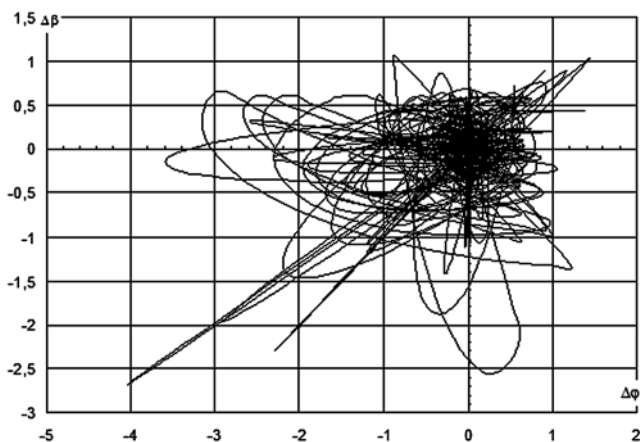


Рис. 2. Статокнезиограмма:

$\Delta\beta$ — угол отклонения фрактальной плоскости, °; $\Delta\phi$ — угол отклонения сагиттальной плоскости, °

(входными данными) и диагнозом (выходными данными). Для создания эффективной системы диагностики целесообразно использовать как регулярные методы (методы математического и статистического анализа), так и методы на основе искусственного интеллекта [6].

Справедливость совместного применения регулярных методов и методов искусственного интеллекта подтверждает анализ данных, используемых при медицинской диагностике, который показывает, что они обладают такими особенностями, как качественный характер информации, наличие пропусков данных, взаимная коррелированность, большое количество переменных при относительно небольшом числе наблюдений. Кроме этого, значительная сложность объекта наблюдения (заболеваний) нередко не позволяет построить даже вербальное описание процедуры диагноза врачом.

Таким образом, создаваемая система должна обладать следующими функциями: сбор, обработка и хранение поступающей информации от стабиллоплатформы, выбор характеристик для обработки, предварительная обработка и классификация, проведение статистического анализа, построение нейронной сети, прогноз и автоматизированное формирование заключения.

Постановка задачи исследования

Для повышения эффективности диагностики заболеваний опорно-двигательного аппарата с использованием стабиллометрической платформы необходимо применение комбинированных методов диагноза и поддержки принятия решений. Для этого необходимо решить следующие задачи:

- оценить достоинства и недостатки существующего и разрабатываемого программного обеспечения;
- выполнить анализ и выбор среды программной реализации интеллектуальной системы;
- разработать архитектуру интеллектуальной системы медицинской диагностики;
- построить интеллектуальную систему диагностики на основе комбинированного подхода.

Рассмотрим далее более подробно предлагаемую архитектуру интеллектуальной системы медицинской диагностики с учетом требований к ее функциональным возможностям.

Описание архитектуры интеллектуальной системы медицинской диагностики

Для достижения поставленной цели предлагается создать гибридную экспертную систему диагностики, в основе функционирования которой содержится комбинированный метод построения диагностирующей процедуры. Архитектура интеллектуальной системы медицинской диагностики представлена на рис. 3. Согласно

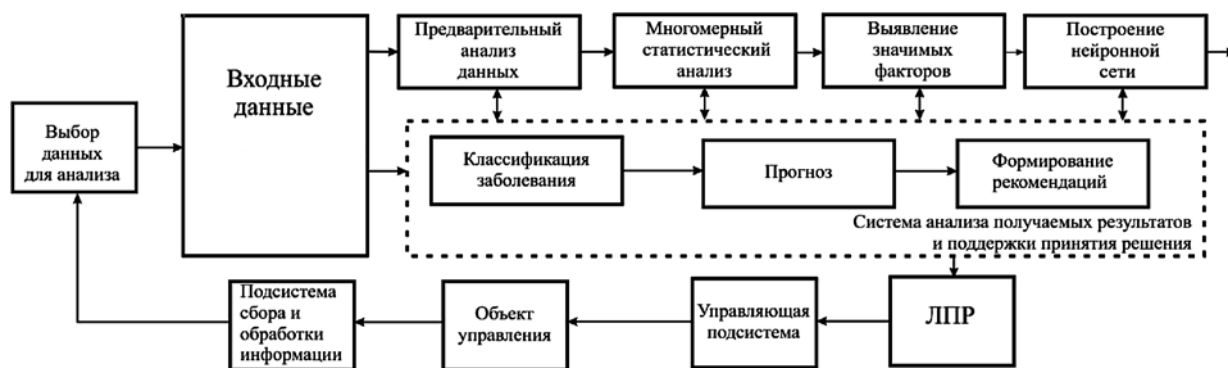


Рис. 3. Архитектура интеллектуальной системы медицинской диагностики

этой архитектуре блок входной информации формирует матрицу исходных данных размерности $M \times N$, где M — число переменных (симптомов, характеристик и т. п.), а N — число наблюдений (больных).

Модуль предварительного анализа служит для исследования особенностей следующей входной информации: наличия и относительного веса переменных качественного характера, пропусков данных; соотношения числа переменных и наблюдений и т. д.

Эксперт (лицо, принимающее решение, ЛПР) принимает решение о настройках модуля предварительного анализа данных по результатам информации предыдущего блока. Этот блок позволяет при необходимости восстановить пропущенные данные одним из выбранных способов, отбросить «дикие» измерения» и т. п.

Модуль многомерного статистического анализа числовых и нечисловых данных позволяет выявить наиболее информативные переменные с точки зрения влияния на диагноз. Его использование позволяет существенно сократить число рассматриваемых переменных и снизить требования по числу наблюдений для построения нейронной сети. В качестве многомерного метода обработки данных целесообразно использовать дискриминантный анализ с итерационной процедурой исключения переменных.

В блоке выбора данных для анализа принимается решение о составе переменных, которые используются для построения нейронной модели диагноза заболеваний. При этом эксперт может дополнить перечень подобных переменных, формируемых по результатам работы предыдущего модуля [7].

Следующий блок — блок построения нейронной классифицирующей сети по заданному множеству входных переменных и диагнозу. Опыт решения подобных задач авторами показал, что целесообразно использовать двухслойные нейронные сети прямого распространения с сигмоидальной функцией активации [8].

В системе анализа получаемых результатов и поддержки принятия решения делается вывод о достижении поставленной цели — создании нейронной сети для формирования диагноза заболевания с заданной точностью. Если задача не решена, то осуществляется переход к предыдущим блокам, и вновь повторяется итерационная комбинированная процедура построения модели диагностики.

Программная реализация

Программный модуль поддержки принятия решений разработан с использованием графического языка G программного комплекса LabVIEW компании National Instruments. Он предназначен для диагностики заболеваний опорно-двигательного аппарата и может применяться в отраслях, связанных с медициной и профилактикой заболеваний, а также в образовательном процессе при обучении специалистов соответствующего профиля. Программа обеспечивает выполнение следующих функций: установление заболеваний ишемии головного мозга; поясничный остеохондроз; гоартроз. С ее помощью могут быть получены заключения о сущности болезни и состоянии пациента, выраженные в принятой медицинской терминологии.

Также важным является наличие средств для организации визуального и звукового каналов обратной связи [10].

Для использования разных режимов демонстрации изображений на нескольких экранах в целях тестирования или тренинга используется стандартная опция Windows для управления несколькими мониторами, в том числе для селективного разделения изображений, «расширения» экрана. Обычно в качестве второго монитора (для организации визуального канала) используют стандартный компьютерный дисплей 19' или 27', или другого размера, на регулируемом штативе. Используют также проектор, обеспечивающий достаточную контрастность. Для организации акустического

канала обычно достаточно встроенных динамиков ноутбука. При необходимости можно подключить внешние колонки.

Проведенный анализ позволил выявить преимущества разработанного программного обеспечения по сравнению с существующим аналогом. К преимуществам предлагаемого комплекса можно отнести одновременную обработку стабилметрических данных в трех модулях (нейронной сети, статистической обработки и математического анализа), их сравнение и анализ; обработку, запись и хранение в базе данных; визуализацию результатов; постановку диагноза и выдачу рекомендаций.

Разработанный программный комплекс (рис. 4, 5, см. третью сторону обложки) обеспечивает прием данных со стабилметрической платформы, последующий их анализ и вывод результатов в виде статокинезиограммы (траектория центра давления на плоскость опоры). Комплекс также предоставляет возможность распечатывать протоколы тестирований и формировать динамику результатов обследований с последующей постановкой предварительного диагноза с применением метода конъюнкции значений параметров, попавших в оценочные интервалы.

При создании программного обеспечения использовались предложенные авторами математические модели и алгоритмы обработки стабилметрического сигнала в сочетании с методами многомерного статистического анализа [9, 11], что позволило расширить диагностические возможности стабилметрии.

Главное преимущество разработанного программного обеспечения — это возможность постановки предварительного диагноза и выдача соответствующего заключения и рекомендаций [12, 13].

Учитывая особенности регистрации стабилграфических показателей, возможность получения их без отвлечения человека от выполняемой деятельности и высокую чувствительность способа, целесообразно использовать стабилграфию как удобный метод оценки динамики функционального состояния человека в условиях функциональных нарушений, различных заболеваниях центральной нервной системы и опорно-двигательного аппарата. Компьютерная стабилграфия является перспективным методом при диагностике, а также при реабилитации (с использованием метода зрительной обратной связи) нарушений после детского церебрального паралича, постинсультных гемипарезов и других нарушений двигательных функций.

Таким образом, на основе предложенной интеллектуальной системы диагностики заболеваний расширяются диагностические возможности стабилметрии.

Заключение

Предложен гибридный модуль поддержки принятия решений, отличающийся использованием мето-

дов статистического анализа и нейронных сетей, что позволило разработать соответствующее специальное математическое обеспечение для поддержки управленческих решений в системах медицинской диагностики. Реализована программа на языке программирования LabVIEW компании National Instruments, которая предназначена для диагностики заболеваний опорно-двигательного аппарата. Программный комплекс может применяться в отраслях, связанных с медициной и профилактикой заболеваний, а также в образовательном процессе при обучении специалистов соответствующего профиля. На основе предложенной интеллектуальной системы диагностики заболеваний расширен диапазон диагностических возможностей стабилметрии.

Список литературы

1. **Кобринский Б. А.** Системы искусственного интеллекта в медицине: состояние, проблемы и перспективы // *Новости искусственного интеллекта*. 1995. № 2. С. 45–46.
2. **Рахманова З. Б.** Медицинское знание: от прошлого к будущему, или экспертные системы в медицине // *Новости искусственного интеллекта*. 1995. № 3. С. 33–36.
3. **Мясникова Н. Е., Проскураков Г. М.** Медико-биологические основы работы стабилметрических платформ // *Сб. науч. тр. «Анализ, синтез и управление в сложных системах»*. Саратов: Изд-во СГТУ, 2009. С. 68–78.
4. **Скворцов Д. В.** Клинический анализ движений: Анализ походки. М.: НМФ «МБН», 1996. 344 с.
5. **Смелянский Р. Л.** Модель функционирования распределенных вычислительных систем // *Вестник МГУ. Сер. 15. Вычислительная математика и кибернетика*, 1990. Вып. 3. С. 14–26.
6. **Большаков А. А., Казимирова Н. Е., Лобанов В. В.** Комбинированный интеллектуальный метод медицинской диагностики // *Математические методы в технике и технологиях: сб. трудов 17 Междунар. науч. конф.* Кострома: Изд-во КГТУ, 2004. Т. 4. С. 132–138.
7. **Большаков А. А., Лобанов В. В.** Диагностическая экспертная система сердечно-сосудистых заболеваний // *Математические методы в технике и технологиях: сб. тр. 14 Междунар. науч. конф.* — Смоленск: Изд-во Смоленского ф-ла МЭИ, 2001. Т. 5. С. 5–7.
8. **Большаков А. А., Лобанов В. В., Смирнов Д. В.** Использование нейронных сетей для решения ряда оптимизационных задач // *Математические методы в технике и технологиях: сб. тр. 15 Международ. науч. конф.* Тамбов: Изд-во ТГТУ. 2002. Т. 5. С. 78–81.
9. **Юнкеров В. И., Григорьев С. Г.** Математико-статистическая обработка данных медицинских исследований. СПб.: Изд-во Воен.-мед. акад., 2002. 266 с.
10. **Нейронные сети.** Statistica Neural Networks. Методология и технологии современного анализа данных. М.: Горячая линия — Телеком, 2008. 392 с.
11. **Большаков А. А., Каримов Р. Н.** Методы обработки многомерных данных и временных рядов: учеб. пособие для вузов. М.: Горячая линия — Телеком, 2007. 522 с.
12. **Колбин И. С.** Программный комплекс для решения задач математического моделирования с использованием нейросетевой методологии // *Программная инженерия*. 2013. № 2. С. 25–31.
13. **Христанков А. С.** Опыт обучения методам проектирования программных систем // *Программная инженерия*. 2012. № 4. С. 44–47.

N. E. Antonova, Postgraduate Student, e-mail: myasn.naty@gmail.com, A. A. Bolshakov, Professor, e-mail: bolshakov@overta.ru, V. V. Lobanov, Deputy Head of Chair, V. V. Kiselev, Assistant, L. G. Perova, Assistant, Yuri Gagarin State Technical University of Saratov

Program Complex of Support of Decision-Making of Medical Diagnostics of Diseases with Use of the Stabilometry Platform

The main idea of this article is to make a program realization of system of support of decision-making at medical diagnostics of diseases of the musculoskeletal device with application of a stabilometry platform is considered. The authors propose a methodology for a processing of medical data. The authors also propose the architecture of intellectual system. This architecture of intellectual system of diagnostics of diseases with use of the combined intellectual procedures and the built-in module of support the decision-making created in the LabVIEW language is offered. Developed software system will significantly improve the efficiency of diagnosis of various diseases of the musculoskeletal system. Given the characteristics of the registration stabilographic indicators, getting them without the distraction of a person from the activities performed and the high sensitivity of the method should be used as a convenient method Stabilographa assess the dynamics of the functional state of human rights in terms of functional disorders, various diseases of the central nervous system and musculoskeletal system. Computer stabilography is a promising method for the diagnosis, as well as rehabilitation (using the visual feedback) disorders after cerebral palsy, post-stroke hemiparesis and other disorders of motor function. This articles may be interesting for doctors, scientists. It may be applied in medicine and technic technology.

Keyword: system of support of decision-making, architecture of system of diagnostics, program complex, LabVIEW

References

1. **Kobrinский B. A.** Systemy iskusstvennogo intellekta v medicine: sostojanie, prolemy i perspektivy. *Novosti iskusstvennogo intellekta*. 1995. N. 2. P. 45–46.
2. **Rahmanova Z. B.** Medicinskoe znanie: ot proshlogo k budushemu, ili jekspertnye sistemy v medicine. *Novosti iskusstvennogo intellekta*. 1995. N. 3. P. 33–36.
3. **Mjasnikova N. E., Proskurjakov G. M.** Mediko-biologicheskie osnovy raboty stabilometricheskikh platform. *V sb. nauchnyh trudov «Analiz, sintez i upravlenie v slozhnyh sistemah»*. Saratov: Izd-vo SGTU, 2009. P. 68–78.
4. **Skvortsov D. V.** *Klinicheskij analiz dvizhenij: Analiz pohodki*. M.: NMF «MBN», 1996. 344 p.
5. **Smelyanskiy R. L.** Model' funkcionirovanija raspredeljonnyh vychislitel'nyh system. *Vestnik MGU. Ser.15, Vychislitel'naja matematika i kibernetika*. 1990. N. 3. P. 14–26.
6. **Bol'shakov A. A., Kazimirova N. E., Lobanov V. V.** Kombinirovannyj intellektual'nyj metod medicinskoj diagnostiki. *Matematicheskie metody v tehnike i tehnologijah: sb. trudov 17 Mezhdunarod. nauch. konf.* Kostroma: Izd-vo KGTU, 2004. Vol. 4. P. 132–138.
7. **Bol'shakov A. A., Lobanov V. V.** Diagnosticheskaja jekspertnaja sistema serdechno-sosudistyh zabolevanij. *Matematicheskie metody v tehnike i tehnologijah: sb. trudov 14 Mezhdunarod. nauch. konf.* Smolensk: Izd-vo Smolenskogo f-la MJeI. 2001. Vol. 5. P. 5–7.
8. **Bol'shakov A. A., Lobanov V. V., Smirnov D. V.** Ispol'zovanie nejronnyh setej dlja reshenie rjada optimizacionnyh zadach. *Matematicheskie metody v tehnike i tehnologijah: sb. trudov 15 Mezhdunar. nauch. konf.* Tambov: Izd-vo TGTU. 2002. Vol. 5. P. 78–81.
9. **Junkerov V. I., Grigor'ev S. G.** *Matematiko-statisticheskaja obrabotka dannyh medicinskih issledovanij*. SPb.: Izd-vo Voen.-med. akademii, 2002. 266 p.
10. **Nejronnye seti**. Statistica Neural Networks. Metodologija i tehnologii sovremennoogo analiza dannyh. M.: Gorjachaja Linija — Telekom, 2008. 392 p.
11. **Bol'shakov A. A., Karimov R. N.** Metody obrabotki mnogomernykh dannyh i vremennyh rjadov. Ucheb. posobie dlja vuzov. M.: Gorjachaja linija — Telekom, 2007. 522 p.
12. **Kolbin I. S.** Programmnyj kompleks dlja reshenija zadach matematicheskogo modelirovanija s ispol'zovaniem nejrosetevoj metodologii. *Programmnaja ingenerija*. 2013. N. 2. P. 25–30.
13. **Hritankov A. S.** Opyt obuchenija metodam proektirovanija programnyh sistem. *Programmnaja ingenerija*. 2012. N. 4. P. 44–47.

ООО "Издательство "Новые технологии". 107076, Москва, Стромьинский пер., 4

Дизайнер Т. Н. Погорелова. Технический редактор Е. М. Патрушева. Корректор Е. В. Комиссарова

Сдано в набор 05.09.2014 г. Подписано в печать 20.10.2014 г. Формат 60×88 1/8. Заказ П11114
Цена свободная.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.