

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 23

2017

№ 5

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ОБЩИЕ ВОПРОСЫ

Юсупов Р. М., Мусаев А. А. К оцениванию эффективности информационных систем. Методологические аспекты 323

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Ульянов М. В., Сметанин Ю. Г. Об одном характеристическом функционале слов над конечным алфавитом 333

Никифоров Д. А., Корж Д. В., Сиваков Р. Л. Обзор инструментов для валидации XML-документов с помощью правил контроля, описанных на объектном языке ограничений (OCL) 342

Алгулиев Р. М., Набибекова Г. Ч. Методы определения степени интеграции стран в среде электронного государства 351

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

Левин В. И. Моделирование систем с неопределенностью с помощью полиинтервальной математики 358

Деменков Н. П., Микрин Е. А., Мочалов И. А. Нечеткое преобразование Лапласа в задачах нечеткого математического моделирования. Часть II 362

Литвин Ю. В., Ноздрин К. С., Полков А. А. Система виртуального прототипирования и моделирования информационно-управляющих систем реального времени 370

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Чан Ван Нгиа. Разработка параллельной архитектуры кодеров БЧХ и LDPC в стандартах систем цифрового телевизионного вещания второго поколения с регулируемыми параметрами кодирования на базе ПЛИС 376

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

Кудряшов А. П., Соловьев И. В. Реконструкция городской обстановки с учетом рельефа местности с использованием топографического плана и сервисов Google Maps 382

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Печенко И. С., Венгер О. В., Плоткин Д. А. ACES Tables: среда для создания и использования табличных спецификаций систем на кристалле 388

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

Барский А. Б. Нейросетевой метод ускоренного принятия решений операционной системой суперкомпьютера EPIC-архитектуры 394

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

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

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

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

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

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

CONTENTS

GENERAL QUESTIONS

- Yusupov R. M., Musaev A. A.** By Estimating the Effectiveness of Information Systems. Methodological Aspects 323

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Ulyanov M. V., Smetanin Yu. G.** On a Characteristic Functional for Words over a Finite Alphabet 333
- Nikiforov D. A., Korj D. V., Sivakov R. L.** A Survey of Tools for XML Validation Based on the Object Constraint Language (OCL) 342
- Alguliyev R. M., Nabibayova G. Ch.** Methods for Determination of the Degree of Integration of Countries in e-Government Environment 351

MODELING AND OPTIMIZATION

- Levin V. I.** Modeling of Systems with Uncertainty by Polyinterval Mathematics 358
- Demchenkova N. P., Mikrin E. A., Mochalov I. A.** Fuzzy Transformation of Laplace in Tasks of Fuzzy Mathematical Modelling. Part II 362
- Litvin Iu. V., Nozdrin K. S., Polkov A. A.** System of Virtual Prototyping and Modeling Management Information Systems of Real Time 370

DIGITAL PROCESSING OF SIGNALS AND IMAGES

- Tran Van Nghia.** Development of the Parallel BCH and LDPC Encoders Architecture for the Second Generation Digital Video Broadcasting Standards with Adjustable Encoding Parameters on FPGA 376

GEOINFORMATION SYSTEMS

- Kudrjashov A. P., Soloviov I. V.** Reconstruction of the City Environment Based on Terrain Using the Topographic Map and Services Google Maps 382

APPLIED INFORMATION TECHNOLOGIES

- Pechenko I. S., Venger O. V., Plotkin D. A.** ACES Tables: a Framework for Creation and Usage of SoC Architectural Specifications in Table Form 388

NEUROTECHNOLOGIES

- Barsky A. B.** Neural Network Method of Accelerated Adoption of the Operating System, Making Supercomputer EPIC-Architecture 394

Editor-in-Chief:

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

Deputy Editor-in-Chief:

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

Chairman:

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

Editorial Board Members:

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

Editors:

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

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

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

ОБЩИЕ ВОПРОСЫ GENERAL QUESTIONS

УДК 621.382.1

Р. М. Юсупов, д-р техн. наук, проф., чл.-корр. РАН, e-mail: yusupov@iias.spb.su,
Санкт-Петербургский институт информатики и информатизации РАН,
А. А. Мусаев, д-р техн. наук, проф., e-mail: amusaev@technolog.edu.ru,
Санкт-Петербургский государственный технологический институт

К оцениванию эффективности информационных систем. Методологические аспекты

Рассмотрена задача построения методологии оценки эффективности информационных систем и технологий, позволяющая количественно оценить результативность их применения. Особенностью информационных систем является их сервисный характер, что приводит к необходимости косвенного оценивания их эффективности через терминальный положительный эффект метасистемы, в интересах которой данная система была создана или адаптирована.

Ключевые слова: информация, информационные системы, информационные технологии, эффективность, качество, методология, вероятностный подход, системный анализ

Введение

Процесс создания и применения любой *информационной системы* (ИС) или *информационной технологии* (ИТ) неизбежно связан с проблемой оценки их эффективности. При этом возникает целый ряд сложных вопросов, обусловленных спецификой объекта, для переработки и хранения которого ИС создавалась — *информации*. Над тайнством природы информации многие годы трудились философы и прикладные специалисты [1–5], однако их успехи существенно отстают от взрывоподобного роста практических приложений ИТ. В результате возникшего разрыва между теоретическим осознанием природы информационных процессов и быстро прогрессирующим развитием практических реализаций ИС образовался пробел в области оценки их эффективности, связанный с общей задачей квалиметрии информационных систем.

Заметим, что ИС, в том числе и программные комплексы, можно рассматривать как терминальные реализации ИТ. В связи с этим в дальнейшем ограничимся задачей оценки эффективности ИС, имея в виду, что предложенная в статье методика распространяется и на ИТ.

Особую проблему при оценивании эффективности информационных систем создает тот факт, что большинство ИС относится к сервисным или инструментальным системам. Это означает, что сами по себе они не создают итогового результата, который можно было бы оценить в терминах экономической эффективности. Иными словами, ре-

зультативность ИС является опосредованной через функционирование других видов систем — технических, экономических, военных, социальных и т.п.

Задача оценивания эффективности ИС неоднозначна и многопланова. Однако реалии времени требуют практических методик, способных конструктивно оценить результативность разрабатываемой или уже действующей ИС. В связи с этим возникает задача формирования методологической базы, увязывающей оценку эффективности ИС с общей теорией системной квалиметрии. Решению данной задачи и посвящена настоящая работа.

1. Анализ подходов к оценке эффективности информационных систем

Отсутствие единого подхода к задаче оценивания эффективности ИС является закономерным следствием диалектического противоречия между общим и частным. Общий подход слабо отражает существенные особенности информационных систем, частные технологии оценки эффективности отдельных решений не подходят для других приложений.

В настоящее время уже разработана (на уровне министерств) и рекомендована к использованию методология оценки эффективности инвестиционных проектов. Информационные системы обычно также относятся к инновационным решениям и, теоретически, они должны быть оценены на основе существующих методик. Однако *de facto* применение экономических методик анализа инноваций к ИС наталкивается на существенные трудности.

Главной причиной этого является нематериальный характер объекта приложений ИС — информации. В частности, следствием нематериального характера информационного "сырья" является косвенный характер влияния информационных инноваций на экономическую эффективность проектов. Сложность задачи оценивания эффективности внедрения ИС привела к скептическому отношению относительно возможности априорного анализа инновационных ИТ-проектов в целом.

В частности, авторитетный американский журнал CIO Insight утверждает, что более половины фирм вообще не анализируют окупаемость вложений в развитие информационной структуры (даже при внедрении столь существенных проектов, как ERP или корпоративные порталы). Вердикт журнала: "Состояние оценивания экономического эффекта от внедрения ИТ выглядит удручающе" [6]. По результатам опроса российских предпринимателей [7], в целом, сохраняется вера в окупаемость вложений в информационную инфраструктуру компании (до 47 % опрошенных). Однако 80 % компаний, из числа не согласившихся на внедрение ИТ, в качестве причины отказа называют недостаточность эффекта от вложений.

Насколько эффективны современные западные методики оценки эффективности инвестиционных проектов? Речь идет, прежде всего, о таких интегральных показателях, как *TCO* (total cost of ownership — полная стоимость владения) и *ROI* (return on investment — возврат на инвестицию). По мнению ряда авторов [6, 7], отказ российских предпринимателей от использования этих показателей связан с иной ориентацией отечественного бизнеса. В РФ основной целевой функцией является не увеличение производительности труда, а рост отдачи от капитала.

Для ИТ-инноваций оба приведенных показателя являются несостоятельными в силу сложности оценки парциальной прибыли, обусловленной внедрением или модификацией информационных систем [8]. К тому же высочайшая скорость развития вычислительных средств крайне затрудняет оценку срока использования ИС. По результатам опроса [9], 70 % компаний находят показатель *ROI* слишком сложным для вычисления и 44 % — субъективным.

Важные результаты в отношении возможности применения традиционных методик оценки эффективности инноваций к ИС показали исследования П. Страссмана [10, 11]. Вывод оказался достаточно неожиданным: между ИТ-бюджетами и целевыми показателями результативности бизнеса статистических корреляций нет! Исключение составил показатель затрат на управление и администрирование, продвижение и поддержку продаж — *Sales, General & Administrative (SG & A)*. Заметим, что данный вывод подтверждает справедливость приведенного выше утверждения о косвенном характере

влияния ИТ на терминальную эффективность бизнеса. На основе приведенных исследований в качестве оценки реальной эффективности ИС был предложен показатель информационной продуктивности — *Information Productivity (IP)*: $IP = EVA/SG \& A$, где *EVA* (*Economic Value Add*) — объем прибыли компании после уплаты налогов и выплаты дивидендов по акциям [10—13]. *SG & A* здесь используется как параметр, чувствительный к внедрению ИТ. Анализ динамики *IP* показал, что лишь немногие компании смогли получить конкурентные преимущества за счет внедрения ИТ. Анализ этого явления приведен в работе [12].

В качестве альтернативы к бизнес-оценкам эффективности ИС в работе [8] приведены так называемые системные методы анализа, демонстрирующие целесообразность применения ИТ в различных сферах деятельности компаний. Примерами соответствующих методик могут служить:

- сбалансированная система показателей *BSC* (*Balanced Score Card*);
- система показателей ИТ *BITS* (*Balanced IT Score Card*);
- референтная модель производительности *PRM* (*Performance Reference Model*);
- бизнес-ценность ИТ *BVIT* (*Business Value of IT*);
- сервисная стратегия *ITIL* (*Information Technology Infrastructure Library Service Strategy*).

Очевидным недостатком приведенных определений является их низкая пригодность для построения точных количественных оценок эффективности ИС на разных этапах их жизненного цикла. Достаточно понятно, что построить гармоничную интегральную оценку эффективности ИТ-инноваций на таком обширном, плохо формализованном пространстве практически невозможно. Результатом применения размытых методик в конечном итоге является множество числовых характеристик, не допускающих ранжирования, а следовательно, и корректного сравнения вариантов реализации ИС без вмешательства субъективной экспертизы.

Следует заметить, что отсутствие объективных подходов к оценке эффективности ИТ, привели некоторых исследователей к парадоксальным выводам. Так, например, в работе Н. Карра [14] обосновывается вывод о том, что ИТ вообще не дают конкурентного преимущества. При этом приводятся вполне убедительные доводы: комплексная автоматизация предприятия на уровне ERP может стоить миллионы долларов и потребовать существенной перестройки всей информационной инфраструктуры предприятия. В то же время сомнительный положительный эффект достигается только косвенным образом, через увеличение уровня прозрачности и, как ожидаемого следствия, управляемости производством. При этом в силу фантастической скорости развития микроэлектроники и ИТ

развернутая дорогостоящая ERP может оказаться устаревшей уже к моменту ее внедрения.

В работах [15, 16] также указывается, что ИС сами по себе не дают преимуществ и лишь в совокупности с другими инициативами способны влиять на ключевые факторы успеха, создавать конкурентные преимущества. Для оценки эффективности в этих работах предлагается использовать процессный подход, разделяющий корпоративные процессы на основной, т.е. добавляющий стоимость к исходному продукту, и вспомогательный. Для промышленных предприятий информационные процессы естественно относятся к категории вспомогательных. Следовательно, эффективность ИС в этом случае также следует оценивать косвенно, через совершенствование управления производственно-го цикла предприятия.

Экспериментальное опровержение парадоксу непроизводительности ИТ было приведено в работе группы экспертов под руководством Э. Бриньолфсона [17]. В частности, было показано, что ИТ оказывают существенное и статистически значимое влияние на итоги работы компании, отдача на инвестиции в компьютерный капитал составила в среднем около 81 % и существенно превосходит отдачу от инвестиций в другие материальные активы.

В заключение обзора, посвященного эффективности ИС, следует указать на материалы проекта *IT-Value* [18], посвященные особенностям экономического анализа инвестиций в ИТ. Исходными постулатами проекта являются следующие утверждения:

- инвестиции в ИТ влияют на организацию в целом, поэтому объект исследования — это компании в целом, а не информационные системы или ИТ-проекты по отдельности;
- развитие и эксплуатация корпоративных ИТ рассматривается как единый и взаимосвязанный комплекс процессов, который влияет на бизнес-результаты компании;
- существуют объективные закономерности влияния ИТ на экономические результаты компании, которые не зависят от того, что думают о них руководители организации или ИТ-директора и др.

Важной особенностью проекта является привязка методов оценки эффекта от инвестиций в ИТ к особенностям российской специфики.

Кратким выводом из приведенного обзора является вывод об актуальности задачи построения конструктивной количественной методики анализа эффективности ИС, допускающей априорную оценку их результативности и возможность объективного сравнения вариантов их реализации с учетом нематериальной специфики информационных сред. В качестве теоретической платформы для построения такой методики предлагается проверенное сочетание вероятностно-статистического подхода к оцениванию эффективности с общими принципами системного анализа [19–23].

2. Основные понятия теории эффективности информационных систем

Выбор критериальной функции, предназначенной для количественной или качественной оценки результативности ИС, достаточно субъективен. Наиболее универсальным показателем качества какой-либо системы является *эффективность* ее функционирования.

Эффективность является одним из фундаментальных свойств любой системы и непосредственно сопоставляется с результативностью ее применения. В связи с этим понятие эффективности системы во многих случаях используется в качестве основного показателя качества ее функционирования.

Для описания столь сложного и многопланового понятия как эффективность применяется целый набор разнородных показателей, отражающих те или иные квалиметрические признаки. Вариант неполной классификации показателей эффективности ИС приведен на рис. 1.

Принципиальным вопросом для оценки эффективности ИС является размерность ее показателя: одномерность (или скалярность) интегрального показателя эффективности является существенной. В этом случае сравнительный анализ вариантов решения поставленной задачи осуществляется путем их прямого ранжирования на основе очевидного упорядочения значений скалярного интегрального показателя эффективности.

В случае отказа от одномерного показателя эффективности и перехода к векторному показателю, элементами которого являются частные характеристики формируемого решения, задача выбора наилучшего варианта не только существенно усложняется, но и становится не вполне корректной. Действительно, сравнение и упорядочение ряда векторов, отвечающих набору возможных решений, могут быть выполнены лишь путем их скаляризации на основе той или иной метрики. В свою очередь, выбор подходящей метрики всегда несет в себе элементы субъективизма. В лучшем случае можно ожидать выбор метрики на основе экспертных оце-



Рис. 1. Классификация показателей эффективности информационных систем

нок, когда весовые коэффициенты частных показателей определяют путем специальной обработки результатов опроса группы или групп специалистов в соответствующих областях знаний.

Тем не менее, векторные показатели эффективности также распространены на практике, поскольку позволяют осуществить естественную декомпозицию общей проблемы повышения эффективности функционирования ИС на ряд частных вопросов, каждый из которых может составлять самостоятельную научную задачу последующего иерархического уровня. Так, например, комплексная проблема совершенствования ИС состоит из задач повышения оперативности доступа к хранимой информации, уровня программной совместимости с другими системами хранения информации (СХИ) и с информационными приложениями, степени безопасности хранения данных, степени доступности данных для пользователей, возможности формирования нестандартных запросов и т. д. Необходимость учета частных показателей приводит к специфическим методам анализа и синтеза систем, основанным на многокритериальном подходе.

На вербальном уровне эффективность ИС определяется как количественная мера способности системы выполнять возложенные на нее задачи. Прямой эффект от применения той или иной системы оценивается так называемой *функциональной эффективностью*. Наряду с функциональной эффективностью широко используется понятие *экономической эффективности*, в котором достигнутый полезный эффект соотносится с экономическими затратами, потребляемым ресурсом и т. п.

Окончательный выбор показателя и критерия эффективности ИС осуществляется исходя из цели, в интересах которой создавалась эта система. Так, например, АСУ, рассматриваемая как информационная система, была создана в целях повышения эффективности формируемых управляющих решений и их реализации, ИС *образовательной системы* (ОС) — в целях повышения качества подготовки выпускников и т. п.

Таким образом, основные показатели эффективности ИС формируются на основе требований, предъявляемых к ней со стороны метасистемы — той или иной иерархически вышестоящей системы, в интересах которой она создается.

Необходимость изучения ИС в иерархическом метасистемном ряду привела к разделению показателей эффективности на *собственные* и *несобственные*. Собственные (или *внутренние*) показатели предназначены для изучения системы *по разомкнутой схеме*, когда ИС рассматривается как самостоятельная, не связанная с метасистемой. Несобственные (или *внешние*) показатели отражают влияние качества функционирования изучаемой ИС на характеристики метасистемы с учетом выбранных критериев эффективности.

Достаточно очевидно, что даже самый полный набор собственных технических характеристик ИС (уровни полноты и достоверности обрабатываемой информации, оперативность, функциональная устойчивость, пропускная способность и т. д.) является недостаточным для интегральной оценки ее эффективности: результирующая оценка результативности исследуемой системы, как уже отмечалось, должна восходить к показателям качества вышестоящей метасистемы.

Следует заметить, что далеко не всегда удается строго, количественно оценить показатели эффективности сложной ИС. В этом случае переходят к качественным шкалам оценивания — номинальной или ординальной. Во многих случаях качественная оценка эффективности ИС формируется в результате опроса экспертов с последующей статистической обработкой их заключений.

В зависимости от степени достижения поставленной цели критерии эффективности могут быть отнесены к категории необходимых или достаточных. В случае если критерий отражает принципиальную допустимость ИС (с достигнутым уровнем эффективности) для решения стоящих перед ней задач, то его следует отнести к категории необходимых. Типичным необходимым критерием является превышение показателем эффективности ИС некоторого *a priori* заданного порога, определенно-го техническим заданием на разработку ИС. Достаточные критерии эффективности ИС могут носить самый разный характер: в их роли могут выступать достижение показателем эффективности максимального значения в выбранном диапазоне исходных данных или, при заданных ограничениях, превышение показателем эффективности соответствующего показателя системы-прототипа и др.

В большинстве случаев, особенно при использовании векторных показателей, оказывается целесообразным сочетать необходимые и достаточные критерии. В частности, допускается оценивать эффективность ИС на основе метода главного показателя, когда один из показателей (главный), например достоверность отображения контролируемой ситуации, относится к категории достаточных и подлежит максимизации, а остальные показатели — к категории необходимых, т. е. отвечающих заданным пороговым нормам. К числу последних можно отнести полноту отображения ситуации, вычислительную реализуемость ИС, чувствительность к контролируемым параметрам, устойчивость к возмущениям и вариациям вероятностной структуры исходных данных и т. п.

Следует отметить, что для полноты классификации показателей эффективности ИС, их можно разделить по характеру измерений (наблюдений) на прямые и косвенные, а также на абсолютные и относительные. В соответствии с этим и оценивание их величин бывает прямым и косвенным, абсолют-

ным и относительным. Косвенное оценивание реализуется в условиях, когда прямое наблюдение показателей эффективности ИС невозможно и, разумеется, уступает последнему по информативности, а, как правило, и по объективности (достоверности).

Разделение показателей на абсолютные и относительные достаточно очевидно: абсолютные показатели применяют при наличии эталонной, как правило, количественной шкалы. Если такую шкалу построить не удастся, то используют относительные показатели. Особенно наглядное применение относительной шкалы имеет место при использовании критерия превосходства. Например, разрабатываемая ИС превосходит по оперативности систему-прототип на 20 %. Следует иметь в виду и очевидные недостатки относительной шкалы: относительные показатели не являются достаточными для принятия решения о практической состоятельности ИС. Так, например, ИС может превосходить прототип, но при этом не удовлетворять критерию пригодности, или используемый показатель не является существенным с точки зрения практических приложений.

3. Критерии эффективности

Основные технические характеристики любой ИС $IS(Q)$ задаются векторной совокупностью ее собственных показателей $Q = (q_1, q_2, \dots, q_n)$.

Для ИС в роли таких показателей могут выступать:

- допустимый объем хранения полезной информации;
- реальный объем памяти *hardware*, занимаемый ИС;
- время доступа (время получения пользователем единицы объема востребованных данных из СХИ);
- степень безопасности ИС в отношении несанкционированного доступа;
- степень надежности сохранения информации в СХИ;
- открытость системы и т. п.

Как видно из приведенного перечня, собственные показатели эффективности ИС могут обладать различной природой: часть из них носит количественный (метрический) характер, хорошо согласующийся с традиционными методами статистического оценивания, другие имеют качественный характер и выражаются через логические переменные. В последнем случае требуются специальные методы обработки, основанные на правилах символического исчисления, неметрическом шкалировании, теории нечетких множеств и т. п. В результате возникает проблема согласования измерительных шкал (метрической, ординальной, номинальной), решение которой позволяет использовать разнородные исходные данные и довести решение задачи анализа эффективности ИС до конструктивного результата.

Как правило, технические требования к ИС задаются в виде области допустимых значений $\{Q_{IS}\}_0$, полученной в результате априорного анализа аналогов, экспериментального и/или имитационного исследования функционирования ИС и метасистемы MS , в интересах которой она была создана. Наличие априорных требований к ИС позволяет сформировать простейший *критерий пригодности ИС* в виде:

$$Q_{IS} \in \{Q_{IS}\}_0. \quad (1)$$

Как уже отмечалось, на практике в критерий (1) подставляется не значения вектора Q , а его оценка \hat{Q} , полученная по результатам измерений или априорного экспертного анализа. Следовательно, критерий (1) носит статистический характер.

Более общий подход основан на предположении о том, что множество допустимых значений технических характеристик ИС $\{Q_{IS}\}_0$ должно отвечать требованию пригодности иерархически вышестоящей метасистемы MS , в интересах которой информационная система создавалась. Предположим, что эффективность (результативность, достигнутый положительный эффект) метасистемы MS определяется скалярной величиной Eff_{MS} , зависящей от технических характеристик входящей в ее состав ИС с параметрами Q_{IS} , т.е.

$$Eff_{MS} = Eff_{MS}(Q_{IS}). \quad (2)$$

Реальная эффективность функционирования метасистемы MS зависит от множества разнообразных факторов: показателей самой метасистемы, качества ее управления, особенностей применения и т.п. Однако для решения задачи оценки эффективности ИС интерес представляет только частная зависимость (2); остальные зависимости и параметры рассматриваются как условия, фиксируемые в ограничительной или дескриптивной части решаемой задачи.

Как уже отмечалось, оценка эффективности ИС осуществляется исходя из результативности функционирования (применения) метасистемы. Оценка результативности метасистемы, рассматриваемая здесь как синоним ее эффективности, в свою очередь, определяется на основе субъективных критериев, имеющих экзогенный (уже по отношению к MS) характер. Следовательно, интегральная оценка эффективности ИС Eff_{IS} будет определяться набором *a priori* заданных критериев, позволяющих определить необходимые и достаточные условия эффективности метасистемы MS в зависимости от собственных показателей ИС.

Необходимым условием эффективности любой системы является ее *пригодность*. Однако в отличие от (1), как это только что было декларировано, она должна выражаться через терминальную эффективность метасистемы MS .

При использовании детерминированного подхода, когда в наличии имеется одна единственная реализация (или в условиях полной статистической определенности) критерий пригодности описывается соотношением

$$Eff_{MS}[Q_{IS}] > Eff_0. \quad (3)$$

Здесь Eff_0 — некоторое пороговое значение эффективности метасистемы, ниже которого MS теряет свою функциональную пригодность. На практике оценка эффективности $Eff_{MS}[Q_{IS}]$ является случайной величиной. Следовательно, для повышения достоверности критерия пригодности (3) целесообразно перейти к ее усредненному значению, полученному на основе серии из N испытаний с одинаковыми условиями их проведения:

$$\overline{Eff_{MS}(Q_{IS})} = \frac{1}{N} \sum_{i=1}^N (Eff_{MS}(Q))_i > Eff_0. \quad (4)$$

Совокупность $Q = Q_{IS}$, отвечающих условию (4), образует множество допустимых значений технических характеристик ИС $\{Q_{IS}\}_0$.

В более общем случае пригодность ИС определяется вероятностью того, что достигнутый метасистемой положительный эффект (результат) будет превышать некоторый установленный порог $Eff_{MS}^0 = Eff_0$. В этом случае критерий пригодности ИС можно представить в виде:

$$P\{Eff_{MS}[Q_{IS}] > Eff_0\} > P_0, \quad (5)$$

где P_0 — критическое значение, выбираемое исходя из требований решаемой метасистемой MS задачи.

Определение значения вероятности P_0 предполагает знание функции распределения $F(Eff_{MS}(Q))$. В некоторых случаях распределение можно задать исходя из общих соображений. Например, из условий выполнения ограничений центральной предельной теоремы можно сделать предположение о нормальном характере распределения $Eff_{MS}(Q)$. При наличии достаточного статистического материала можно построить непараметрическую оценку плотности распределения. Однако для большинства практических задач оценку функции распределения эффективности метасистемы получить не удастся и следует использовать более доступные для реализации критерии (3, 4).

Критерий пригодности является необходимым условием: его невыполнение делает бессмысленным разработку или применение соответствующей ИС. Таким образом, выполнение критерия пригодности является некоторым императивом: с его помощью можно однозначно указать на нецелесообразность разработки/применения ИС, но нельзя ответить на другой не менее важный вопрос — насколько эффективна ее реализация.

В связи с этим, как уже отмечалось в предыдущем разделе, в практике квалиметрии одновремен-

но с (3—5) используются условия достаточности — критерии превосходства или оптимальности.

Критерий превосходства позволяет оценить ИС на основе сопоставления ее параметров с параметрами системы-прототипа IS^0 :

$$Eff_{MS}[Q(IS)] > Eff_{MS}^0 \text{ для } \forall Q_{IS} \in \{Q_{IS}\}_0, \quad (6)$$

где множество допустимых значений параметров ИС $\{Q_{IS}\}_0$ формируется из условия выполнения требования пригодности MS (1).

Величина $Eff_{MS} = Eff(Q) - Eff(Q^0)$ будет отражать выигрыш, достигнутый от применения ИС IS вместо системы-прототипа IS^0 .

Применение критерия (6) требует существенных оговорок.

1. Желательно, чтобы эффективность метасистемы являлась монотонно-неубывающей функцией от параметров качества ИС Q , т. е.

$$\text{если } Q_1 \geq Q_2, \text{ то } Eff_{MS}(Q_1) \geq Eff_{MS}(Q_2). \quad (7)$$

На практике выполнения этого условия ожидать не следует. Повышение качества ИС не гарантирует повышение эффективности MS .

2. Понятия "больше" или "меньше" при сравнении векторов определяются через построенные на них скалярные меры. Сам выбор меры представляет собой субъективный экзогенный процесс и требует дополнительного обоснования.

3. Критерий превосходства в форме (6) правомерен лишь для независимых параметров q_1, q_2, \dots, q_n . В случае, когда последние являются зависимыми величинами, увеличение какого-то частного параметра q_i может привести к уменьшению другого параметра q_j , что, для нелинейного оператора $Eff_{MS}(Q)$ может привести к серьезным трудностям при формализованном синтезе ИС по критерию превосходства.

Очевидно, что критерий (6) не гарантирует, что исследуемая ИС будет иметь наибольшую эффективность. Поэтому более строгое решение может быть получено на основе критерия оптимальности, выраженного через эффективность метасистемы:

$$Eff_{MS}(Q^*) > Eff_{MS}(Q_{IS}) \text{ для } \forall Q_{IS} \in \{Q_{IS}\}_0. \quad (8)$$

При выполнении приведенных выше ограничений 1—3, в частности при выполнении условия монотонности (7), векторный критерий оптимальности можно представить в виде $Q = Q^* = \text{extr}\{Q_{IS}\}_0$, т. е. имеет наибольшее (или наименьшее) значение среди $\forall Q_{IS} \in \{Q_{IS}\}_0$.

В ряде прикладных задач, связанных с оценкой эффективности ИС, крайне плодотворной оказывается представление качественного состояния системы в виде точки m -мерного фазового пространства. При этом изменение эффективности ИС будет соответствовать перемещению фазовой точки в данном пространстве $\{Q^m\}$. Попадание вектора Q в об-

ласть $\{Q_0\}$ будет соответствовать выполнению критерия пригодности, превышение значений вектора $q_j^0 \in \{Q_0\}, j = 1, \dots, n$ — критерию превосходства и нахождение глобального максимума $q_j^*, j = 1, \dots, n$ — критерию оптимальности.

Заметим, что приведенные выше показатели эффективности относятся к детерминированной ситуации. В случае, когда имеется возможность получить избыточный объем наблюдений показателей качества, целесообразно перейти к вероятностно-статистической постановке, обеспечивающей более реалистическую оценку эффективности ИС. В частности, при наличии вероятностного распределения вектора Q , критерии пригодности, превосходства и оптимальности можно представить в виде, соответственно:

$$P\{Q \in \{Q_0\}\} > P_0; \quad (9)$$

$$P\{Q > Q^0\} > P_0; \quad (10)$$

$$P\{Q = Q^*\} > P_0. \quad (11)$$

При этом соотношение (9) выступает в качестве необходимого условия-ограничения для (10) и (11).

В более общем, не обусловленном монотонной зависимостью эффективности метасистемы от параметров используемой ИС $\{Q\}$, вероятностный подход требует знания функции распределения показателя эффективности метасистемы $F\{Eff_{MS}(Q)\}$. По аналогии с необходимым критерием (критерием пригодности) (5) можно сформулировать и достаточные критерии эффективности. В частности, эффективность ИС по критерию превосходства будет определяться вероятностью того, что достигнутый метасистемой положительный эффект (результат) $Eff_{MS}(Q(IS))$ будет превышать соответствующий эффект (полученный в тех же условиях), достигнутый при использовании ИС-прототипа $IS^0 = IS(Q^0)$ с вероятностью, не ниже заданной P_0 . В этом случае критерий превосходства ИС можно представить в виде:

$$P\{Eff_{MS}[Q_{IS}] > Eff_{MS}[Q^0]\} > P_0. \quad (12)$$

При отсутствии полных априорных знаний о распределении параметра эффективности $F\{Eff_{MS}(Q)\}$, критерий превосходства можно сформировать на основе средних значений

$$\overline{Eff_{MS}(Q_{IS})} > \overline{Eff_{MS}(Q_{IS^0})}, \text{ для } \forall Q_{IS} \in \{Q_{IS}\}_0. \quad (13)$$

Аналогичные соотношения могут быть получены для критерия оптимальности. При этом необходимо заранее определить множество, на котором осуществляется оптимизация. В частности, если осуществляется синтез или поиск оптимальной ИС на некотором множестве допустимых значений показателей качества ИС $\{Q_{IS}\}_0$, критерии оптималь-

ности, в зависимости от уровня априорной статистической определенности, могут быть, по аналогии с (12, 13), записаны в виде

$$P\{Eff_{MS}[Q_{IS}^*] > Eff_{MS}[Q_{IS}]\} > P_0, \text{ для } \forall Q_{IS} \in \{Q_{IS}\}_0,$$

где $Q^*(IS) = Q_{IS}^*$ — оптимальные значения параметров ИС.

Рассмотрим пример приложения предложенной методики к оценке эффективности различных типов ИС.

4. Оценка эффективности АСУ как информационной системы

Как уже отмечалось, выбор показателя и критерия эффективности ИС осуществляется исходя из цели, определяемой со стороны иерархически вышестоящей метасистемы MS . В случае промышленного производства роль метасистемы MS по отношению к АСУ выполняет технологический процесс (ТП) или технологический цикл предприятия. Таким образом, эффективность АСУ как информационной системы следует оценивать через эффективность ТП.

Рассмотрим формализованную постановку оценки эффективности промышленной АСУ. Предположим, что ТП (т. е. метасистема MS), управляемый АСУ, функционирует в сложной, не полностью определенной среде под воздействием последовательностей управляющих воздействий $\{u_k, k = 1, \dots, m\}$. Функциональная схема взаимодействия АСУ и метасистемы (ТП) представлена на рис. 2.

Состояние объекта управления $X_k, k = 1, \dots, N$, в общем случае, представляет собой векторный динамический стохастический процесс. Множество состояний, как правило, является континуумом, однако в процессе взаимодействия с цифровой системой мониторинга оно преобразуется в дискретную временную последовательность.

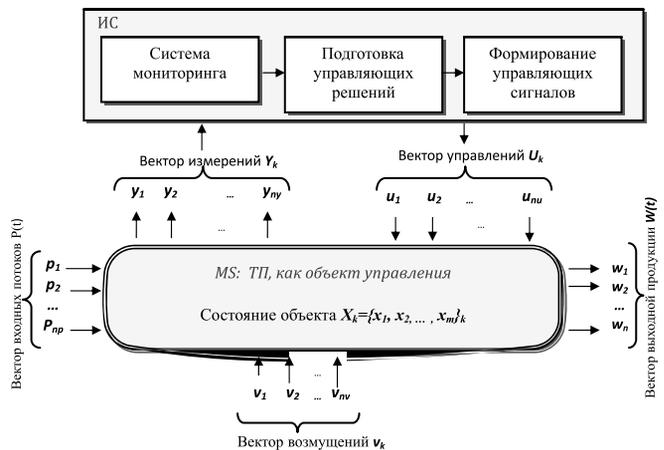


Рис. 2. Функциональная схема взаимодействия АСУ как информационной системы, и метасистемы — технологического процесса

Векторная совокупность текущего состояния ТП x_k , управляющих u_k и возмущающих v_k воздействий образует текущую ситуацию $S_k = \{x_k, u_k, v_k, k = 1, \dots, N\}$.

Предположим, что $U = \{u_k\}$ — множество допустимых векторных управлений, формируемых АСУ. Каждое $u_k \in U$ представляет собой текущее управление, сформированное на основе анализа текущей ситуации S_k . В результате мониторинга состояния ТП образуется последовательность n_v -мерных векторов наблюдения $\{Y_k = (y_1, \dots, y_{n_v})_k, k = 1, \dots, N\}$, используемая для подготовки управляющих решений. Реализация управления приводит к реакции ТП, которая будет проявляться:

- в переходе ТП из состояния x_k в состояние x_{k+1} под воздействием управляющего сигнала $u_k \in U$;
- в выработке выходных воздействий (выходной продукции) $W_{k+\tau}$, определяющей терминальное качество управления.

В конечном итоге ТП под воздействием управлений $U_N = \{u_1, \dots, u_N\}$ переходит из состояния x_1 в состояние x_N , в результате чего формируется положительный эффект \mathcal{E}_N (функциональный или экономический, т. е. соотношенный с затратами). Для одного производственного цикла эффективность управления будет оцениваться терминальным эффектом самого ТП, т. е. $Eff_{IS} = \mathcal{E}_N$. В случае, когда имеются данные по m завершенным производственным циклам, эффективность может быть уточнена

путем традиционного усреднения $Eff_{IS} = \frac{1}{m} \sum_{i=1}^m \mathcal{E}_{N_i}$

Рассмотрим задачу оценки эффективности пошагового корректирующего управления ТП. Пусть в результате теоретического анализа или моделирования установлено, что наибольшее достижимое значение положительного эффекта равно \mathcal{E}_N^* . Для достижения такого результата \mathcal{E}_N^* известными методами оптимального управления формируется оптимальный план (программа) — последовательность управлений $U_N^* = \{u_1, \dots, u_N\}^*$. При этом в соответствии с принципами системного анализа [19—23] эффект, достигаемый на каждом конкретном шаге, может не быть наибольшим: важно, чтобы максимум достигался в конце управления при $k = N$, т. е. в момент достижения терминальной цели.

Поскольку объект управления находится под воздействием большого числа не полностью определенных возмущений, его состояние отклоняется от оптимальной фазовой траектории, обеспечивающей оптимальное решение поставленной задачи. Для демпфирования возникающих отклонений ИС на основе обработки совокупности текущих наблюдений формирует последовательность корректирующих управлений $U_k = \{u_1, \dots, u_N\}_k, k = 1, \dots, N$, позволяющую поддерживать реализацию оптимального плана.

В случае, когда формируемая последовательность управления обеспечивает пригодное или оптимальное функционирование ТП, можно говорить, соответственно, о пригодности или оптимальности оцениваемой ИС. Наоборот, если низкое качество управления приводит к тому, что система приходит в состояние $x_N \neq x_N^*$, то это означает неоптимальную работу ИС, приводящую к дефициту результирующего эффекта $d\mathcal{E}_N = \mathcal{E}_N^* - \mathcal{E}_N$. При этом если $d\mathcal{E}_N$ настолько велика, что полученный эффект оказывается меньше допустимого порога \mathcal{E}_0 , то ИС оказывается непригодной для выполнения поставленной перед ней задачи.

Отсюда, в соответствии с приведенными выше критериями системной эффективности, можно получить конкретные формы оценки качества АСУ как информационной системы.

ИС является *эффективной по критерию пригодности*, если качество формируемой ею программ управления $U_N^0 = \{u_1, \dots, u_N\}_0$ обеспечивает результирующий положительный эффект \mathcal{E}_N не меньше минимально допустимого \mathcal{E}_0 (область эффективных (здесь — допустимых) программ управлений $\{U_0\} = \{u: \mathcal{E}_N \geq \mathcal{E}_0\}$).

В случае если имеется возможность набрать статистику управления, т. е. повторить процесс несколько раз, можно использовать вероятностные оценки эффективности. В этом случае АСУ, как ИС, признается эффективной по критерию пригодности, если:

- результирующий положительный эффект \mathcal{E}_N в среднем оказывается не меньше минимально допустимого \mathcal{E}_0 . Иными словами, область эффективных программ управлений $\{U_0\} = \{u: \bar{\mathcal{E}}_N \geq \mathcal{E}_0\}$;
- вероятность превышения результирующим положительным эффектом \mathcal{E}_N минимально допустимого порога \mathcal{E}_0 превышает заданное значение P_0 . Соответственно, область эффективных программ управлений равна $\{\mathcal{E}_0\} = \{\mathcal{E}: P\{\mathcal{E}_N \geq \mathcal{E}_0\} \geq \alpha_0\}$.

ИС является *эффективной по критерию превосходства*, если качество формируемых ею управляющих решений является достаточным для реализации программы управления $U_N^0 = \{u_1, \dots, u_N\}_0$, такой, что:

- результирующий положительный эффект \mathcal{E}_N не меньше эффекта \mathcal{E}^0 , обеспечиваемого применением системы-прототипа: $\mathcal{E}_N \geq \mathcal{E}^0$ (область эффективных программ управления $\{U_0\} = \{u: \mathcal{E}_N \geq \mathcal{E}^0\}$);
- результирующий положительный эффект \mathcal{E}_N в среднем оказывается не меньше эффекта \mathcal{E}^0 , обеспечиваемого применением системы-прототипа: $\bar{\mathcal{E}}_N \geq \mathcal{E}^0$ (область эффективных программ управления $\{U_0\} = \{u: \bar{\mathcal{E}}_N \geq \mathcal{E}^0\}$);

— вероятность превышения результирующим положительным эффектом \mathcal{E}_N эффекта \mathcal{E}^0 , обеспечиваемого применением системы-прототипа, пре-

вышает заданное значение α_0 (область эффективных программ управления $\{U_0\} = \{u: P[\bar{\mathcal{E}}_N \geq \mathcal{E}^0] \geq \alpha_0\}$).

АСУ как информационная система является *эффективной по критерию оптимальности*, если качество формируемых управляющих решений, предоставляемых ею, является достаточным для реализации наиболее эффективной программы управления $U_N^* = \{u_1, \dots, u_N\}^*$ такой, что:

— результирующий положительный эффект \mathcal{E}_N является оптимальным (максимальным) на классе всех допустимых управлений $\{U\}$, т. е. $\mathcal{E}_N = \mathcal{E}_N^*$ или, при недостижимости \mathcal{E}_N^* , $\delta \mathcal{E}_N = \mathcal{E}_N^* - \mathcal{E}_N = \min$ (область эффективных программ управления $\{U^*\} = \{u: \partial \mathcal{E}_N = \min\}$);

— результирующий положительный эффект $\bar{\mathcal{E}}_N$ является оптимальным (максимальным) в среднем, т. е. $\bar{\mathcal{E}}_N = \bar{\mathcal{E}}_N^*$ или $\delta \bar{\mathcal{E}}_N = \bar{\mathcal{E}}_N^* - \bar{\mathcal{E}}_N = \min$ (область эффективных программ управления $\{U^*\} = \{u: \partial \bar{\mathcal{E}}_N = \min\}$);

— с вероятностью $\alpha \geq \alpha_0$ результирующий положительный эффект \mathcal{E}_N превышает значение эффекта \mathcal{E} , обеспечиваемого любой другой допустимой программой управления (область эффективных программ управления $\{U^*\} = \{u: P[\partial \mathcal{E}_N = \min] \geq \alpha_0\}$).

Заключение

Основным выводом из предложенных материалов следует считать принципиальную возможность и необходимость осуществлять количественную оценку эффективности информационных систем и технологий через положительный эффект, достигаемый метасистемой, в интересах которой информационная система или технология созданы и применяются.

При этом и сама метасистема может не являться терминалом. Например, для АСУ ТП метасистемой является сам технологический процесс. Однако реальный эффект достигается не на выходе ТП, а на выходе всего производственного цикла. Более того, и товарная продукция не отражает конечный эффект, ее необходимо еще продать и учесть всю совокупность издержек. Поэтому выбор этапа, на котором осуществляется оценка положительного эффекта, как и выбор самого критерия эффективности, остается субъективной экзогенной задачей, формируемой на уровне здравого смысла.

Тем не менее на основе предложенной методологии может быть построена формализованная система статистических критериев, определяющих необходимые и достаточные условия эффективности ИС.

Список литературы

1. **Урсул А. Д.** Природа информации. Философский очерк. Челябинск: ЧГАКИ, 2010. 231 с. URL: http://www.inion.ru/files/File/Ursul_A_D_Priroda_informacii.pdf.
2. **Соколов А. В.** Философия информации: учеб. пособие. Челябинск: ЧГАКИ, 2011. 454 с.
3. **Колін К. К.** Философия информации: структура реальности и феномен информации. М.: Институт проблем информатики, 2016. URL: http://telecomlaw.ru/young_res/Kolin_Filosofiya_informacii.pdf
4. **Мелик-Гайказян И. В.** Информационные процессы и реальность. М.: Наука, 1998. 192 с.
5. **Юсупов Р. М., Заболотский В. П.** Концептуальные и научно-методологические основы информатизации. СПб.: Наука, 2009. 542 с.
6. **Арутюнян М., Ермошкин Н., Карминский С.** и др. Демистификация ИТ: Что на самом деле информационные технологии дают бизнесу / Под общ. ред. Н. Ермошкина. М.: Альпина Бизнес Букс, 2006. 296 с.
7. **Ермошкин Н. Н.** Шесть способов оценить пользу и эффективность информационных технологий // Банковский ри-тейл. 2006. № 4. URL: <https://www.lawmix.ru/bux/76476>
8. **Аншина М.** Оценка эффективности ИТ // Журнал школы IT-менеджмента "Системы управления бизнес-процессами". Академия народного хозяйства при Правительстве РФ. 2011. Вып. № 7. URL: <http://journal.itmane.ru/node/591>
9. **ROI 2003: Do You Have Any Faith In Your ROI Numbers?** URL: <http://www.cioinsight.com/c/a/Research/ROI-2003-Do-You-Have-Any-Faith-In-Your-ROI-Numbers#sthash.uvUbGa45.dpuf>
10. **Strassmann P. A.** The squandered computer. Evaluating the business alignment of information technologies. New Canaan: Information Economics Press, 1997. 232 p.
11. **Pisello T., Strassmann P.** IT value chain management. Maximizing the ROI from IT investment. New Canaan: Information Economics Press, 2003. 236 p.
12. **Ананьин В. И.** В поисках эффективности ИТ // Intelligent Enterprise. 2009. № 7, 8, 10. URL: <https://www.iemag.ru/analytics/detail.php?ID=18833>
13. **Скрипкин К. Г.** Экономическая эффективность информационных систем. М.: ДМК Пресс, 2003. 256 с.
14. **Карр Н.** Блеск и нищета информационных технологий. Почему ИТ не являются конкурентным преимуществом. М.: Секрет фирмы, 2005. 176 с.
15. **Новак Е. В.** Информационно-коммуникационные технологии: оценка эффективности // Информационные технологии. 2014. № 8. С. 74—80.
16. **Новак Е. В.** Как измерить эффективность информационно-коммуникационных технологий // Статистика и экономика. 2014. № 3. С. 186—191.
17. **Bresnahan T., Brynjolfsson E., Hitt L.** Information Technology, Workplace Organization and Demand for Skilled Labor: An Empirical Evidence // Quarterly Journal of Economics, 2002. Vol. 117, N. 1. P. 339—376.
18. **Эффективность инвестиций в ИТ.** Альманах лучших работ // Information Management. 2012. 196 с.
19. **Андрейчиков А. В., Андрейчикова О. Н.** Системный анализ и синтез стратегических решений в инноватике: Математические, эвристические и интеллектуальные методы системного анализа и синтеза. М.: Ленанд, 2015. 306 с.
20. **Белов П. Г.** Управление рисками, системный анализ и моделирование. Ч. 3: учебник и практикум для бакалавриата и магистратуры. Люберцы: Юрайт, 2016. 272 с.
21. **Дрогобыцкий И. Н.** Системный анализ в экономике: учебник. М.: ЮНИТИ, 2016. 423 с.
22. **Кириллов В. И.** Квалиметрия и системный анализ: учеб. пособие. М.: ИНФРА-М, 2012. 440 с.
23. **Козлов В. Н.** Системный анализ, оптимизация и принятие решений. М.: Проспект, 2016. 176 с.

R. M. Yusupov, Professor, Corr. member RAS, e-mail: yusupov@ias.spb.su,
Saint-Petersburg Institute for Informatics and Automation RAS, Saint-Petersburg,

A. A. Musaev, Professor, e-mail: amusaev@technolog.edu.ru,
Saint-Petersburg State Institute of Technology

By Estimating the Effectiveness of Information Systems. Methodological Aspects

The task of creation a methodology for evaluating the effectiveness of information systems (IS) and technology. The specificity of this problem is the service nature of the information applications, which leads to the indirect estimation of IS effectiveness through the positive terminal effect of metasystem, in which interests the information system has been created or adapted.

The brief summary of the existing approaches and techniques to estimation of information systems and technologies efficiency is provided. Ambiguity of the used efficiency determinations, and, as a result, difficulties of creation of the formalized, numerical technique of its estimation is established. In particular, the general opinion of most of authors on unfitness of general methodology of the investments analysis for estimation of information systems and technologies efficiency is given. The intangible nature of a subject of information systems applications — information is the essential reason of these difficulties. The offered methodology is based on a combination of traditional methods of probabilistic approach and the system analysis. Classification of performance indicators of information systems is given. The numerical efficiency evaluations of an information system considering the probabilistic nature of the data used for creation of these estimates are offered. The options of necessary and sufficient criteria of information systems efficiency expressed through efficiency of metasystems for the benefit of which they function are provided.

Numerical efficiency evaluations for an industrial control system, for the analytical information system oriented to tasks of decision support, and the educational information system intended for support of educational process are given as examples.

In conclusion of article connection between a information system probabilistic efficiency and the risk consisting in failure to carry out of the task set for system is specified. Features of a risk assessment and a technique its calculation on the basis of logical-probabilistic approach are specified.

Keywords: information, information systems, information technologies, efficiency, quality, methodology, probabilistic approach, system analysis

References

1. **Ursul A. D.** *Priroda informacii. Filosofskij ocherk* (The nature of the information. Philosophical Essays), Tseljabinsk, TchSACI, 2010, 231 p., URL: http://www.inion.ru/files/File/Ursul_A_D_Priroda_informacii.pdf (in Russian).
2. **Sokolov A. V.** *Filosofija informacii: uchebnoe posobie* (Philosophy of information: Textbook), Tseljabinsk, TchSACI, 2011, 454 p. (in Russian).
3. **Colin K. K.** *Filosofija informacii: struktura real'nosti i fenomen in-formacii* (Philosophy of information: the structure of reality and the phenomenon of information), Moscow: Institute of Informatics Problems, 2016, URL: http://telecomlaw.ru/young_res/Kolin_Filosofiya_informacii.pdf (in Russian).
4. **Melikk-Gaicasyan I. V.** *Informacionnye processy i real'nost'* (Information processes and Reality), Moscow, Nauka, 1998, 192 p. (in Russian).
5. **Yusupov R. M., Zabolotski V. P.** *Konceptual'nye i nauchno-metodologicheskie osnovy informatizacii* (Conceptual, scientific and methodological bases of information), S. Petersburg, Nauka, 2009, 542 p. (in Russian).
6. **Arutyunyan M., Ermoshkin N., Karminski S.** et al. *Chto na samom dele informacionnye tehnologii dajut biznesu* (Demystifying IT: What actually give the information technology business), Moscow, Alpina Business Books, 2006, 296 p. (in Russian).
7. **Ermoshkin N. N.** *Shest' sposobov ocenit' pol'zu i jeffektivnost' informacionnyh tehnologij* (Six ways to evaluate the usefulness and efficiency of information technology), *Bankovskij ritejl*, 2006, no. 4, URL: <https://www.lawmix.ru/bux/76476> (in Russian).
8. **Anshina M.** *Ocenka jeffektivnosti IT* (Assessment of IT effectiveness), *Zhurnal shkoly IT-menedzhmenta "Sistemy upravlenija biznes-processami"*. Akademija narodnogo hozjajstva pri Pravitel'stve RF, 2011, no. 7, URL: <http://journal.itmane.ru/node/591> (in Russian).
9. **ROI 2003: Do You Have Any Faith In Your ROI Numbers?** URL: <http://www.ciainsight.com/c/a/Research/ROI-2003-Do-You-Have-Any-Faith-In-Your-ROI-Numbers#sthash.uvUbGa45.dpuf>
10. **Strassmann P. A.** *The squandered computer. Evaluating the business alignment of information technologies*, New Canaan, Information Economics Press, 1997, 232 p.
11. **Pisello T., Strassmann P.** *IT value chain management. Maximizing the ROI from IT investment*, New Canaan, Information Economics Press, 2003, 236 p.
12. **Ananjin V. I.** *V poiskah jeffektivnosti IT* (Finding IT efficiency), *Intelligent Enterprise*, 2009, no. 7, 8, 10, URL: <https://www.iemag.ru/analitics/detail.php?ID=18833> (in Russian).
13. **Scripkin K. G.** *Jekonomicheskaja jeffektivnost' informacionnyh sistem* (The cost-effectiveness of information systems), Moscow, DMK Press, 2003, 256 p. (in Russian).
14. **Carr N.** *Blesk i nishheta informacionnyh tehnologij. Pochemu IT ne javlja-jutsja konkurentnym preimushhestvom*. (Shine and poverty of information technology. Why IT is not a competitive advantage), Moscow, Sekret firmy, 2005, 176 p. (in Russian).
15. **Novak E. V.** *Informacionno-kommunikacionnye tehnologii: ocenka jeffektivnosti* (Information and communication technologies: evaluation of the effectiveness), *Information Technology*, 2014, no. 8, pp. 74–80.
16. **Novak E. V.** *Kak izmerit' jeffektivnost' informacionno-kommunikacionnyh tehnologij* (How to measure the effectiveness of information and communication technologies), *Statistika i jekonomika*, 2014, no. 3, pp. 186–191.
17. **Bresnahan T., Brynjolfsson E., Hitt L.** *Information Technology, Workplace Organization and Demand for Skilled Labor: An Empirical Evidence*, *Quarterly Journal of Economics*, 2002, vol. 117, no. 1, pp. 339–376.
18. **Jefferktivnost' investicij v IT. Al'manah luchshih rabot** (The effectiveness of IT investments. Almanac best works), *Information Management*, 2012, 196 p.
19. **Andreychikov A. V., Andreychikova O. N.** *Sistemnyj analiz i sintez strategicheskikh reshenij v innovatike: Matematicheskie, jevristicheskie i intel-lektual'nye metody sistemnogo analiza i sinteza*. (System analysis and synthesis of strategic decisions in Innovation: Mathematical, heuristic and intelligent methods of system analysis and synthesis), Moscow, Lenand, 2015, 306 p.
20. **Belov P. G.** *Upravlenie riskami, sistemnyj analiz i modelirovanie*. Chast' 3: uchebnik i praktikum dlja bakalavriata i magistratury (Risk management, systems analysis and modeling. Part 3: Tutorial and workshop for undergraduate and graduate), Lyuberci, Yurait, 2016, 272 p. (in Russian).
21. **Drogobicki I. N.** *Sistemnyj analiz v jekonomike: uchebnik* (System analysis in economics: Textbook), Moscow, UNITI, 2016, 423 p. (in Russian).
22. **Kirillov V. I.** *Kvalimetrija i sistemnyj analiz: Uchebnoe posobie* (Qualimetry and Systems Analysis: A Tutorial), Moscow, INFRA-M, 2012, 440 p. (in Russian).
23. **Koslov V. N.** *Sistemnyj analiz, optimizacija i prinjatje reshenij* (System analysis, optimization and decision-making), Moscow, Prospect, 2016, 176 p. (in Russian).

УДК 519.16, 51-76

М. В. Ульянов, д-р техн. наук, проф., вед. науч. сотр., проф., e-mail: muljanov@mail.ru,
Институт проблем управления им. В. А. Трапезникова РАН,
Московский государственный университет им. М. В. Ломоносова,

Ю. Г. Сметанин, д-р физ.-мат. наук, доц., гл. науч. сотр., проф., e-mail: ysmetanin@rambler.ru,
Федеральный исследовательский центр "Информатика и управление" РАН,
Федеральное государственное образовательное учреждение высшего профессионального образования
"Московский физико-технический институт (государственный университет)"

Об одном характеристическом функционале слов над конечным алфавитом

Исходными объектами данной статьи являются конечные слова над некоторым конечным алфавитом. Эти слова представляют собой символьные коды исследуемых объектов и процессов, которые и являются предметом последующего анализа. Предложена формализация функции энтропии слов на основе энтропии дискретных распределений. На основе анализа множества всех слов фиксированной длины над заданным алфавитом введено понятие мажоранты функции энтропии. Функция энтропии слов и мажоранта являются основой для предлагаемых в статье двух форм характеристического функционала, дающих количественную оценку близости исследуемого слова конечной длины к подслову такой же длины, случайно выбранному из случайного бесконечного слова над этим алфавитом. Предложенные формы функционала могут быть использованы при решении различных задач анализа данных, в том числе задач кластеризации и распознавания.

Ключевые слова: информационная энтропия, слова над конечным алфавитом, функция энтропии слов, мажоранта функции энтропии, характеристические функционалы

Введение

В данной статье предполагается, что анализируемая информация представлена в виде символьных кодов, т. е. образы исследуемых объектов или процессов кодированы словами над некоторым конечным алфавитом. Достоинством такого представления, возникающего при числовом кодировании при решении задач классификации, является отсутствие ложных связей и отношений между объектами. Такие символьные представления получили широкое распространение в различных задачах, от комбинаторной теории групп и символической динамики [1] до различных прикладных задач анализа информации.

Здесь будет рассмотрен случай достаточно длинных, но конечных слов, возникающих в задачах исследования объектов и процессов, заданных символьным кодированием, в частности, в задачах классификации и кластерного анализа [2]. В этих задачах возникает естественный вопрос о характеристизации классов (кластеров). При этом требуется выполнение некоторых естественных требований. Первое из них — обеспечение большей близости объектов из одного класса по сравнению с объек-

тами из других классов, т. е. выбор адекватного пространства кластеризации и метрики. Второе требование связано с обеспечением независимости от конкретного выбора кода: желательно, чтобы при выборе другого способа кодирования классы не изменялись. Использование характеристических функционалов на множестве слов является одним из способов обеспечения выполнения этих требований.

Характерным примером использования таких функционалов является энтропия пространства сдвигов в символической динамике [1]. Предположим, что заданы два различных кода, и в каждом задано пространство сдвигов, т. е. некоторое подмножество конечных слов, определяемое языком (или, что то же самое, набором запрещенных подслов). Требуется ответить на вопрос, не являются ли эти пространства по сути одинаковыми в том смысле, что переводятся одно в другое простым перекодированием (кодом скользящего блока). Доказано [1], что необходимым условием существования такого перекодирования является равенство энтропий этих пространств сдвигов.

К настоящему времени предложено много различных характеристических функционалов, адаптированных к различным типам данных и задач.

Так, энтропия Реньи используется в анализе случайности и неопределенности процессов; энтропия Колмогорова — Синяя — в исследовании динамических систем; взаимная информация характеризует количество информации, которое можно получить об одной случайной переменной с помощью анализа другой случайной переменной; негэнтропия дает оценку расстояния распределения от нормального; квантовая относительная энтропия есть мера неразличимости двух квантовых состояний и т. д.

В задачах классификации и кластерного анализа полезными являются также и пиковые характеристики [2]: максимальные значения некоторого характеристического функционала и значения аргументов, при которых они достигаются, дают возможность выбирать наилучшие признаки и добиваться высокой точности и робастности решений. Эти характеристики оказываются полезными также в решении важной задачи идентификации слов, близких к случайным. Такая идентификация позволяет в задачах прогнозирования с представлением в виде символьных кодов выявлять плохо прогнозируемые процессы.

Необходимо отметить, что характеристические функционалы, как правило, не дают полного ответа на поставленные вопросы. Так, в приведенном выше примере об энтропии пространства сдвигов получено только необходимое условие сводимости; достаточные условия при этом не известны [1]. Если удастся определить несколько независимых функционалов с аналогичными свойствами, ситуация станет намного яснее: неравенство значений хотя бы одного из них означает различие пространств сдвигов, а равенство всех значений делает весьма правдоподобным предположение о совпадении.

Сходная ситуация возникает во многих задачах анализа данных. Отсюда вытекает важность задачи расширения множества характеристических функционалов. При анализе функционалов указанного вида полезной оказывается также информация об элементах, на которых достигаются экстремумы функционалов. Такие элементы могут играть роль своеобразных эталонов для сравнения с остальными элементами, выделения классов элементов и т. д.

Объектом исследования в данной статье являются конечные слова над конечным алфавитом, порожденные символьным кодированием образов исследуемых объектов или процессов. *Предмет исследования* — характеристические функционалы на множестве слов, построенные на основе функции энтропии.

Постановка задачи — предложить для исследуемого слова конечной длины над конечным алфавитом характеристический функционал, отражающий близость этого слова (в некотором метрическом пространстве) к случайно выбранному (из случайного бесконечного слова) подслову такой же длины.

При исследовании разнообразных объектов или процессов определенный интерес представляет подход к анализу информации, при котором образы исследуемых объектов или процессов представляются словами над некоторым конечным алфавитом. Преобразование информации в этом случае происходит с использованием символьного кодирования. Данный подход не требует числового кодирования образов объектов, зачастую приводящего к появлению ложной информации, вызванной исключительно особенностями использованного кода. Вместо этого предлагается проводить анализ символьных строк в целях выявления их эффективных характеристик в аспекте решаемых задач, например, в целях их классификации или распознавания. Заметим, что в пользу символьного кодирования говорит и тот факт, что переход от числовых значений к символам, кодирующим числовой сегмент, приводит к меньшей чувствительности полученных слов по отношению к шумам, сопутствующим измерительной аппаратуре.

Изучением таких символьных представлений занимается раздел современной дискретной математики — комбинаторика слов [3]. Отличительной особенностью этого подхода является тот факт, что аппарат комбинаторики слов применяется к символьным кодам объектов или процессов из разнообразных областей исследований. В связи с этим символьное кодирование можно рассматривать как средство унификации описаний разнородных объектов и процессов, что позволяет в дальнейшем выявлять их общности или общие особенности.

Результатом символьного кодирования является слово в выбранном алфавите, которое затем изучается методами комбинаторики слов. Выбор алфавита и способа символьного кодирования, очевидно, обуславливает и качественные результаты последующего анализа. В ряде случаев сама специфика проблемной области четко обуславливает алфавит и код. Например, в молекулярной биологии и биоинформатике при исследовании геномов очевидным является четырехсимвольный алфавит нуклеотидов ДНК — $\{A, G, T, C\}$.

В случае, когда исходными являются числовые характеристики, символьное кодирование требует определенной аккуратности. Мы продемонстрируем это на примере символьного кодирования временных рядов. Пусть текущим объектом исследования является временной ряд произвольной природы

$$T = \{(f_i, t_i), f_i \in R^1, i = 1, \dots, n\},$$

где f_i — значение характеристики наблюдаемого процесса в момент t_i ; n — число наблюдений (отсчетов), а предметом исследования является построение символьного кода для значений временного

ряда T . При символьном кодировании временных рядов возникает требование универсальности метода кодирования, поскольку различные временные ряды имеют различную точность измерений (число значащих цифр в значениях f_i) и различный диапазон значений. Возможное решение может состоять в едином масштабировании значений наблюдаемой функции процесса и построении на этой основе строки символов, отражающей динамику числовых значений исследуемого ряда [4].

В целях такого масштабирования вводится разбиение y_i , $i = 1, \dots, m$, диапазона размаха варьирования значений f_i , где $y_1 = \min_{k=1, n} f_k$, $y_m = \max_{k=1, n} f_k$,
 $y_{i+1} = y_i + \Delta y_i$, $i = \overline{1, m-1}$.

Поскольку значения временного ряда могут попасть в точки разбиения, рассматриваются множества $[y_i, y_{i+1}) = \{y | y_i \leq y < y_{i+1}\}$, которые далее будем называть полусегментами, при этом определение числа (m) и границ полусегментов $[y_i, y_{i+1})$ является самостоятельной интересной задачей. Один из вариантов ее решения — применение бикритериального метода построения гистограмм [5], при этом число полученных полусегментов гистограммы определяет мощность алфавита, а сами полусегменты кодируются символами выбранного алфавита. Подробное изложение этого метода по отношению к символьному кодированию приведено в работе [6]. Выбор символов алфавита, по сути, не принципиален, иллюстративно далее будут использоваться строчные символы латинского алфавита. Каждому полусегменту ставится в соответствие уникальный символ алфавита, а числовое значение ряда кодируется символом полусегмента, в котором оно находится. В результате мы получаем представление временного ряда в виде строки символов. Для временного ряда, содержащего n наблюдений, мы получаем его представление в виде строки из n символов над алфавитом Σ .

Отметим еще одно преимущество подхода символьного кодирования. За редким исключением, значения в отсчетах временных рядов не являются точными. Одним из таких исключений являются, например, ряды курсов валют. Для значений, имеющих погрешность измерений, в математической статистике принято строить доверительные интервалы. Используемый бикритериальный метод построения гистограмм как раз и определяет ширину полусегмента гистограммы, а следовательно, и "ширину" полосы значений для кодирующего этот полусегмент символа, на основе доверительной вероятности для среднего значения [5]. Таким образом, подход символьного кодирования более достоверно отражает исследуемый процесс с точки зрения математической статистики, нивелируя возможные шумы наблюдений.

Функция энтропии конечных слов

Реализация поставленной задачи представляется нам в виде обобщения функции энтропии конечных слов, детальную формализацию которой мы даем ниже. Отметим, что в символической динамике функции энтропии рассматриваются для слов бесконечной длины [1], но для конечного случая подход, предусматривающий переход к пределу в бесконечности, неприемлем.

Терминология и обозначения. Далее будем использовать терминологию и применять обозначения как общепринятые в комбинаторике слов и символической динамике [1, 3], так и специальные авторские обозначения, связанные с особенностью данной задачи. Введем следующие обозначения:

$\Sigma = \{s_1, s_2, \dots, s_m\}$ — конечный алфавит, s — произвольный символ алфавита;

Σ^k — k -я декартова степень множества Σ ;

$c \in \Sigma^k$ — k -элементный кортеж, $c = (s^{(1)}, s^{(2)}, \dots, s^{(k)})$, $s^{(l)} \in \Sigma$;

w — слово (над алфавитом Σ) — последовательность символов алфавита;

$WD(c) = w$ — оператор порождения слова w из кортежа c путем последовательной, в порядке кортежа, конкатенации символов

$$WD((s^{(1)}, s^{(2)}, \dots, s^{(k)})) = s^{(1)} s^{(2)} \dots s^{(k)};$$

$|w| = k$ — длина слова, понимаемая как мощность порождающего кортежа;

$L(\cdot)$ — оператор порождения множества слов, действующий на множество кортежей посредством оператора $WD(\cdot)$. Пусть $C \subseteq \Sigma^*$ — множество кортежей, W — множество слов, тогда

$$L(C) = W = \{w | \forall c \in C w = WD(c)\}.$$

Будем говорить, что оператор $L(\cdot)$ порождает некоторый язык L над алфавитом Σ , например, L_k — язык всех слов длины k над алфавитом Σ ,

$$L_k = L(\Sigma^k) = \{w | |w| = k\};$$

$SW(w, i, l)$ — оператор выделения под слова v длины l в слове w , начиная с символа в позиции i . Пусть $|w| = k$, тогда оператор определен при $i + l - 1 \leq k$:

$$SW(s_1 s_2 \dots s_k, i, l) = v = s_i s_{i+1} \dots s_{i+l-1};$$

$SH1(w, k)$ — оператор сдвига 1, действующий на слово w окном ширины k . Определенный при $|w| > k$ оператор порождает (в общем случае) мультимножество подслов длины k с мощностью $|w| - k + 1$, выполняя сдвиг на единицу окна ширины k по слову w , начиная с крайней левой позиции:

$$SH1(w, k) = \{v_i | v_i = SW(w, i, k), i = \overline{1, |w| - k + 1}\},$$

где подслова v_i могут, очевидно, породить мультимножество

$$SH1(w, k) = \{v_i^{(c_i)} \mid i = \overline{1, l}\}, \quad \sum_{i=1}^l c_i = |w| - k + 1.$$

Например, для алфавита $\Sigma = \{a, b\}$ и слова $bbababa$ при окне ширины 4 имеем:

$$SH1(bbababa, 4) = \{bbab, baba, abab, baba\} = \{bbab^{(1)}, baba^{(2)}, abab^{(1)}\}.$$

Энтропия дискретных распределений с конечным носителем. Рассмотрим классическую вероятностную модель с конечным носителем $M = \langle \Omega, P(\cdot) \rangle$, в которой вероятностное пространство конечно — $\Omega = \{\omega_1, \omega_2, \dots, \omega_k\}$, а вероятностная мера $P(\cdot)$ определяет вероятности случайных событий ω_i — $P(\omega_i) = p_i$. Тогда по определению [7] информационная энтропия такого дискретного распределения определяется в виде

$$H_P \stackrel{\text{def}}{=} - \sum_{i=1}^k p_i \log p_i. \quad (1)$$

Индекс P у H_P подчеркивает, что энтропия порождена вероятностной мерой $P(\cdot)$. Заметим, что энтропия инвариантна по отношению к элементам вероятностного пространства ω_i . По сути, H_P есть функционал, действующий в классе дискретных распределений с конечным носителем. Важно, что для этого класса распределений максимум энтропии достигается для равномерного распределения [8], т. е. когда $\forall i P(\omega_i) = 1/|\Omega| = 1/k$, что характеризует наибольшую непредсказуемость событий в такой вероятностной модели. Выбрав в качестве основания логарифма мощность вероятностного пространства, мы можем нормировать максимальное значение энтропии (1) к единице, т. е. при $P(\omega_i) = 1/|\Omega| = 1/k$ имеем

$$H_P = - \sum_{i=1}^k p_i \log_{|\Omega|} p_i = - \sum_{i=1}^k \frac{1}{k} \log_k \frac{1}{k} = 1.$$

Отметим, что при любом другом распределении, отличном от равномерного, значение $0 \leq H_P < 1$, а для вырожденного распределения $\exists \omega_i: P(\omega_i) = 1, \forall \omega_j: j \neq i P(\omega_j) = 0$ энтропия равна нулю.

Функция энтропии слова. Предлагается выполнить построение этой функции в два этапа — на первом этапе вычисляется значение функции энтропии для слова w по подсловам фиксированной длины k , а на втором этапе на этой основе вычисляется функция энтропии слова w для всех возможных длин подслов.

На первом этапе мы фиксируем длину подслова $k, 1 \leq k \leq n$. Пусть $U = L(\Sigma^k) = L_k$ — множество всех слов длины k над алфавитом Σ . Введем в рассмотрение формальное мультимножество

$$\tilde{U} = \{u_i^{(0)} \mid i = \overline{1, |\Sigma|^k}\},$$

элементы которого (все возможные слова длины k над алфавитом Σ) имеют нулевую кратность. Далее применим к исследуемому слову w оператор $SH1(w, k)$ и получим мультимножество \tilde{V} :

$$\tilde{V} = SH1(w, k) = \{v_i^{(c_i)} \mid i = \overline{1, l}\}.$$

Пусть m есть число позиций окна ширины k на слове w , и $|w| = n$, тогда

$$\sum_{i=1}^l c_i = |w| - k + 1 = n - k + 1 = m.$$

Построим объединенное мультимножество $\tilde{V} \cup \tilde{U}$. Поскольку \tilde{U} содержит все возможные слова длины k , то $\tilde{V} \cup \tilde{U}$ не будет содержать новых по отношению к множеству \tilde{U} элементов. Тем самым объединение мультимножеств приведет только к изменению кратностей некоторых, или быть может всех, элементов из \tilde{U} . На этой основе построим вероятностную модель

$$M_k = \langle \Omega_k = \tilde{V} \cup \tilde{U}, P_k(\cdot) \rangle, \quad (2)$$

где мощность $\Omega_k = |\Sigma|^k$, а вероятностная мера отражает частоту появления различных слов длины k в слове w и вычисляется на основе кратности элементов мультимножества \tilde{V} :

$$P_k(\cdot): \begin{cases} p_i = p(\omega_i) = \frac{c_i}{m}, \omega_i \in \tilde{V}; \\ p_i = p(\omega_i) = 0, \omega_i \in \tilde{U} \setminus \tilde{V}. \end{cases} \quad (3)$$

Для полученной вероятностной модели с вероятностной мерой $P_k(\cdot)$ мы определяем нормированную энтропию, используя $\Omega_k = |\Sigma|^k$ в качестве основания логарифма в (1):

$$H_{P_k} = - \sum_{i=1}^{|\Sigma|^k} p_i \log_{|\Sigma|^k} p_i = - \sum_{i=1}^l \left(\frac{c_i}{m}\right) \log_{|\Sigma|^k} \left(\frac{c_i}{m}\right). \quad (4)$$

Отметим, что значение $H_{P_k} = 0$ при $k \neq n$ означает, что все подслова, порожденные оператором $SH1(w, k)$, одинаковы и, следовательно, состоят из одного и того же символа — мы констатируем фундаментальное отсутствие разнообразия в исследуемом слове w . При $k = n$ мы получаем одно подслово, совпадающее с w , и $H_{P_n} = 0$. Просто показать, что значение $H_{P_n} = 1$ может быть получено только в случае совпадения множеств \tilde{V} и \tilde{U} , при этом все слова в \tilde{U} имеют одинаковую кратность, т. е. при равночастотности и полноте всех возможных подслов длины k в исследуемом слове w .

Построение функции энтропии слова. На втором этапе мы используем естественное расширение энтропии (4) путем введения функции $H(w, k) = H_{P_k}$, аргументами которой является слово w и длина

подслова k , с областью определения $1 \leq k \leq n$. Значения функции $H(w, k)$ при фиксированном значении аргумента k вычисляются по формуле (4) на основе вероятностной модели (3), полученной в результате применения оператора $SH1(w, k)$ к исходному слову w . Последовательный перебор значений ширины окна k от 1 до n и дает искомые значения $H(w, k)$. Таким образом, функция $H(w, k)$ для фиксированного слова w является действительно-значной функцией ограниченного целочисленного аргумента k .

В работе [4] авторы показали, что в области значений аргумента k от $\log_{|\Sigma|} n$ до \sqrt{n} имеет место соотношение $H(w, k+1) \leq H(w, k)$, и на этом сегменте $H(w, k)$ является монотонно убывающей функцией. В целом на сегменте значений аргумента от 1 до n можно говорить, что $H(w, k)$ является функцией, "убывающей по совокупности".

Модельный пример для функции энтропии слов

В качестве модельных примеров приведем (табл. 1 и 2) значения функции $H(w, k)$, вычисленные по формулам (3) и (4) для двух слов длины 10 в алфавите $\Sigma = \{a, b\}$, функция энтропии которых обладает различным качественным поведением. Первое из них — периодическое слово $w_1 = (ab)_5$, второе — слово $w_2 = (aaabbbabaa)$ построено с использованием предложенного ранее авторами метода реконструкции слов по подсловам фиксированной длины [9] и порождает в окне длины 3 при сдвиге 1 все возможные восемь подслов с единичной встречаемостью в алфавите $\Sigma = \{a, b\}$, доставляя тем самым значение $H(w_2, 3) = 1$. Отметим, что для слова w_2 функция $H(w_2, k)$ не является монотонно убывающей при начальных значениях аргумента.

Соответствующие графики приведены на рис. 1 и 2, мы показываем функции $H(w, k)$ как кусочно-линейные для наглядности отображения тенденций.

Укажем на то, что $H(w_1, 1) = 1$, поскольку символы алфавита равночастотны в слове $w_1 = (ab)_5$, а резкое падение энтропии в точке $k = 2$ — $H(w_1, 2) = 0,496$ отражает потерю разнообразия слов в окне длины 2, поскольку мы наблюдаем только два подслова (ab) и (ba) , а два других возможных подслова длины 2 — (aa) и (bb) вообще не встречаются в w_1 .

Значения функции энтропии для слова $w_1 = (ab)_5$

k	1	2	3	4	5	6	7	8	9	10
$H(k)$	1,000	0,496	0,333	0,246	0,200	0,162	0,143	0,115	0,111	0,000

Таблица 1

Значения функции энтропии для слова $w_2 = (aaabbbabaa)$

k	1	2	3	4	5	6	7	8	9	10
$H(k)$	0,971	0,987	1,000	0,702	0,517	0,387	0,286	0,198	0,111	0,000

Таблица 2

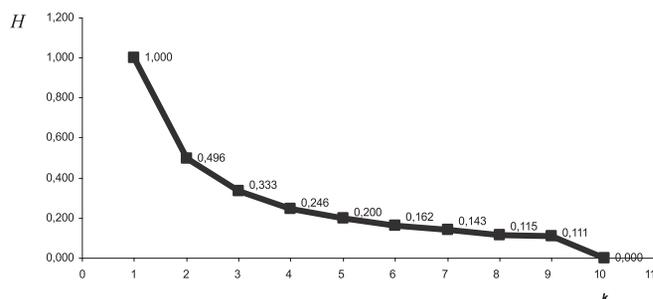


Рис. 1. График функции энтропии $H(w_1, k)$ для модельного слова $w_1 = (ab)_5$

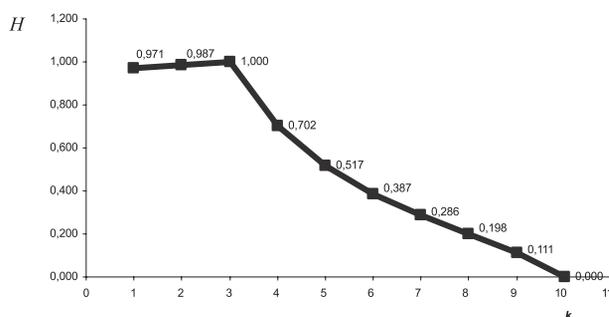


Рис. 2. График функции энтропии $H(w_2, k)$ для модельного слова $w_2 = (aaabbbabaa)$

При этом функция $H(w_1, k)$ — монотонно убывающая на всей области определения.

Для этого слова максимум энтропии приходится на окно длины 3, меньшие единицы значения для аргумента $k = 1$ и $k = 2$ объясняются отсутствием равночастотности символов и подслов длины 2 в исследуемом слове. Потеря разнообразия происходит при $k = 4$, ниже мы объясним этот факт.

Мажоранта функции энтропии для слов фиксированной длины

При изучении качественного поведения функции $H(w, k)$ возникает несколько вопросов, из которых, по нашему мнению, наибольший интерес представляют следующие два:

1. Как долго при начальных значениях аргумента k функция энтропии может принимать значения, близкие или равные единице? Иначе — когда

происходит потеря разнообразия подслов в увеличивающемся окне сдвига один?

2. Какие максимальные значения может принимать функция $H(w, k)$ при фиксированной длине слова для различных значений аргумента k ?

Порог потери разнообразия. Ответ на первый вопрос можно получить на основе сопоставления числа возможных позиций окна ширины k в слове w и числа возможных подслов длины k в данном алфавите Σ .

Для дальнейшего изложения введем следующее обозначение:

$$X = \{x_i^*\} = \underset{x \in G}{\text{Arg zero } h(x)},$$

где X — множество значений аргумента, при которых исследуемая функция действительного аргумента $h(x)$ обращается в ноль на области $G \subseteq R^1$, где R — множество действительных чисел. Формально множество X может быть пустым множеством (например, для $h(x) = x^2 + 1$, $G = R^1$). Обозначение допускает очевидное расширение на множество комплексных чисел.

Пусть $|w| = n$ при данном алфавите Σ . Рассмотрим функцию действительного аргумента $h(x) = n - x + 1 - |\Sigma|^x$ на сегменте $x \in [1, n]$. Поскольку на этом сегменте производная $h'(x) = -1 - |\Sigma|^x \ln |\Sigma|$ отрицательна, то функция всюду убывает на $[1, n]$. При выполнении условия $n > |\Sigma|$ значение $h(1) > 0$, следовательно, $h(x)$ имеет единственный ноль на сегменте $[1, n]$ (т. е. $|X| = 1$), который мы обозначим как

$$x^* = \underset{x \in [1, n]}{\text{arg zero } h(x)}. \quad (5)$$

Тогда ответ на первый вопрос сводится к рассмотрению следующих двух случаев.

Случай 1. Решение уравнения $h(x) = 0$ — целочисленное, $x^* \in N$, где N — множество натуральных чисел. Обозначим $k^* = x^*$, тогда $h(k^*) = n - k^* + 1 - |\Sigma|^{k^*} = 0 \Rightarrow n - k^* + 1 = |\Sigma|^{k^*}$. При этом если для исследуемого слова w наблюдаемые в окне ширины k^* подслово совпадают со всеми возможными словами языка L_{k^*} , то значение $H(w, k^*) = 1$. При увеличении ширины окна на единицу $k = k^* + 1$ число возможных позиций окна уменьшается на единицу, а число всевозможных слов в алфавите Σ увеличивается в $|\Sigma|$ раз. Таким образом, мы констатируем потерю возможного разнообразия в окне ширины $k^* + 1$. Именно этот случай мы и наблюдаем для слова $w_2 = (aaabbbabaa)$ длины $n = 10$ в алфавите мощности 2 в окне ширины 4 ($k^* = 3$ и $10 - 3 + 1 = 2^3$). При этом $H(w_2, 3) = 1$, а $H(w_2, 4) = 0,702$. Оценка для k^* может быть получена в предположении, что длина слова значительно превышает ширину окна, т. е. при условии $n \gg k$, тогда значение $k^* \approx \lfloor \log_{|\Sigma|} n \rfloor$.

Случай 2. Уравнение $h(x) = 0$ имеет решение в действительных числах $x^* \in R^1 \setminus N$. В этом случае последнее значение ширины окна, при котором воз-

можно наблюдать полное разнообразие подслов — $k^* = \lfloor x^* \rfloor$. Однако это разнообразие не гарантирует, что $H(w, k^*) = 1$, поскольку при этом $n - k^* + 1 > |\Sigma|^{k^*}$, что может быть препятствием для получения равночастотности. Например, при $|\Sigma| = 2$ и $n = 14$, решая соответствующее трансцендентное уравнение, получаем $x^* \approx 3,52$ и значение $k^* = \lfloor 3,52 \rfloor = 3$. Тем самым мы получаем 12 позиций окна ширины 3 при 8 возможных подслогах, что приводит к невозможности получения равных наблюдаемых частот, и, следовательно, $H(w, 3) < 1$ для любого слова w , имеющего длину 14 в бинарном алфавите. Можно показать, что и в общем случае при $x^* \in R^1 \setminus N$ значение $H(w, k^*) < 1$. Вместе с тем, при $k^* = \lfloor x^* \rfloor + 1$ ситуация меняется, что связано с увеличением в $|\Sigma|$ раз полного разнообразия подслов (в нашем примере 11 позиций окна при 16 возможных подслогах) и $H(w, k^* + 1) < H(w, k^*)$.

Таким образом, в общем случае последнее значение k , при котором функция $H(w, k)$ может иметь значения, близкие или равные единице, определяется значением $k^* = \lfloor x^* \rfloor$, где

$$x^* = \underset{x \in [1, n]}{\text{arg zero } h(x)}, \text{ а } h(x) = n - x + 1 - |\Sigma|^x. \quad (6)$$

Мажоранта функции энтропии слов. Ответ на вопрос: "Какие максимальные значения может принимать функция $H(w, k)$ при фиксированной длине слова n для различных значений аргумента k ?" можно получить на основе следующих рассуждений. Зафиксируем длину слова n и рассмотрим язык L_n — язык всех слов длины n над алфавитом Σ . Для конкретной ширины окна k оператора сдвига 1 (т. е. для аргумента k функции $H(w, k)$), в силу конечности множества L_n , существует, по крайней мере, одно слово, доставляющее максимум функции энтропии. Тем самым, мы переходим от функции $H(w, k)$ к функции $\tilde{H}(n, k)$, определяемой на всем языке L_n следующим образом:

$$n = \text{const}, \forall k = \overline{1, n}, \tilde{H}(n, k) \stackrel{\text{def}}{=} \max_{w \in L_n} H(w, k). \quad (7)$$

Будем называть функцию $\tilde{H}(n, k)$ мажорантой функции энтропии слов, поскольку по определению (7) $\forall w \in L_n \forall k = \overline{1, n}, H(w, k) \leq \tilde{H}(n, k)$. Заметим, что определение (7) не предусматривает наличие какого-то конкретного слова из L_n , обладающего мажорирующей энтропией на всей области определения, функция $\tilde{H}(n, k)$ получает свои значения для различных аргументов k , вполне вероятно, от различных слов из L_n , что и приводит к введению обобщающего (по языку L_n) первого аргумента этой функции в виде длины слова n .

Приведем метод вычисления значений функции $\tilde{H}(n, k)$ при фиксированной длине слова n , не связанный напрямую с определением (7). Рассмотрим два сегмента значений аргумента, определяемые порогом потери разнообразия $k^* = \lfloor x^* \rfloor$ (см. фор-

Значения мажоранты функции энтропии слов $\tilde{H}(10, k)$

k	1	2	3	4	5	6	7	8	9	10
$H(k)$	1,000	0,987	1,000	0,702	0,517	0,387	0,286	0,198	0,111	0,000

мулу (6)), — сегмент $[1, k^*]$ и сегмент $[k^* + 1, n]$. Для каждого из них определим наибольшие возможные значения энтропии следующим образом.

1. Для сегмента $[1, k^*]$ — при фиксированном k из этого сегмента мы наблюдаем $m = n - k + 1$ подслов в окне сдвига 1, имея $N = |\Sigma|^k$ возможных подслов в данном алфавите. В данном сегменте по построению выполняется неравенство $m \geq N$. Тогда каждому подслову длины k первоначально назначается частота, равная $m \text{ div } N$, кроме того, остаток $m \text{ mod } N$ распределяется по единице между любыми (в силу свойств энтропии) $m \text{ mod } N$ подсловами, максимизируя тем самым значение функции энтропии.

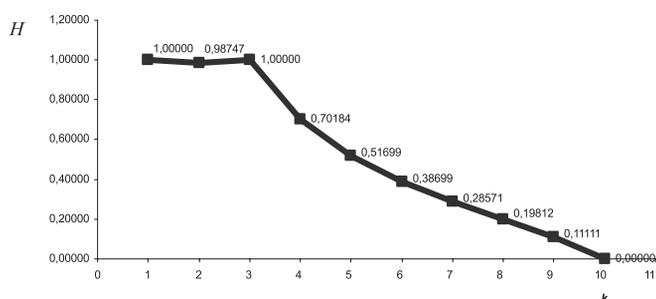
2. Для сегмента $[k^* + 1, n]$ — по построению мы фиксируем потерю разнообразия в любом окне ширины k из этого сегмента. Между значениями m и N выполняется неравенство $m < N$, и мы предполагаем, что существует слово, доставляющее единичную встречаемость всех подслов в окне ширины k оператора сдвига 1. Тогда на основе формулы (4) имеем:

$$\begin{aligned} \tilde{H}(n, k) &= - \sum_{i=1}^m \left(\frac{1}{m}\right) \log_{|\Sigma|^k} \left(\frac{1}{m}\right) = - \log_{|\Sigma|^k} \left(\frac{1}{m}\right) = \\ &= \log_{|\Sigma|^k} (n - k + 1), \end{aligned} \quad (8)$$

при этом очевидно, что $\tilde{H}(n, n) = 0$.

Приведем пример вычисления значений функции $\tilde{H}(n, k)$ при $n = 10$ в соответствии с указанным методом для бинарного алфавита. При $n = 10$ значение $k^* = 3$. Значения $\tilde{H}(10, k)$ в первом сегменте $[1, 3]$ вычисляются следующим образом:

- при $k = 1$ есть 10 позиций окна длины 1 и, следовательно, возможно получить равночастотную встречаемость символов алфавита (как, например, в слове $w_1 = (ab)_5$), что влечет $\tilde{H}(10, 1) = 1$;
- при $k = 2$ есть 9 позиций окна длины 2 при четырех возможных подсловах. Поскольку $9 \text{ div } 4 = 2$,

Рис. 3. График мажоранты функции энтропии слов $\tilde{H}(10, k)$

то всем подсловам первоначально назначается частота 2, а поскольку $9 \text{ mod } 4 = 1$, то одно подслово будет иметь частоту 3. Тем самым одна из вероятностей в модели будет равна $3/9$, а остальные три — $2/9$, что доставляет значение $\tilde{H}(10, 2) \approx 0,987$;

- при $k = 3$ есть восемь позиций окна ширины 3 и 8 возможных подслов, таким образом, мы можем получить равночастотность и $\tilde{H}(10, 3) = 1$. Пример — слово $w_2 = (aaabbbabaa)$.

Потеря разнообразия происходит в окне ширины 4, и в сегменте $[4, 10]$ значения вычисляются по формуле (8). Результаты приведены в табл. 3 и на рис. 3.

Заметим, что в первом сегменте значений аргумента k (до потери разнообразия) не все значения $\tilde{H}(10, k)$ равны единице — для значения $k = 2$ это объясняется тем, что в этом случае число позиций окна не кратно числу возможных подслов данного алфавита.

Характеристический функционал на множестве слов над конечным алфавитом

В силу определения (7) любое случайно выбранное подслово длины n из случайного (в смысле фон Мизеса [10]) бесконечного слова не может иметь значения функции энтропии большие, чем мажоранта $\tilde{H}(n, k)$. Поскольку и функция энтропии исследуемого слова, и мажоранта определены на сегменте $[1, n]$, мы можем использовать формализм метрического пространства с интегральным или чебышевским расстоянием для функций, определенных на сегменте $[1, n]$.

Для решения поставленной задачи для слова w , $|w| = n$, введем в рассмотрение следующие две формы характеристического функционала, отражающие расстояние функции энтропии исследуемого слова $H(w, k)$ до функции $\tilde{H}(n, k)$. При этом мы учитываем целочисленность аргумента обеих функций.

1. Нормированная форма, основанная на интегральном расстоянии

$$\mu_M(w) = \frac{1}{n} \sum_{k=1}^n (\tilde{H}(n, k) - H(w, k)). \quad (9)$$

2. Форма, основанная на чебышевском расстоянии

$$\mu_T(w) = \max_{k=1, n} (\tilde{H}(n, k) - H(w, k)). \quad (10)$$

Поскольку все значения функции энтропии и мажоранты находятся в сегменте $[0, 1]$, то и значения каждой из форм также принадлежат этому сегменту. По определениям (9) и (10) малые значения $\mu_M(w)$ и $\mu_T(w)$ говорят о близости исследуемого слова к "случайным" словам такой же длины. "Случайное" слово здесь понимается как любое случайно выбранное подслово длины n из случайного в смысле фон Мизеса [10] бесконечного слова. Большие значения позволяют говорить о наличии периодичности или об отсутствии возможного разнообразия, например, при отсутствии в исследуемом слове части символов используемого алфавита или их очень редкой встречаемости.

Вычисление функций $H(w, k)$ и $\tilde{H}(n, k)$ на всем сегменте $[1, n]$, особенно функции $H(w, k)$, достаточно трудоемко, особенно для слов большой длины (геномы и т. д.). Вместе с тем, при больших значениях аргумента, близких к длине исследуемого слова, изучение поведения обеих функций не представляет особого интереса. Наибольшие различия проявляются в области от $k = 1$ до аргумента, кратного точке потери разнообразия, т. е. до $\hat{k} \approx mk^*$. В связи с этим мы рекомендуем для длинных слов в формулах (9) и (10) изменить верхние пределы суммирования и поиска максимума (n) до порога, кратного k^* , например до $\hat{k} \approx 3k^*$.

Для функционала в чебышевской форме интерес представляет и значение аргумента, при котором достигается максимум — его сравнение с k^* также позволяет получить полезную информацию. Например, если точка максимума разности лежит левее k^* и значение $\mu_T(w)$ велико, то, скорее всего, слово w — периодическое или близкое к периодическому, например слово Штурма [3].

Приведем примеры вычисления характеристического функционала для проанализированных выше модельных слов (данные из табл. 1, 2 и 3):

- для слова $w_1 = (ab)_5$: $\mu_M(w) = 0,238247$ и $\mu_T(w) = 0,49193$ при $k = 2$;
- для слова $w_2 = (aaabbbabaa)$: $\mu_M(w) = 0,002905$ и $\mu_T(w) = 0,02905$ при $k = 1$. В этом примере отметим, что для всех остальных значений аргумента разность $\tilde{H}(n, k) - H(w, k) = 0$, $\mu_T(w)$ мало и поэтому слово w_2 очень близко к случайному.

Возможные применения в задачах кластеризации, распознавания и прогнозирования

Предложенные выше формы функционала $\mu_M(w)$ и $\mu_T(w)$ могут быть применены в задачах анализа данных для выбора эталонных значений классов. В задачах кластеризации возникает проблема оценки числа кластеров и положения их центров. При недостаточно плотных выборках обучающих данных решение этих проблем приводит к значительным трудностям [11]. Очевидное требование — чтобы расстояния между элементами одного кластера были по возможности значительно меньше расстояний

между разными кластерами. Для уменьшения расстояний между элементами одного кластера желательно выбрать центр кластера так, чтобы был обеспечен минимум максимального расстояния от этого центра до элементов этого кластера. Этот центр может не совпадать ни с одним элементом, но в большинстве методов за центр выбирается именно некоторый элемент. Выбор центра на основе предложенного характеристического функционала позволяет устранить это ограничение — как было отмечено, мажоранта функции энтропии не обязана соответствовать реальному элементу. Аналогичная ситуация возникает при выборе эталонов классов в задачах классификации.

В задачах прогнозирования временных рядов функция мажоранты может быть полезна для оценки степени случайности анализируемых процессов. Для решения задачи прогнозирования необходимо выделять в анализируемых временных рядах некоторые закономерности. Наибольшие трудности при прогнозировании возникают при работе с последовательностями данных, в которых таких закономерностей не наблюдается. Энтропия таких последовательностей велика и близка к мажоранте.

Тот же критерий пригоден и для проверки качества генераторов псевдослучайных чисел. Здесь, наоборот, энтропия получающихся последовательностей должна быть близка к максимуму. Поскольку критерий дает количественную оценку близости к случайности, он полезен и для оценки "вырождения" генераторов при увеличении размерности пространства, заполняемого последовательно снимаемыми с данного генератора кортежами чисел. Известно, что генераторы псевдослучайных чисел с равномерным распределением, основанные на детерминированных процедурах, которые очень хорошо работают в одномерном случае, перестают корректно работать в многомерном уже при небольших размерностях. Порог размерности, при котором еще целесообразно применять генератор, можно оценить по изменению значений предложенных форм функционала при увеличении размерности заполняемого пространства.

Заключение

Таким образом, для заданного конечного алфавита и множества слов над этим алфавитом определены две формы характеристического функционала, дающие количественную оценку близости изучаемого слова конечной длины k подслову такой же длины, случайно выбранному из случайного бесконечного слова над этим же алфавитом. Указаны возможности использования предложенных двух форм характеристического функционала при решении различных задач анализа данных, в том числе задач кластеризации, распознавания и прогнозирования.

Список литературы

1. Lind D., Marcus B. *An Introduction to Symbolic Dynamics and Coding*. Cambridge, UK: Cambridge University Press, 1995. 495 p.
2. Сметанин Ю. Г., Ульянов М. В., Пестова А. С. Энтропийный подход к построению меры символического разнообразия слов и его применение к кластеризации геномов растений // *Математическая биология и биоинформатика*. 2016. Т. 11, № 1. С. 114—126. DOI: 10.17537/2016.11.114.
3. Lothaire M. *Algebraic Combinatorics on Words*. Cambridge: Cambridge University Press. 2002. 455 p. URL: <http://www-igm.univ-mlv.fr/~berstel/Lothaire/>.
4. Сметанин Ю. Г., Ульянов М. В. Мера символического разнообразия: подход комбинаторики слов к определению обобщенных характеристик временных рядов // *Бизнес-информатика*. 2014. № 3 (29). С. 40—46.
5. Петрушин В. Н., Ульянов М. В., Чертихина И. А., Никольчев Е. В. Бикритериальный метод построения гистограмм // *Информационные технологии и вычислительные системы*. 2012. № 4. С. 22—31.

6. Ульянов М. В., Сметанин Ю. Г. Подход к определению характеристик колмогоровской сложности временных рядов на основе символических описаний // *Бизнес-Информатика*. 2013. № 2. С. 49—54.
7. Шеннон К. Работы по теории информации и кибернетике. М.: Изд-во иностр. лит., 1963. 830 с.
8. Мартин Н., Инглэнд Дж. *Математическая теория энтропии*. М.: Мир, 1988. 350 с.
9. Smetanin Y. G., Ulyanov M. V. Reconstruction of a Word from a Finite Set of its Subwords under the unit Shift Hypothesis. I. Reconstruction without for Bidden Words // *Cybernetics and Systems Analysis*. January 2014. Vol. 50, Is. 1. P. 148—156.
10. Верещагин Н. К., Успенский В. А., Шень А. Колмогоровская сложность и алгоритмическая случайность. М.: МЦНМО, 2010. 576 с.
11. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. *Прикладная статистика: Классификация и снижение размерности*. М.: Финансы и статистика, 1989. 607 с.

M. V. Ulyanov, Leading Researcher, Professor, e-mail: muljanov@mail.ru,
V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences,
Lomonosov Moscow State University,

Yu. G. Smetanin, Chief Researcher, Professor, e-mail: ysmetanin@rambler.ru,
Federal Research Center "Informatics and Control" of the Russian Academy of Science,
Federal state autonomous educational institution of higher professional education
Moscow Institute of Physics and Technology (State University)

On a Characteristic Functional for Words over a Finite Alphabet

The objects studied in the article are finite words over some finite alphabet. These words are symbolical codes of the investigated objects and processes that are analyzed further. A formalization of the entropy on words function is proposed that is based on the entropy of discrete distributions. Using the results of analysis of sets of words with fixed length over a given alphabet, a notion of entropy function majorant is proposed. The entropy on words function and the majorant form the basis for the forms of a characteristic functional proposed in the article, which give a quantitative estimation of the proximity of the tested finite word to a subword of the same length, randomly selected from a random infinite word over the same alphabet. The proposed forms of the functional may be used for solving different tasks of data analysis, including clustering and pattern recognition tasks.

Keywords: information entropy, words over a finite alphabet, entropy on words function, majorant of the entropy function, characteristic functional

References

1. Lind D., Marcus B. *An Introduction to Symbolic Dynamics and Coding*. Cambridge, UK, Cambridge University Press, 1995. 495 p.
2. Smetanin Yu. G., Ulyanov M. V., Pestova A. S. Entropiyni podkhod k postroeniiu mery simvol'nogo raznoobrazia slov i ego primneniye k klasterizatsii genomov rastenii, *Matematicheskaya biologiya i bioinformatika*, 2016, vol. 11, no. 1, pp. 114—126.
3. Lothaire M. *Algebraic Combinatorics on Words*. Cambridge, Cambridge University Press. 2002. 455 p. URL: <http://www-igm.univ-mlv.fr/~berstel/Lothaire/>.
4. Smetanin Yu. G., Ulyanov M. V. Mera simvol'nogo raznoobrazia: podkhod kombinatoriki slov k opredeleniiu obobshchennykh kharakteristik vremennykh riadov, *Biznes-informatika*, 2014, no. 3 (29), pp. 40—46 (in Russian).
5. Petrushin V. N., Ulyanov M. V., Chertikhina I. A., Nikul'chev E. V. Bikriteriial'nyi metod postroeniia gistogramm, *Informatisionnye tekhnologii i vychislitel'nye sistemy*, 2012, no. 4, pp. 22—31 (in Russian).

6. Ulyanov M. V., Smetanin Yu. G. Podkhod k opredeleniiu kharakteristik kolmogorovskoi slozhnosti vremennykh riadov na osnove simvol'nykh opisaniy, *Biznes-Informatika*, 2013, no. 2, pp. 49—54 (in Russian).
7. Shennon K. *Raboty po teorii informatsii i kibernetike*. Moscow, Izd-vo inostr. lit., 1963. 830 p. (in Russian).
8. Martin N., Ingland Dzh. *Matematicheskaya teoriya entropii*. Moscow, Mir, 1988, 350 p. (in Russian).
9. Smetanin Y. G., Ulyanov M. V. Reconstruction of a Word from a Finite Set of its Subwords under the unit Shift Hypothesis. I. Reconstruction without for Bidden Words, *Cybernetics and Systems Analysis*, January 2014, vol. 50, Is. 1, pp. 148—156.
10. Vereshchagin N. K., Uspenskii V. A., Shen' A. *Kolmogorovskaia slozhnost' i algoritmicheskaya sluchainost'*. Moscow, MTsNMO, 2010. 576 p. (in Russian).
11. Aivazian S. A., Bukhshtaber V. M., Eniukov I. S., Mashal'kin L. D. *Prikladnaya statistika: Klassifikatsiya i snizhenie razmernosti*. Moscow, Finansy i statistika, 1989, 607 p. (in Russian).

Д. А. Никифоров, аналитик, e-mail: Denis.Nikiforov@centre-it.com,
 Д. В. Корж, нач. отд. бизнес-анализа, e-mail: Dmitriy.Korj@centre-it.com,
 Р. Л. Сиваков, директор, e-mail: Ruslan.Sivakov@centre-it.com,
 ООО "Центр информационных технологий", г. Екатеринбург

Обзор инструментов для валидации XML-документов с помощью правил контроля, описанных на объектном языке ограничений (OCL)

Описывается подход к семантической валидации XML-документов, основанный на архитектуре, управляемой моделями. Рассмотрены инструменты, реализующие данный подход (ShapeChange, NIEM PIM to Schematron Transformation, eXolutio, Dresden OCL, Eclipse OCL, OCLE), проведено их сравнение. Обозначены основные направления развития таких инструментов.

Ключевые слова: семантическая валидация XML, объектный язык ограничений, модельно-ориентированная разработка, платформено-независимая модель

Введение

Человечество занимается хранением, обработкой и передачей данных на протяжении тысячелетий. Начиная со второй половины XX века эффективность этой деятельности существенно возросла благодаря разработке целого ряда языков моделирования данных (UML, IDEF1X, Anchor, XSD и др.), форматов представления данных (XML, JSON, EDIFACT, CSV и др.), языков программирования как общего назначения (Java, C# и др.), так и специализированных (SQL, OQL, XQuery и др.), используемых для обработки данных. Однако число новых языков и технологий только возрастает, благодаря чему современные информационные системы или комплексы интегрированных информационных систем все больше напоминают Вавилонскую башню, которую очень сложно строить и поддерживать.

Например, в некоторой информационной системе структуру базы данных может описывать реляционная модель, структуру передаваемых XML-сообщений — XML-схемы, структуру данных в слое бизнес-логики — Java-классы. Как правило, все эти модели описывают сведения об одних и тех же объектах реального мира, а различаются только техническими деталями [1]. При изменении одной из моделей соответствующим образом должны быть изменены и остальные. Такое дублирование приводит к возникновению ошибок в информационных системах, усложняет их разработку и сопровождение. Отчасти это может быть нивелировано инструментариями отображения моделей данных, например, инструментариями объектно-реляционного отображения [2] (рис. 1).

В общем случае число подобных отображений пропорционально квадрату числа используемых технологий представления данных. Можно упростить эту схему, выбрав одну из моделей в качестве основной, при этом остальные модели будут фор-

мироваться на ее основе. Однако если проект относительно масштабный и долгосрочный, то целесообразно исключить из этой модели все избыточные технические детали, которые подвержены частым изменениям и которые могут варьироваться у разных компаний, задействованных в разработке информационной системы. Такие модели называют платформено-независимыми (Platform-Independent Model, PIM), а модели, привязанные к особенностям некоторой технологической платформы, называют платформено-зависимыми (Platform-Specific Model, PSM). Описанный подход был предложен консорциумом Object Management Group (OMG) в рамках архитектуры, управляемой моделями [3], и применяется во многих проектах [4—6].

Для разработки платформено-независимых моделей данных (рис. 2) используют различные языки моделирования, один из наиболее популярных на данный момент — это унифицированный язык моделирования (Unified Modeling Language, UML) [7]. Хорошо известны подходы к формированию на основе UML-моделей реляционных моделей данных

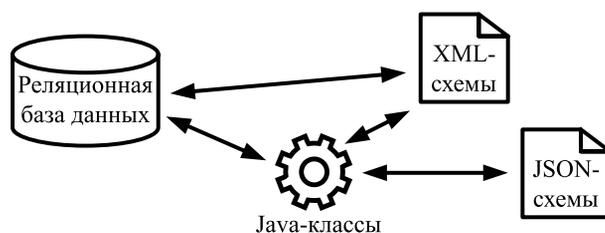


Рис. 1. Пример отображений моделей данных



Рис. 2. Пример отображений моделей данных в соответствии с архитектурой, управляемой моделями

[8], XML-схем [9], объектных моделей [10] и других платформо-зависимых моделей (рис. 2).

В простейшем случае такие отображения учитывают только элементарные структурные ограничения:

- на допустимые виды элементов данных (виды сущностей, объектов);
- на реквизитный состав сведений о сущностях (набор допустимых атрибутов объектов);
- на область значений элементов данных (типы данных);
- на обязательность и допустимую множественность элементов данных.

Однако модель данных может содержать и более сложные ограничения, например, на сумму значений нескольких элементов данных [11, 12]. Такие правила контроля называют семантическими или прикладными [13]. Равно как и простейшие структурные ограничения, прикладные правила контроля, даже в рамках одного приложения, могут быть реализованы на различных языках. В реляционных СУБД для этого может использоваться SQL, для валидации XML-документов — Schematron и XPath, на стороне сервера — Java, на стороне клиента — JavaScript и т. д. С прикладной точки зрения все это одни и те же правила контроля, отличающиеся только технической реализацией. Для упрощения разработки информационных систем целесообразно описывать их на уровне платформо-независимых моделей данных и затем использовать для валидации данных на требуемых технологических платформах. Существуют различные языки спецификации прикладных правил контроля и различные подходы к валидации данных с их помощью [14–17].

В данной работе рассмотрены инструменты, позволяющие валидировать XML-документы с помощью платформо-независимых правил контроля, описанных на объектном языке ограничений (Object Constraint Language, OCL) [18–20]. В разд. 1 дано описание схемы работы каждого инструмента, в разд. 2 — сравнение инструментов между собой. В разд. 3 обозначены основные направления развития подобных инструментов.

1. Обзор инструментов

1.1. ShapeChange

ShapeChange — это инструмент, который позволяет на основе платформо-независимых UML-моделей формировать платформо-зависимые схемы данных: XML-схемы, JSON-схемы, схемы реляционных баз данных и другие схемы [21]. Исходная UML-модель должна соответствовать стандарту ISO 19109 [22] и описывать структуру некоторой географической информации. UML-модель может содержать дополнительные ограничения, описанные на языке OCL, которые преобразуются в выражения соответствующего платформо-зависимо-

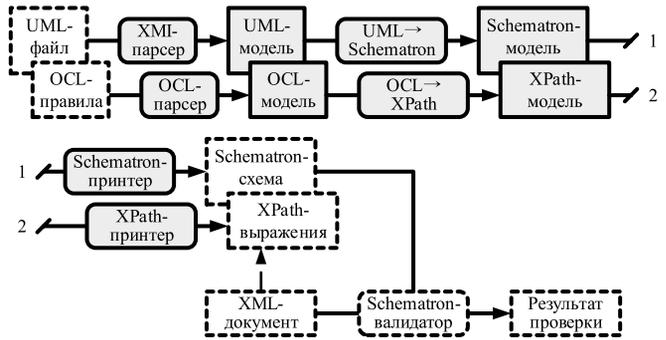


Рис. 3. Схема валидации XML-документов с помощью ShapeChange

го языка ограничений. В частности, для валидации XML-документов ShapeChange может формировать Schematron-схемы [23] с XPath-выражениями [24] (рис. 3).

На этом и последующих рисунках прямоугольники с прямыми углами изображают данные; прямоугольники с закругленными углами — преобразования данных; серые прямоугольники со сплошной границей — данные, структура которых определена рассматриваемым инструментом, и преобразования, реализованные в рамках данного инструмента; прямоугольники с пунктирной границей — соответственно данные, структура которых определена некоторым внешним инструментом или стандартом, и преобразования, реализованные в рамках некоторого внешнего инструмента; сплошные стрелки без подписи — потоки данных; штриховые стрелки — соответствие данных или процесса их спецификации.

Исходная UML-модель, как и в остальных инструментах, рассматриваемых в данной работе, должна быть представлена в формате XML Metadata Interchange (XMI) [25]. В ShapeChange реализован XMI-парсер, который на основе текстового представления UML-модели формирует ее объектное представление. Затем на его основе формируется объектная модель Schematron-схемы, которая с помощью Schematron-принтера преобразуется в ее текстовое представление. Параллельно осуществляется аналогичное преобразование для каждого OCL-правила из исходной UML-модели.

Результатом работы ShapeChange является Schematron-схема с XPath-выражениями, которая может использоваться для валидации XML-документов, структура которых соответствует исходной UML-модели. При использовании описанного подхода предъявляются относительно низкие требования к среде, в которой будет осуществляться валидация XML-документов. Для этого можно использовать произвольный Schematron-валидатор. Также на основе Schematron-схем могут быть сформированы XSLT-преобразования, и для валидации XML-документов можно использовать произвольный XSLT-процессор.

Безусловно, ShapeChange решает задачи, для которых он предназначен, однако область его применения ограничена.

Во-первых, исходная UML-модель должна соответствовать стандарту ISO 19109. ShapeChange неприменим в проектах, в которых используются платформо-независимые модели, основанные на других стандартах [4–6].

Во-вторых, в ShapeChange определены правила преобразования OCL-выражений в XPath-выражения только для подмножества стандартной библиотеки OCL. Это существенно ограничивает выразительность поддерживаемых OCL-правил.

В-третьих, не ясен механизм, позволяющий использовать в OCL-выражениях типы данных и операции, которые не определены в стандартной библиотеке OCL, например, операции для работы с датой и временем.

В-четвертых, не рассматривается вопрос описания на языке OCL правил контроля, основанных на внешних по отношению к валидируемому XML-документу источниках данных, например правил проверки кода по справочнику. Не рассматривается преобразование таких правил в XPath-выражения.

1.2. NIEM

Национальная модель обмена информацией (National Information Exchange Model, NIEM) — это открытый стандарт в области межведомственного обмена данными, основанный на архитектуре, управляемой моделями. В спецификации [4] описаны правила построения платформо-независимых и платформо-зависимых моделей данных, описаны правила преобразования моделей данных, а также утверждается, что платформо-независимая UML-модель данных может быть дополнена правилами контроля на языке OCL, которые могут быть преобразованы в Schematron-схему с XPath-выражениями.

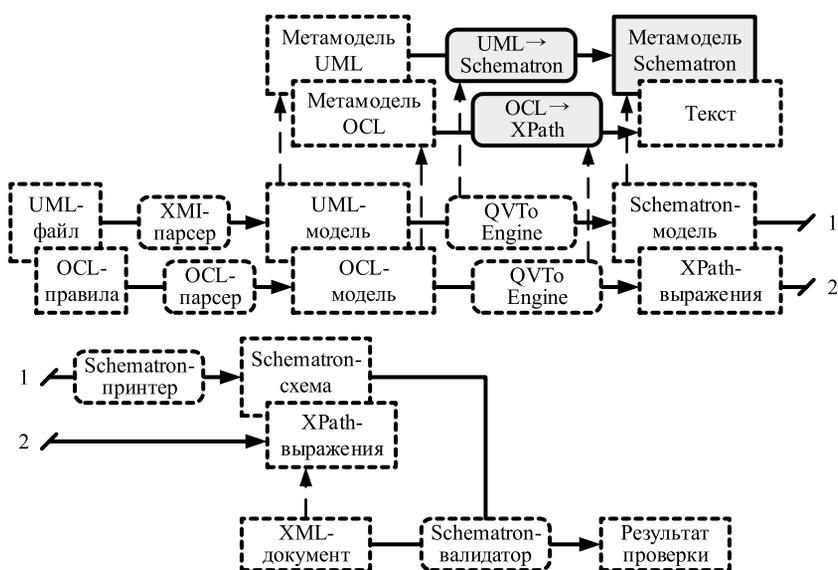


Рис. 4. Схема валидации XML-документов в рамках проекта NIEM

В спецификации [4] данное преобразование не описано, неофициальный вариант доступен в репозитории [26]. Схема работы преобразования представлена на рис. 4.

Исходные и выходные артефакты преобразования аналогичны соответствующим артефактам ShapeChange. Отличие заключается в том, что исходная UML-модель должна соответствовать спецификации [4], а не стандарту ISO 19109. Преобразование OCL-выражений в XPath-выражения покрывает большую, чем ShapeChange, часть спецификации OCL. Однако оно не поддерживает расширение OCL дополнительными операциями, например, для работы с датой и временем, не поддерживает правила контроля, основанные на внешних источниках данных.

В отличие от ShapeChange преобразование реализовано на предметно-ориентированном языке Operational Query/View/Transformation Language (QVTo) [27], а не на языке Java. QVTo — это один из рекомендованных консорциумом OMG языков для описания преобразований моделей в рамках архитектуры, управляемой моделями. Использование этого и других стандартов консорциума OMG позволяет существенно упростить реализацию подобных преобразований. Все необходимые парсеры и принтеры для моделей либо уже реализованы, например, в рамках проекта Eclipse Modeling Project [28], либо могут быть автоматически сформированы на основе соответствующих метамodelей. Для выполнения QVTo-преобразований может использоваться Eclipse QVT Operational [28].

Преобразование UML-моделей в Schematron-схемы описано на уровне моделей. Исходная модель должна соответствовать метамодели UML [7] и спецификации [4], а результирующая — метамодели Schematron [29]. При преобразовании OCL-выражений в XPath-выражения исходная модель должна соответствовать метамодели OCL [18], однако результирующие XPath-выражения формируются сразу в текстовом представлении, минуя объектное представление. Недостаток такого подхода заключается в том, что в описываемом преобразовании полностью отсутствует какой-либо анализ результирующих XPath-выражений, например, производимый в целях их упрощения.

1.3. eXolutio

eXolutio — это инструмент, предназначенный для концептуального моделирования XML-схем на платформо-независимом и платформо-зависимом уровнях [30]. Так же как и описанные выше инструменты, eXolutio позволяет на основе платформо-независимых UML-моделей с OCL-правилами формиро-

вать Schematron-схемы с правилами контроля на языке XPath (рис. 5).

Входные и выходные артефакты eXolutio аналогичны соответствующим артефактам инструментов, описанных выше. Основное отличие заключается в том, что OCL-правила можно использовать на двух уровнях — платформо-независимом и платформо-зависимом. Платформо-независимая модель представляет собой диаграмму классов UML. На ее основе могут быть построены производные платформо-зависимые модели, в которых один из классов выбирается в качестве корневого, ассоциации заменены на отношения "часть—целое", и соответствующим образом преобразованы OCL-правила.

Также в eXolutio используется отличный от других инструментов подход к преобразованию OCL-правил в XPath-выражения. Для многих операций из стандартной библиотеки OCL в рамках библиотеки OclX [31] реализованы аналогичные операции на языке XSLT. Вызовы OCL-операций преобразуются в вызовы их OclX-реализаций. При этом структура результирующих XPath-выражений в целом повторяет структуру исходных OCL-выражений. Фактически в рассматриваемом подходе OCL реализован на XSLT.

В работе [32] предлагается следующий механизм расширения OCL дополнительными типами данных и операциями. Пользователь определяет необходимые типы данных и операции в исходной UML-модели. Для операций указывает их спецификацию на языке OCL. Затем данные операции транслируются в XSLT-функции, которые можно использовать в XPath-выражениях подобно функциям из OclX. Недостаток такого подхода заключается в том, что в XPath уже реализованы многие операции, которые отсутствуют в OCL, например, операция *timezone-from-date*, которая определяет временную зону, указанную в дате. Очевидно, что в XPath-выражениях лучше использовать существующую операцию, чем аналогичную функцию, преобразованную из OCL в XSLT и XPath. Тем более что описывать подобные функции на OCL весьма затруднительно, так как данный язык предназначен для описания правил контроля, а не алгоритмов разбора текста.

В eXolutio не поддерживаются OCL-правила, использующие для валидации XML-документов внешние службы или источники данных. Однако в работе [32] рассматривается вариант расширения языка OCL конструкциями, позволяющими ссылаться на внешние по отношению к валидируемому XML-документы. Описанный подход слишком частный и не позволяет валидировать XML-документы с помощью удаленных служб или по справочникам, хранящимся в базах данных.

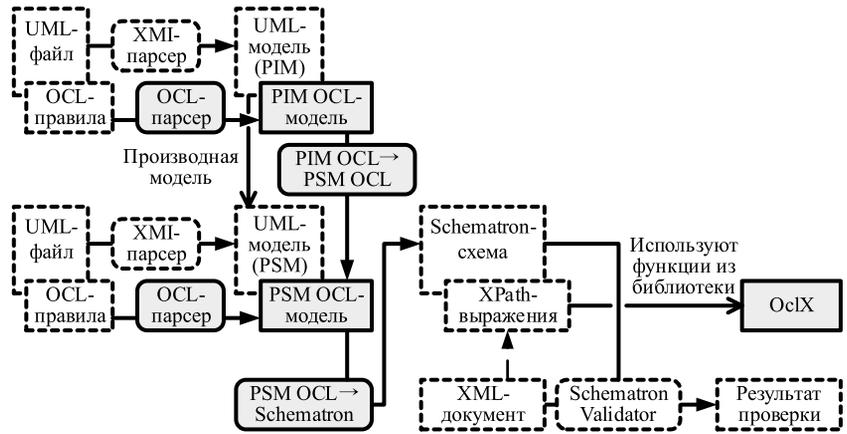


Рис. 5. Схема валидации XML-документов с помощью eXolutio

1.4. Dresden OCL

Dresden OCL — это инструмент для редактирования и интерпретации OCL-выражений, а также для формирования на их основе SQL или Java-кода [33]. Принципиальное отличие Dresden OCL от инструментов, рассмотренных выше, заключается в том, что для валидации XML-документов не формируются Schematron-схемы с XPath-выражениями. Вместо этого для XML-документа с помощью соответствующего адаптера формируется объектная модель, на которой интерпретируются OCL-выражения (рис. 6).

Аналогичным образом могут быть провалидированы не только XML-документы, но и экземпляры произвольных платформо-зависимых моделей данных, представимые в виде обобщенного экземпляра объектной модели данных, структура которого описана в документации Dresden OCL. Для этого достаточно разработать соответствующий адаптер экземпляров моделей данных. На данный момент в качестве экземпляров моделей Dresden OCL поддерживает XML-документы, экземпляры объектных моделей, реализованных на Java, или основанных на Ecote [34].

Более того, Dresden OCL позволяет описывать OCL-правила контроля не только для UML-модели

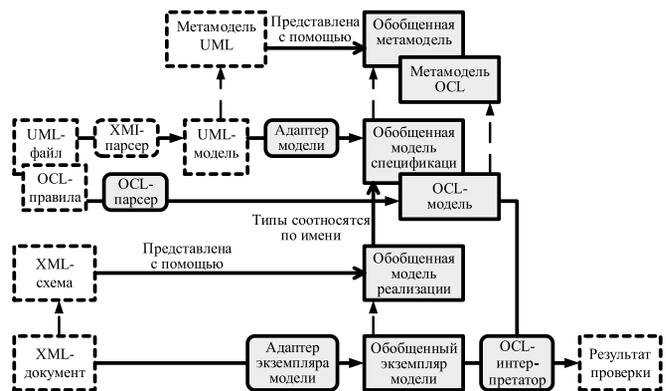


Рис. 6. Схема валидации XML-документов с помощью Dresden OCL

лей, но и для XML-схем, для объектных моделей, описанных с помощью Java-классов, для произвольных Ecore-моделей и для любых моделей, для которых разработан соответствующий адаптер.

Преимущество описанного подхода заключаются в том, что Dresden OCL с минимальными усилиями может быть адаптирован для поддержки произвольных видов моделей и их экземпляров. Реализовать необходимые адаптеры существенно проще, чем реализовать преобразования моделей, описанные выше. Также существенно проще проводить отладку OCL-правил, потому что они интерпретируются в неизменном виде, без трансляции их на другой язык.

Однако данный подход имеет и некоторые недостатки.

Во-первых, смешиваются платформо-независимые (на рис. 6 — обобщенная модель спецификации) и платформо-зависимые (на рис. 6 — обобщенная модель реализации) модели данных. Типы данных в них должны иметь одинаковые имена и в целом структурно соответствовать друг другу в отличие от подходов, описанных выше, в которых платформо-независимые и платформо-зависимые модели могут отличаться относительно сильно.

Во-вторых, на целевой платформе могут быть реализованы функции, отсутствующие в OCL, например, XPath-функция *timezone-from-date* для извлечения временной зоны из даты. Чтобы в OCL-правилах контроля можно было использовать такие функции, их необходимо реализовать либо на OCL, либо на базовом языке (в данном случае Java).

В-третьих, такой подход имеет относительно высокие требования к среде, в которой будет осуществляться валидация XML-документов. В информационной системе уже может существовать инфраструктура, ориентированная на валидацию XML-документов с помощью Schematron-схем, XML-схем версии 1.1 или XSLT-преобразований. Ее адаптация под Dresden OCL может быть затруднительной или невозможной. В частности, валидация с помощью Schematron-схем может быть более производительной и менее требовательной к объему оперативной памяти, чем формирование для XML-документа объектной модели и интерпретация на ней OCL-выражений.

Описанные недостатки несколько нивелируются тем, что Dresden OCL позволяет не только интерпретировать OCL-выражения, но и формировать на их основе SQL или Java-код. Преобразование OCL-Java используется в службе для валидации XML-документов [35], разработанной компанией Nomos Software (рис. 7).

Данная служба используется для валидации XML-документов, сформированных в соответствии со стандартом ISO 20022. Их структура описывается в виде платформо-независимых UML или Ecore-моделей данных, на основе которых формируются

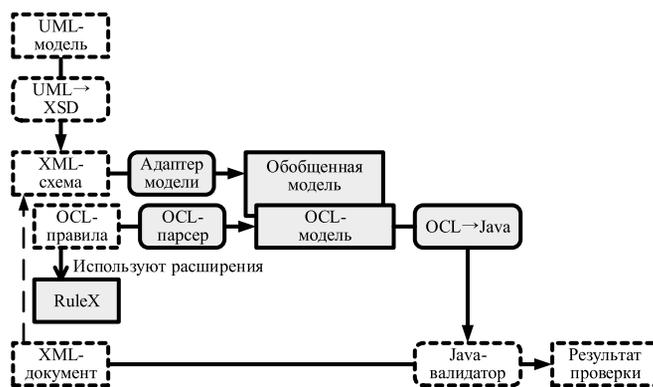


Рис. 7. Схема валидации XML-документов с помощью службы Nomos Software

XML-схемы. Однако OCL-правила пишутся не для исходных UML-моделей, как это делается в инструментах, описанных выше, а для XML-схем, преобразованных в обобщенные объектные модели.

Фактически, правила специфицируются на уровне платформо-зависимой модели, они привязаны к структуре XML-документов и их использование для валидации данных, представленных в виде реляционной или иной платформо-зависимой модели, невозможно. Хотя для описания правил и используется язык OCL, правила привязаны только к одной платформе. В связи с этим вполне естественно, что в рамках рассматриваемого проекта язык OCL был дополнен типами данных и операциями, специфичными для XML. Данное расширение называется RuleX [36]. Также в нем определены дополнительные операции для типов данных из стандарта ISO 20022 [37], включая операции для проверки кодов по внешним справочникам.

1.5. Eclipse OCL

Eclipse OCL — это реализация языка OCL, основанная на Eclipse Modeling Framework (EMF) [34]. Eclipse OCL, как и Dresden OCL, позволяет интерпретировать OCL-правила для обобщенной объектной модели. На данный момент поддерживаются только UML и Ecore-модели. Однако Eclipse OCL можно использовать и для валидации XML-документов следующим образом (рис. 8).

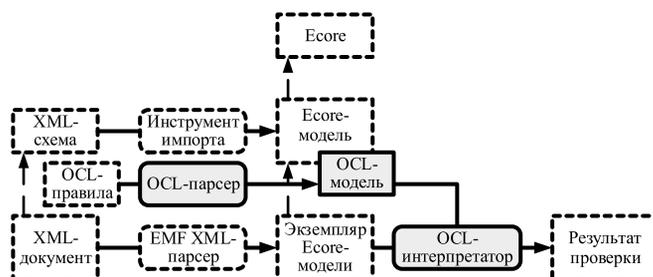


Рис. 8. Схема валидации XML-документов с помощью Eclipse OCL

На основе XML-схемы документа с помощью инструмента импорта моделей создается Ecore-модель, соответствующая этой схеме. Данная модель содержит расширенные метаданные, позволяющие (де)сериализовать ее экземпляры не только в формате XML, но и в виде XML-документов, соответствующих исходной XML-схеме. Таким образом XML-документ может быть десериализован в экземпляр Ecore-модели, построенной на основе исходной XML-схемы. А уже на данном экземпляре модели с помощью Eclipse OCL можно интерпретировать OCL-правила контроля.

Инструмент импорта моделей и парсер экземпляров Ecore-моделей реализованы в рамках EMF. Отметим, что Eclipse OCL совместно с Eclipse QVT Operational можно использовать и для кодогенерации. Пример был описан выше, в разделе, посвященном NIEM.

1.6. OCLE

Object Constraint Language Environment (OCLE) — это инструмент, предназначенный для UML-моделирования и поддерживающий язык OCL [38]. OCLE, как и Dresden OCL, позволяет на основе OCL-правил формировать Java-код. Также OCLE позволяет интерпретировать OCL-правила на XML-документах [39] (рис. 9).

В инструментах, рассмотренных выше, платформо-независимые и платформо-зависимые модели данных описывают структуру одних и тех же данных. В OCLE же дополнительно присутствует логический уровень, на котором описывается модель данных информационной системы в целом, охватывающая все XML-документы, базы данных и т. д. Для валидации некоторого XML-документа необходимо на основе его DTD-схемы сформировать UML-модель, описывающую его структуру в

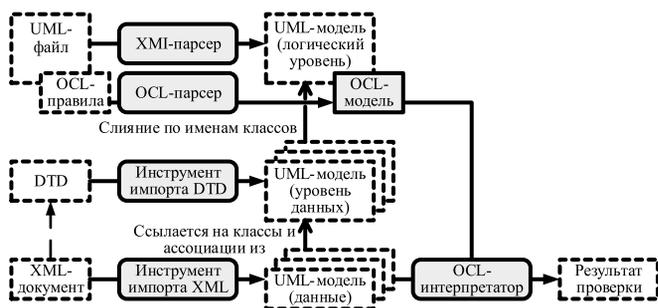


Рис. 9. Схема валидации XML-документов с помощью OCLE

платформо-независимом виде. Данная модель может быть слита с единой логической моделью по именам классов. Также для XML-документа необходимо сформировать диаграмму объектов UML. Затем с помощью OCLE можно интерпретировать на полученных моделях OCL-правила. Отличительная особенность OCLE заключается в том, что OCL-правила могут охватывать разные XML-документы, базы данных и другие источники данных.

2. Сравнение инструментов

В рассмотренных инструментах применяют два подхода к валидации XML-документов с помощью OCL-правил: 1) преобразование OCL-правил в выражения на исполняемом на некоторой технологической платформе языке: Schematron и XPath или Java; 2) интерпретация OCL-правил на объектной модели XML-документа (табл. 1).

При использовании 1-го подхода OCL-правила адаптируются для исполнения на целевой технологической платформе. При использовании 2-го подхода, наоборот, данные, представленные в специфическом для некоторой технологической платформы виде, адаптируются для интерпретации на

Таблица 1

Сравнение инструментов

Инструмент	Механизм валидации	Поддержка OCL	Исходная PIM
ShapeChange	Преобразование OCL в Schematron и XPath	Очень ограниченная	UML-модель, соответствующая стандарту ISO 19109
NIEM	Преобразование OCL в Schematron и XPath	Ограниченная	UML-модель, соответствующая спецификации NIEM
eXolutio	Преобразование OCL в Schematron и XPath	Средняя: поддерживаются iterate, closure и др., но, например, неизвестно поддерживается ли substring, в работе [32] это не описано; потенциально поддержка OCL может быть полной	Диаграмма классов UML
Dresden OCL	Интерпретация OCL или преобразование OCL в Java	Потенциально полная; последняя версия от 2015 г., может не соответствовать актуальной спецификации OCL	Произвольная модель, с адаптером; адаптеры реализованы для UML, Ecore, Java, XSD; в службе валидации Nomos Software — XML-схема, соответствующая стандарту ISO 20022
Eclipse OCL	Интерпретация OCL	Полная; на данный момент Eclipse OCL является наиболее полной и актуальной реализацией спецификации OCL	Произвольная модель, с адаптером; адаптеры реализованы для UML и Ecore
OCLE	Интерпретация OCL или преобразование OCL в Java	Потенциально полная; последняя версия от 2005 г., может не соответствовать актуальной спецификации OCL	Диаграмма классов UML

Сравнение инструментов (продолжение)

Инструмент	Механизм расширения OCL	Поддержка внешних источников	Реализация
ShapeChange	Отсутствует	Отсутствует	Все преобразования реализованы на языке Java на уровне объектных моделей
NIEM	Отсутствует	Отсутствует	Все преобразования реализованы на языке QVTo, XPath-выражения формируются сразу в текстовом виде
eXolutio	Пользовательские UML-классы с операциями; операции специфицированы на языке OCL и преобразуются в XSLT-функции	Отсутствует; описывается вариант расширения синтаксиса OCL для ссылки на другие XML-документы	Сведения отсутствуют, предположительно преобразования реализованы на языке C#
Dresden OCL	Пользовательские UML-классы с операциями; операции специфицированы на языке OCL	Отсутствует	Преобразования и интерпретатор реализованы на языке Java с использованием EMF
Eclipse OCL	Пользовательские UML-классы с операциями; операции специфицированы на языке OCL	Отсутствует	Интерпретатор реализован на языке Java с использованием EMF
OCLC	Пользовательские UML-классы с операциями; операции специфицированы на языке OCL	Все источники данных определены в единой логической модели, на них можно ссылаться из OCL-правил	Сведения отсутствуют

них OCL-правил. 1-й подход предпочтителен, если предъявляются высокие требования к среде, в которой будет проводиться валидация XML-документов. Если же накладные расходы, связанные с построением объектной модели валидируемых данных, не играют существенной роли, и нет ограничений на набор используемых технологий, то применение 2-го подхода более целесообразно. При этом проще добиться более полной поддержки спецификации языка OCL для описания правил.

Во всех рассмотренных инструментах для описания структур валидируемых XML-документов использованы разные, несовместимые между собой платформо-независимые модели. Если в проекте по разработке некоторой информационной системы использована другая платформо-независимая модель, то для применения одного из инструментов требуется ее предварительная адаптация.

В спецификации OCL определена очень ограниченная система примитивных типов данных, в которой отсутствуют даже типы для представления даты и времени. Это компенсируется тем, что дополнительные типы данных и операции могут быть определены в UML-моделях. Однако не во всех инструментах реализован механизм их интерпретации или трансляции (табл. 2).

Иногда требуется проверка данных, содержащихся в XML-документе, с помощью удаленных служб или внешних источников данных. Например, кодовые значения могут проверяться по справочникам, хранящимся в реляционных базах данных. В существующих инструментах поддержка подобных правил контроля данных очень ограничена.

Многие из рассмотренных инструментов представляют собой монолитные приложения, написанные на языках программирования общего назначения. Это существенно затрудняет понимание

того, как именно в них реализованы преобразования UML-моделей и OCL-правил, затрудняет разработку данных преобразований для поддержки других платформо-независимых моделей, для более полной поддержки спецификации OCL и т. д. Исключение составляет только преобразование, реализованное в рамках NIEM на языке QVTo.

Заключение

В работе рассмотрены подходы к интерпретации OCL-правил и их преобразованию в XPath-выражения, проблемы, возникающие при валидации таким образом XML-документов, а также различные варианты решения данных проблем. Аналогичным образом валидация данных может осуществляться и на других технологических платформах (реляционные базы данных, модели данных, реализуемые на языках программирования, и др.). Данный обзор может быть полезен для разработчиков таких инструментов. Независимо от используемой технологической платформы, возникают вопросы расширения системы типов OCL, валидации данных с помощью внешних источников и др.

Однако есть ряд вопросов, которые ни в одном из рассмотренных инструментов не затрагиваются.

Во-первых, неясно насколько эквивалентны исходные OCL-выражения и результирующие XPath-выражения, не изменяется ли при преобразовании их семантика. Требуется формальное доказательство корректности таких преобразований, но во всех рассмотренных подходах оно отсутствует. Для инструментов, интерпретирующих OCL, требуется формальное доказательство того, что применять OCL-правила к адаптированной объектной модели валидируемых данных корректно.

Во-вторых, система типов OCL существенно отличается от системы типов XSD. Во многих инст-

рументах первая расширяется типами из второй. Однако это привязывает OCL-правила к конкретной технологической платформе, и, например, использовать их для валидации данных, хранящихся в реляционной базе данных, невозможно, так как там используются совершенно другие типы. В качестве решения может быть разработана универсальная система примитивных типов данных, обобщающая типы XSD, SQL и др.

В-третьих, предполагается, что все правила контроля проверяются всегда и последовательно, в пакетном режиме. Не рассматривается возможность описания предусловий, на основе которых может быть принято решение о необходимости проверки того или иного правила контроля.

В-четвертых, не рассматривается механизм определения конкретного элемента данных, в котором обнаружена ошибка. Предполагается, что это всегда элемент, для которого определено правило.

В-пятых, в рассмотренных инструментах предполагается, что результат валидации XML-документа — это всегда перечень найденных ошибок. В общем случае это не так, может потребоваться отчет об успешно проверенных элементах данных, о том, что некоторые элементы данных не проверялись, так как не выполнены необходимые предусловия или элементы отсутствуют.

В-шестых, ни в одном из инструментов не рассматривается использование для XPath отличных от Schematron базовых языков, например, XSD версии 1.1, XSLT, Java. Вопрос, насколько при этом должно измениться преобразование OCL в XPath, остается открытым.

Перечисленные вопросы могут быть предметом дальнейших исследований в области семантической валидации XML-документов или данных, основанных на реляционной или иной платформо-зависимой модели.

Список литературы

1. **Nikiforov D. A., Lisikh I. G., Sivakov R. L.** An Approach to Multi-Domain Data Model Development Based on the Model-Driven Architecture and Ontologies // Supplementary Proceedings of the 4th International Conference on Analysis of Images, Social Networks and Texts (AIST'2015), Yekaterinburg, Russia, April 9–11, 2015. P. 106–117.
2. **Ambler S. W.** Mapping objects to relational databases // Software Development. 1995. Vol. 3. P. 63–72.
3. **Miller J., Mukerji J.** MDA Guide Version 1.0.1 // Object Management Group. 2003. URL: <http://www.omg.org/cgi-bin/doc?omg/03-06-01> (accessed: 14.12.2016).
4. **OMG.** UML Profile for National Information Exchange Model (NIEM), version 3.0. Object Management Group, 2015.
5. **ISO 20022-1:2003.** Financial services — Universal financial industry message scheme. Part 1: Metamodel.
6. **UN/CEFACT.** Core Components Technical Specification, version 3.0, United Nations Centre for Trade Facilitation and Electronic Business, 2009.
7. **OMG.** Unified Modeling Language (UML), version 2.5. Object Management Group, 2015.
8. **Li L., Zhao X.** UML Specification of Relational Database // Journal of Object Technology. September–October 2003. Vol. 2, N. 5. P. 87–100.

9. **Dominquez E., Lloret J., Pérez B., Rodriguez A., Rubio A. L., Zapata M. A.** A survey of UML models to XML schemas transformations. Berlin Heidelberg: Springer-Verlag, 2007. P. 184–195.
10. **Harrison W., Barton C., Raghavachari M.** Mapping UML designs to Java // ACM SIGPLAN Notices. Minneapolis, Minnesota, USA. 2000. Vol. 35, pp. 178–187.
11. **Wahler M., Koehler J., Brucker A. D.** Model-Driven Constraint Engineering // Workshop on OCL for (Meta-)Models in Multiple Application Domains at the UML/ModelS Conference. Genoa, Italy. 2006. P. 111–125.
12. **Dang D. H., Cabot J.** Automating Inference of OCL Business Rules from User Scenarios // 20th Asia-Pacific Software Engineering Conference, APSEC 2013. Ratchathewi, Bangkok, Thailand. 2013. Vol. 1. P. 156–163.
13. **Moskal J., Kokar M., Morgan J.** Semantic Validation of T & E XML Data // Proceedings of the International Telemetering Conference (ITC 2015). Las Vegas, NV, USA. 2015.
14. **Cabot J., Teniente E.** Constraint Support in MDA Tools: A Survey // Model Driven Architecture — Foundations and Applications: Second European Conference, ECMDA-FA 2006, Bilbao, Spain, July 10–13, 2006. Proceedings. Berlin-Heidelberg: Springer, 2006. P. 256–267.
15. **Demuth B., Hussmann H., Loecher S.** OCL as a Specification Language for Business Rules in Database Applications // In: UML-2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools: 4th International Conference Toronto, Canada, October 1–5, 2001 Proceedings. Berlin-Heidelberg: Springer, 2001. P. 104–117.
16. **Kallel S., Tibermacine C., Tramoni B., Dony C., Kacem A. H.** Automatic Translation of OCL Meta-Level Constraints into Java Meta-Programs // Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2015. Takamatsu, Japan. 2016. P. 213–226.
17. **Vergara-Niedermayr C., Wang F., Pan T., Kurc T., Saltz J.** Semantically Interoperable XML Data // International journal of semantic computing. 2013. Vol. 7. P. 237–255.
18. **OMG.** Object Constraint Language (OCL), version 2.4. Object Management Group, 2014.
19. **Nikiforov D. A.** The Object Constraint Language // Habrahabr. 2015. URL: <https://habrahabr.ru/company/cit/blog/264963/> (accessed: 14.12.2016).
20. **Gaafar A., Sakr S.** Towards a Framework for Mapping Between UML/OCL and XML/XQuery // UML — 2004 — The Unified Modeling Language. Modeling Languages and Applications: 7th International Conference, Lisbon, Portugal, October 11–15, 2004. Proceedings. Berlin — Heidelberg: Springer, 2004. P. 241–259.
21. **ShapeChange** — Processing application schemas for geographic information [Online] URL: <http://shapechange.net> (accessed: 14.12.2016).
22. **ISO 19109:2015.** Geographic information — Rules for application schema.
23. **ISO/IEC 19757-3:2016.** Information technology — Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation — Schematron.
24. **OCL Conversion to Schematron** // ShapeChange: [Web site]. URL: <http://shapechange.net/targets/xsd/extensions/ocl/> (accessed: 14.12.2016).
25. **OMG.** XML Metadata Interchange (XMI), version 2.5.1. Object Management Group, 2015.
26. **PIM 2 Schematron Transformation** // NIEM-UML: [Web site]. [2015]. URL: <https://github.com/NIEM/NIEM-UML/blob/master/Archive/Phase1/Specification/NIEM%20UML%20Machine%20Readable%20Files/NIEMpim2Schematron.qvto> (accessed: 14.12.2016).
27. **OMG.** Meta Object Facility (MOF) 2.0 Query/View/Transformation, version 1.2. Object Management Group, 2015.
28. **Gronback R. C.** Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit. 1st ed. Addison-Wesley Professional, 2009.
29. **Schematron Metamodel** [Online] // NIEM-UML: [Web site]. [2015]. URL: <https://github.com/NIEM/NIEM-UML/blob/master/Archive/Phase1/Specification/NIEM%20UML%20Machine%20Readable%20Files/Nonnormative/schematron.ecore> (accessed: 14.12.2016).
30. **Klmeck J., Malý J., Nečaský M., Holubová I.** eXolutio: Methodology for Design and Evolution of XML Schemas Using Conceptual Modeling // Informatica, Lith. Acad. Sci. 2015. Vol. 26. P. 453–472.
31. **Malý J., Nečaský M.** Evaluation of OCL Expressions over XML Data Model // Journal of Universal Computer Science. Mar 2014. Vol. 20. P. 329–365.

32. **Malif J.** XML Document Adaptation and Integrity Constraints in XML. Doctoral Thesis. Prague. 2013.

33. **Demuth B.** The Dresden OCL toolkit and its role in Information Systems development // In 13th International Conference on Information Systems Development: Methods and Tools, Theory and Practice. Vilnius, Lithuania. 2004. P. 1–12.

34. **Steinberg D., Budinsky F., Paternostro M., Merks E.** EMF: Eclipse Modeling Framework 2.0. 2nd ed. Addison-Wesley Professional, 2009.

35. **Garry D., Balfe T.** Experiences Using OCL for Business Rules on Financial Messaging // Proceedings of the 12th Workshop on OCL and Textual Modelling. New York, NY, USA. 2012. P. 65–66.

36. **RuleX Operations Reference** // Nomos Software: [Web site]. URL: <http://clients.nomos-software.com/RuleXOperationsReference.html> (accessed: 15.12.2016).

37. **RuleX ISO 20022 Operations** // Nomos Software: [Web site]. URL: <http://clients.nomos-software.com/RuleXISO20022Operations.html> (accessed: 14.12.2016).

38. **Chiorean D., Bortes M., Corutiu D.** Object Constraint Language Environment, a Tool Supporting Teaching and Learning UML and OCL, the Understanding and Using of Metamodeling, Abstraction and Design by Contract // Proceedings of Eight Workshop on Pedagogies and Tools for Teaching and Learning Object Oriented Concepts. 2000.

39. **Using OCLE** to ensure the syntactic and semantic validity of XML documents // Object Constraint Language Environment: [Web site]. URL: <http://lci.cs.ubbcluj.ro/ocle/um/xmlvalidation.htm> (accessed: 14.12.2016).

D. A. Nikiforov, Analyst, e-mail: Denis.Nikiforov@centre-it.com,
D. V. Korj, Head of Business Analysis, e-mail: Dmitriy.Korj@centre-it.com,
R. L. Sivakov, Director, e-mail: Ruslan.Sivakov@centre-it.com,
Centre of Information Technologies, Yekaterinburg

A Survey of Tools for XML Validation Based on the Object Constraint Language (OCL)

The article describes an approach to the semantic validation of XML documents based on the Model-Driven Architecture (MDA). The authors have examined several instruments implementing that approach: ShapeChange, NIEM PIM to Schematron Transformation, eXoluto, Dresden OCL, Eclipse OCL, Object Constraint Language Environment (OCLE). The authors have described two main validation methods: 1) a transformation of the OCL constraints to XPath assertions, and 2) an interpretation of the OCL constraints on the object models of XML documents. The authors have also listed the problems encountered within the implementation of that approach: 1) differences in systems of types and operations between the OCL and the XPath, 2) validation of XML documents by means of the external services and data sources, and 3) the completeness of the OCL specification support.

The authors have listed several directions for the development of tools for validation of XML documents based on the OCL: a formal proof of the correctness of the OCL constraints transformations into XPath assertions and a formal proof of the correctness of the OCL constraints application to the object model of XML document; the development of an universal platform-independent data type system, which generalizes type systems of XSD, SQL, etc., and the usage of host languages for XPath different from the Schematron (such as XSD 1.1, XSLT, Java).

Keywords: XML Semantic Validation, Object Constraint Language (OCL), Model-Driven Architecture (MDA), Platform-Independent Model (PIM)

References

1. **Nikiforov D. A., Lisikh I. G., Sivakov R. L.** An Approach to Multi-Domain Data Model Development Based on the Model-Driven Architecture and Ontologies, *Supplementary Proceedings of the 4th International Conference on Analysis of Images, Social Networks and Texts (AIST'2015)*, Yekaterinburg, Russia, April 9–11, 2015, pp. 106–117.

2. **Ambler S. W.** Mapping objects to relational databases, *Software Development*, 1995, vol. 3, pp. 63–72.

3. **Miller J., Mukerji J.** MDA Guide Version 1.0.1, *Object Management Group*, 2003, URL: <http://www.omg.org/cgi-bin/doc?omg/03-06-01> (accessed: 14.12.2016).

4. **OMG.** *UML Profile for National Information Exchange Model (NIEM)*, version 3.0, Object Management Group, 2015.

5. **ISO 20022-1:2003.** *Financial services — Universal financial industry message scheme — Part 1: Metamodel.*

6. **UN/CEFACT.** *Core Components Technical Specification, version 3.0*, United Nations Centre for Trade Facilitation and Electronic Business, 2009.

7. **OMG.** *Unified Modeling Language (UML)*, version 2.5, Object Management Group, 2015.

8. **Li L., Zhao X.** UML Specification of Relational Database, *Journal of Object Technology*, September–October, 2003, vol. 2, no. 5, pp. 87–100.

9. **Dominquez E., Lloret J., Pérez B., Rodriguez A., Rubio A. L., Zapata M. A.** A survey of UML models to XML schemas transformations, Springer—Verlag Berlin Heidelberg, 2007, pp. 184–195.

10. **Harrison W., Barton C., Raghavachari M.** Mapping UML designs to Java, ACM SIGPLAN Notices. Minneapolis, Minnesota, USA, 2000, vol. 35, pp. 178–187.

11. **Wahler M., Koehler J., Brucker A. D.** Model-Driven Constraint Engineering, *Workshop on OCL for (Meta-)Models in Multiple Application Domains at the UML/MoDELS Conference*, Genoa, Italy. 2006. pp. 111–125.

12. **Dang D. H., Cabot J.** Automating Inference of OCL Business Rules from User Scenarios // 20th Asia-Pacific Software Engineering Conference, APSEC 2013. Ratchathewi, Bangkok, Thailand. 2013. Vol. 1. pp. 156–163.

13. **Moskal J., Kokar M., Morgan J.** Semantic Validation of T&E XML Data // in Proceedings of the International Telemetering Conference (ITC 2015). Las Vegas, NV, USA. 2015.

14. **Cabot J., Teniente E.** Constraint Support in MDA Tools: A Survey // In: Model Driven Architecture — Foundations and Applications: Second European Conference, ECMDA-FA 2006, Bilbao, Spain, July 10–13, 2006, Proceedings, Berlin—Heidelberg, Berlin—Heidelberg, Springer, 2006, pp. 256–267.

15. **Demuth B., Hussmann H., Loecher S.** OCL as a Specification Language for Business Rules in Database Applications, *"UML" 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and*

Tools: 4th International Conference Toronto, Canada, October 1–5, 2001 Proceedings. Berlin (Heidelberg), Springer, 2001, pp. 104–117.

16. **Kallel S., Tibermacine C., Tramoni B., Dony C., Kacem A. H.** Automatic Translation of OCL Meta-Level Constraints into Java Meta-Programs, *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2015, Takamatsu, Japan*. Takamatsu, Japan, 2016, pp. 213–226.

17. **Vergara-Niedermayr C., Wang F., Pan T., Kurc T., Saltz J.** Semantically Interoperable XML Data, *International journal of semantic computing*, 2013, vol. 7, pp. 237–255.

18. **OMG.** *Object Constraint Language (OCL), version 2.4*, Object Management Group, 2014.

19. **Nikiforov D. A.** The Object Constraint Language, *Habrahabr*, 2015. URL: <https://habrahabr.ru/company/cit/blog/264963/> (accessed: 14.12.2016).

20. **Gaafar A., Sakr S.** Towards a Framework for Mapping Between UML/OCL and XML/XQuery, "UML" 2004 — *The Unified Modeling Language. Modeling Languages and Applications: 7th International Conference, Lisbon, Portugal, October 11–15, 2004. Proceedings*. Berlin (Heidelberg): Springer, 2004, pp. 241–259.

21. **ShapeChange** — *Processing application schemas for geographic information* [Online] URL: <http://shapechange.net> (accessed: 14.12.2016).

22. **ISO 19109:2015**, *Geographic information — Rules for application schema*.

23. **ISO/IEC 19757-3:2016**. *Information technology — Document Schema Definition Languages (DSDL), Part 3: Rule-based validation — Schematron*.

24. **OCL Conversion to Schematron** [Online], *ShapeChange: [Web site]*, URL: <http://shapechange.net/targets/xsd/extensions/ocl/> (accessed: 14.12.2016).

25. **OMG.** *XML Metadata Interchange (XMI), version 2.5.1*, Object Management Group, 2015.

26. **PIM 2 Schematron Transformation** [Online], *NIEM-UML: [Web site]*, 2015, URL: <https://github.com/NIEM/NIEM-UML/blob/master/Archive/Phase1/Specification/NIEM%20UML%20Machine%20Readable%20Files/NIEMpim2Schematron.qvto> (accessed: 14.12.2016).

27. **OMG.** *Meta Object Facility (MOF) 2.0 Query/View/Transformation, version 1.2*, Object Management Group, 2015.

28. **Gronback R. C.** *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*. 1st ed. Addison-Wesley Professional, 2009.

29. **Schematron** Metamodel [Online], *NIEM-UML, [Web site]*, 2015, URL: <https://github.com/NIEM/NIEM-UML/blob/master/Archive/Phase1/Specification/NIEM%20UML%20Machine%20Readable%20Files/Nonnormative/schematron.ecore> (accessed: 14.12.2016).

30. **Klmeck J., Malý J., Nečáský M., Holubová I.** eXolutio: Methodology for Design and Evolution of XML Schemas Using Conceptual Modeling, *Informatica, Lith. Acad. Sci.*, 2015, vol. 26, pp. 453–472.

31. **Malý J., Nečáský M.** Evaluation of OCL Expressions over XML Data Model, *Journal of Universal Computer Science*, mar 2014, vol. 20, pp. 329–365.

32. **Malý J.** *XML Document Adaptation and Integrity Constraints in XML*. Doctoral Thesis. Prague. 2013.

33. **Demuth B.** The Dresden OCL toolkit and its role in Information Systems development, *In 13th International Conference on Information Systems Development: Methods and Tools, Theory and Practice*. Vilnius, Lithuania, 2004, pp. 1–12.

34. **Steinberg D., Budinsky F., Paternostro M., Merks E.** *EMF: Eclipse Modeling Framework 2.0*, 2nd ed. Addison-Wesley Professional, 2009.

35. **Garry D., Balfe T.** Experiences Using OCL for Business Rules on Financial Messaging, *Proceedings of the 12th Workshop on OCL and Textual Modelling, New York, NY, USA*, 2012, pp. 65–66.

36. **RuleX** Operations Reference [Online], *Nomos Software, [Web site]*, URL: <http://clients.nomos-software.com/RuleXOperations-Reference.html> (accessed: 15.12.2016).

37. **RuleX** ISO 20022 Operations, *Nomos Software: [Web site]*. URL: <http://clients.nomos-software.com/RuleXISO20022Operations.html> (accessed: 14.12.2016).

38. **Chiorean D., Bortes M., Corutiu D.** Object Constraint Language Environment, a Tool Supporting Teaching and Learning UML and OCL, the Understanding and Using of Metamodeling, Abstraction and Design by Contract, in *Proceedings of Eight Workshop on Pedagogies and Tools for Teaching and Learning Object Oriented Concepts*. 2000.

39. **Using OCLE** to ensure the syntactic and semantic validity of XML documents, *Object Constraint Language Environment: [Web site]*, URL: <http://ci.cs.ubbcluj.ro/ocle/um/xmlvalidation.htm> (accessed: 14.12.2016).

УДК 004.9

Р. М. Алгулиев, д-р техн. наук, проф., директор, e-mail: rasim@science.az,

Г. Ч. Набибекова, зав.отделом, e-mail: gulnarara58@mail.ru,

Институт информационных технологий НАНА, г. Баку, Азербайджан

Методы определения степени интеграции стран в среде электронного государства

В качестве индикаторов определения степени интеграции стран предлагаются: объем информационных потоков между национальными доменами стран в Интернете, число международных встреч сотрудников государственных организаций и число международных договоров, заключенных между странами. Предложены методы расчета весовых коэффициентов индикаторов, а также методы определения степени интеграции стран, при этом рассматривается двухуровневая иерархия индикаторов.

Ключевые слова: интеграция стран, индикаторы интеграции, национальный домен верхнего уровня, международный договор, теория графов, интегральный показатель, весовые коэффициенты, метод ранжирования, метод приписывания баллов

Введение

Сегодня в условиях глобализации между странами в различных сферах осуществляются интеграционные процессы. В зависимости от контекста различаются экономическая, политическая и со-

циальная интеграции. Но в условиях глобализации актуальным является исследование интеграционных процессов между странами в среде электронного государства, выявление индикаторов интеграции, определение степени интеграции стран именно в

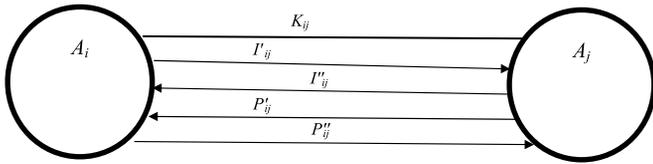


Рис. 2. Связи между двумя выбранными странами A_i и A_j

страны, а M — множество дуг, соединяющих эти вершины.

Граф, изображенный на рис. 2, демонстрирует связи между некоторыми двумя странами A_i и A_j .

Граф на рис. 2 является смешанным, поскольку ребра P'_{ij} и P''_{ij} , соответствующие международным встречам между представителями различных стран, являются ориентированными, т. е. дугами; ребра I'_{ij} и I''_{ij} , соответствующие объемам информационных потоков, обмениваемых странами (импорт и экспорт информации), также являются ориентированными, т. е. дугами; пятое ребро K_{ij} , соответствующее заключенным международным договорам, является неориентированным.

Граф является взвешенным, поскольку каждое ребро имеет вес. Вес ребра, относящегося к заключенным международным договорам, определяется их числом и типами; вес ребер, относящихся к международным встречам, также определяется их числом и типами; вес ребер, относящихся к информационным потокам между странами, определяется объемом информации, которой обмениваются эти страны.

Описанные выше три критерия (показателя), которые мы рассматриваем для определения интеграции, имеют разную значимость и, следовательно, по-разному влияют на интеграцию.

Для количественного представления значимости показателей будем использовать весовые коэффициенты. *Весовой коэффициент* — это числовой коэффициент, отражающий значимость, относительную важность показателя в сравнении с другими факторами, оказывающими влияние на изучаемый процесс [9].

Данные по показателям, используемые для определения степени интеграции стран, берутся за период времени, интересующий лиц, принимающих решение. Это может быть единица времени — месяц, год и т. д., а также более длительный период времени — 2 года, 3 года и т. д.

Определим степень интеграции i -й и j -й стран как интегральный показатель интеграции W_{ij} . Для его получения нужно найти взвешенную сумму индикаторов:

$$W_{ij} = \alpha_K w_{ij}^K + \alpha_P w_{ij}^P + \alpha_I w_{ij}^I, \quad (1)$$

где w_{ij}^K — значение индикатора "контракты" (международные договоры, заключенные между i -й и

j -й странами); w_{ij}^P — значение индикатора "контакты" (встречи между официальными лицами i -й и j -й стран); w_{ij}^I — значение индикатора "информация" (объем информации, которой обмениваются i -я и j -я страны в Интернете); α_K — весовой коэффициент индикатора "контракты"; α_P — весовой коэффициент индикатора "контакты"; α_I — весовой коэффициент индикатора "информация".

Внутри критериев "контакты" и "контракты" также есть критерии, которые соответствуют их типам, т. е. в данном случае мы имеем дело с иерархией критериев.

Назовем критерии "контакты", "контракты" и "информация" критериями 1-го уровня, а критерии внутри критериев "контакты" (например, саммит, форум, конференция и т. д.) и "контракты" (например, договор, соглашение, контракт и т. д.) — критериями 2-го уровня, или подкритериями. Каждый тип (подкритерий) критериев "контакты" и "контракты" имеет свою степень важности, поэтому их веса различны. В этом случае возникает необходимость объективной оценки важности подкритериев, которые учитываются при расчете весовых коэффициентов, а именно весовые коэффициенты должны качественно отражать важность соответствующих подкритериев.

Для определения весовых коэффициентов всех критериев данной иерархии, исключая показатель "информация", рассмотрим два метода: метод ранжирования и метод приписывания баллов [10—12]. Определение весовых коэффициентов и весов показателя "информация" будет рассматриваться отдельно.

Метод ранжирования для расчета весовых коэффициентов в целях определения степени интеграции стран

При использовании метода ранжирования рассмотрим случай, когда ранг каждого из трех критериев, а также ранги подкритериев критерия "контакты" (саммит, форум, конференция и т. д.) и критерия "контракты" (договор, соглашение, контракт и т. д.) единогласным решением устанавливаются группой экспертов, являющихся квалифицированными специалистами в этой области.

Для определения весовых коэффициентов как критериев, так и подкритериев используем метод ранжирования.

Пусть группа экспертов установила для имеющихся у нас трех критериев 1-го уровня следующие ранги в соответствии со степенью их важности:

- ранг критерия "контракты" соответствует 3 ($r_K = 3$);
- ранг критерия "контакты" соответствует 2 ($r_P = 2$);
- ранг критерия "информация" соответствует 1 ($r_I = 1$).

Весовой коэффициент важности критерия вычисляется по формуле

$$\alpha_i = \frac{r_i}{\sum_{i=1}^m r_i}, \quad (2)$$

где α_i — весовой коэффициент i -го критерия; r_i — ранг i -го критерия; m — общее число критериев.

Тогда для нашего случая, где $m = 3$ и $i \in \{K, P, I\}$, весовые коэффициенты критериев "контракты", "контакты" и "информация" соответственно будут определяться выражениями:

$$\alpha_K = \frac{r_K}{r_K + r_P + r_I} = \frac{3}{3 + 2 + 1} = \frac{1}{2};$$

$$\alpha_P = \frac{r_P}{r_K + r_P + r_I} = \frac{2}{3 + 2 + 1} = \frac{1}{3};$$

$$\alpha_I = \frac{r_I}{r_K + r_P + r_I} = \frac{1}{3 + 2 + 1} = \frac{1}{6}.$$

Таким образом, $\alpha_K = \frac{1}{2}$, $\alpha_P = \frac{1}{3}$, $\alpha_I = \frac{1}{6}$.

Формула (2) позволяет определить все критерии таким образом, что для их весовых коэффициентов α_i , $i = \overline{1, m}$, выполняется классическое балансовое условие:

$$\sum_{i=1}^m \alpha_i = 1. \quad (3)$$

Очевидно, что в нашем случае это условие также выполняется:

$$\alpha_K + \alpha_P + \alpha_I = \frac{1}{2} + \frac{1}{3} + \frac{1}{6} = 1.$$

Далее будем определять веса критериев "контракты" и "контакты".

Для определения веса критерия "контракты", поскольку он имеет 2-й уровень иерархии в виде типов (подкритериев), сначала определим весовые коэффициенты для этих подкритериев. Также будем использовать метод ранжирования.

Пусть имеется s типов контрактов между странами i и j и пусть группой экспертов каждому типу контракта присвоен ранг в диапазоне от 1 до s . Типы контрактов отсортируем так, что 1-й из них имел бы наименьший ранг, равный 1, последний — наибольший, равный s , т. е. так, чтобы номер типа контракта совпадал бы с его рангом. Тогда весовой

коэффициент каждого типа контракта между этими странами w_{ijb}^K будет определяться формулой

$$w_{ijb}^K = \frac{b}{\sum_{b=1}^s b}, \quad (4)$$

где $b = \overline{1, s}$.

Далее, зная весовой коэффициент каждого типа контракта, заключенного между i -й и j -й странами, находим значение индикатора "контракты", относящегося к этим странам, по формуле

$$w_{ij}^K = \sum_{b=1}^s w_{ijb}^K K_{ijb}, \quad (5)$$

где K_{ijb} — число контрактов b -го типа (каждого типа), заключенных между i -й и j -й странами.

Аналогичным образом находим значение индикатора "контакты":

$$w_{ij}^P = \sum_{b=1}^{s_1} w_{ijb}^P P_{ijb}, \quad (6)$$

где s_1 — число типов контактов.

Отметим, что число контактов P_{ijb} представляет собой сумму въездов из i -й страны в j -ю (P'_{ijb}) и из j -й страны в i -ю (P''_{ijb}), т. е.

$$P_{ijb} = P'_{ijb} + P''_{ijb}. \quad (7)$$

Полученные $\alpha_K = \frac{1}{2}$, $\alpha_P = \frac{1}{3}$, $\alpha_I = \frac{1}{6}$, а также

w_{ij}^K и w_{ij}^P подставляем в формулу (1) для нахождения интегрального показателя интеграции W_{ij} .

Метод приписывания баллов для расчета весовых коэффициентов в целях определения степени интеграции стран

Рассмотрим случай, когда при использовании метода приписывания баллов каждый эксперт из группы по-своему оценивает важность каждого из трех критериев 1-го уровня, а также каждого из подкритериев критериев "контакты" (саммит, форум, конференция и т. д.) и "контракты" (договор, соглашение, контракт и т. д.).

В этом случае учитывается не только число критериев и подкритериев, но также и число экспертов, участвующих в группе.

Метод приписывания баллов (рангов) базируется на том, что каждый эксперт оценивает важность каждого критерия по шкале [0—10]. При этом разрешается оценивать важность критерия дробными величинами, а также приписывать одну и ту же величину из выбранной шкалы нескольким критериям.

Сначала рассмотрим критерии 1-го уровня — "контракты", "контакты" и "информация". Пусть

этим трем критериям в вышеуказанном порядке приписывают баллы t экспертов, т. е. h_{11} — балл, приписанный l -м экспертом критерию "контракты", h_{12} — балл, приписанный l -м экспертом критерию "контакты", h_{13} — балл, приписанный l -м экспертом критерию "информация". То есть в общем виде h_{lk} — балл l -го эксперта для k -го критерия, где $l = \overline{1, t}$, $k = \overline{1, 3}$.

Построим матрицу оценок:

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \\ \dots & \dots & \dots \\ h_{t1} & h_{t2} & h_{t3} \end{pmatrix}. \quad (8)$$

Тогда вес, подсчитанный для k -го критерия l -м экспертом, будет

$$r_{lk} = \frac{h_{lk}}{\sum_{k=1}^3 h_{lk}}, \quad (9)$$

где $l = \overline{1, t}$, $k = \overline{1, 3}$, $\sum_{k=1}^3 h_{lk}$ — сумма l -й строки.

Построим матрицу весов, подсчитанных для трех критериев t экспертами:

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \\ \dots & \dots & \dots \\ r_{t1} & r_{t2} & r_{t3} \end{pmatrix}. \quad (10)$$

Найдем сумму k -го столбца r_k , т. е. сумму весов, подсчитанных всеми t экспертами для каждого из трех критериев:

$$r_k = \sum_{l=1}^t r_{lk}, \quad (11)$$

где $k = \overline{1, 3}$.

После этого получим весовые коэффициенты для критериев 1-го уровня — "контракты", "контакты" и "информация" по формуле

$$\lambda_k = \frac{r_k}{\sum_{k=1}^3 r_k},$$

где $k = \overline{1, 3}$.

Тогда имеем $\alpha_K = \lambda_1$, $\alpha_P = \lambda_2$, $\alpha_I = \lambda_3$.

Теперь рассмотрим критерии 2-го уровня (подкритерии) для одного из критериев 1-го уровня, например для критерия "контракты", также используя метод приписывания баллов.

Пусть имеется s типов контрактов, т. е. s подкритериев. Матрица оценок (8) теперь выглядит следующим образом:

$$\begin{pmatrix} h_{11} & h_{12} & \dots & h_{1s} \\ h_{21} & h_{22} & \dots & h_{2s} \\ h_{31} & h_{32} & \dots & h_{3s} \\ \dots & \dots & \dots & \dots \\ h_{t1} & h_{t2} & \dots & h_{ts} \end{pmatrix}, \quad (12)$$

где h_{lp} — балл l -го эксперта для p -го подкритерия, где $l = \overline{1, t}$, $p = \overline{1, s}$.

Тогда вес, подсчитанный для p -го критерия l -м экспертом по аналогии с (9) будет:

$$r_{lp} = \frac{h_{lp}}{\sum_{p=1}^s h_{lp}}, \quad (13)$$

где $l = \overline{1, t}$, $p = \overline{1, s}$, $\sum_{p=1}^s h_{lp}$ — сумма l -й строки.

По аналогии с (10) построим матрицу весов, подсчитанных уже для s критериев t экспертами:

$$\begin{pmatrix} r_{11} & r_{12} & \dots & r_{1s} \\ r_{21} & r_{22} & \dots & r_{2s} \\ r_{31} & r_{32} & \dots & r_{3s} \\ \dots & \dots & \dots & \dots \\ r_{t1} & r_{t2} & \dots & r_{ts} \end{pmatrix}. \quad (14)$$

Найдем сумму p -го столбца r_p , т. е. сумму весов, подсчитанных всеми t экспертами для каждого из s критериев:

$$r_p = \sum_{l=1}^t r_{lp}, \quad (15)$$

где $p = \overline{1, s}$.

Находим весовые коэффициенты для каждого из s критериев 2-го уровня (подкритериев) критерия "контракты", т. е. для s типов контрактов, заключенных между странами i и j :

$$w_{ijp}^K = \frac{r_p}{\sum_{p=1}^s r_p}, \quad (16)$$

где $p = \overline{1, s}$.

Находим вес показателя "контракты" для стран i и j :

$$w_{ij}^K = \sum_{p=1}^s w_{ijp}^K K_{ijp}, \quad (17)$$

где K_{ijp} — число контрактов p -го типа для стран i и j .

Аналогичным образом находим вес показателя "контакты" для стран i и j :

$$w_{ij}^P = \sum_{p=1}^{s_1} w_{ijp}^P P_{ijp}, \quad (18)$$

где s_1 — число типов контактов.

Отметим, что и в этом случае P_{ijp} определяется по формуле (7).

Таким образом, для получения интегрального показателя W_{ij} интеграции, т. е. степени интеграции стран i и j , согласно (1) осталось найти w_{ij}^I — вес критерия "информация" (объем информации, обмениваемой между странами i и j в Интернете).

Вес критерия "информация" для стран i и j принимаем равным v_{ij} , где v_{ij} — объем информации, обмениваемой между i -й и j -й странами, т. е.

$$w_{ij}^I = v_{ij}. \quad (19)$$

Затем значение w_{ij}^I подставляем в формулу (1) и получаем интегральный показатель степени интеграции между i -й и j -й странами.

Таким образом, после нахождения весов всех критериев w_{ij}^K , w_{ij}^P и w_{ij}^I , а также весовых коэффициентов каждого критерия α_K , α_P , α_I , используя формулу (1), находим интегральный показатель W_{ij} , т. е. степень интеграции i -й страны с j -й.

Отметим, что можно также определить степень интеграции стран по различным областям деятельности в отдельности: научной, образовательной, экономической, медицинской, социальной, военной, культурной и др. В этом случае формула (1) примет вид:

$$W_{ij, dom} = \alpha_{K, dom} w_{ij, dom}^K + \alpha_{P, dom} w_{ij, dom}^P + \alpha_{I, dom} w_{ij, dom}^I. \quad (20)$$

Здесь к интегральному показателю степени интеграции W_{ij} и к остальным параметрам добавлен индекс dom (*domain* — сфера деятельности (англ.)), т. е. указывается конкретная сфера деятельности, и для вычисления интегрального показателя используются данные, относящиеся к этой сфере деятельности.

Полезным является вычисление степени интеграции стран периодически каждый год. Это позволяет проследить динамику изменения интеграции одной страны в другую по годам. При этом учитывается число контактов и объем обмениваемой информацией за отчетный год, а также число контрактов, имеющих силу в отчетном году.

Также можно вычислить интеграцию страны в международную организацию, например, ЕС, ОПЕС, СЧЭС и т. д. Будем иметь:

$$W_{org} = \sum_{j=1}^l W_j, \quad (21)$$

где W_{org} — интеграция страны в какую-либо международную организацию, включающую l стран, а W_j — интеграция страны в каждую из l стран, входящих в эту организацию.

Заключение

Преимущества интеграции очевидны. Она способствует ускорению развития стран, реализации крупных инновационных проектов, пополнению бюджета, развитию инфраструктуры и т. д. Кроме того, для развивающихся между странами интеграционных процессов в условиях глобализации характерны: углубление международного разделения труда, интернационализация капитала, повышение свободы торговли, глобализация научно-технического прогресса и т. д. Интеграция также способствует созданию в различных регионах мира всевозможных интеграционных объединений, которые впоследствии могут трансформироваться в экономические, политические и военные союзы, имеющие большое значение на мировой арене.

Таким образом, для видения изнутри всех этих процессов, их анализа, прогнозирования, выявления тенденций их дальнейшего развития большое значение приобретают индикаторы, определяющие степень интеграции стран.

Список литературы

1. **Prakash A., Jeffrey A.** Indicators of Economic Integration // *Global Governance*, 2000, Vol. 6, N. 1. P. 95—114. URL: http://www.jstor.org/stable/27800255?seq=1#page_scan_tab_contents
2. **Arribas I., Perez F., Tortosa-Ausina E.** Measuring Globalization of International Trade: Theory and Evidence. URL: <http://www.sciencedirect.com/science/article/pii/S0305750X08001447>
3. **De Lombaerde P., Dorrucci E., Genna G., Mongelli F. P.** Composite indexes and systems of indicators of regional integration. *The Regional Integration Manual: Quantitative and Qualitative Methods* / Ed. by De Lombaerde P., Flôres R. G., Iapadre P. L., Schulz M. Routledge / Warwick studies in globalization. 2011. P. 323—346.
4. **Система индикаторов евразийской интеграции ЕАБР 2009.** Алматы: RUAN, 2010. 160 с.
5. **Международный договор.** URL: ru.wikipedia.org/wiki/Международный_договор
6. **Alguliyev R., Aliguliyev R., Nabibayova G.** The Method of Measuring the Integration Degree of Countries on the Basis of International Relations // *Intelligent Systems and Applications*. 2015. N. 11. P. 10—18.
7. **Набибекова Г.** Об одном подходе к измерению интеграционных процессов между странами // *Телекоммуникации*. 2015. № 3. С. 44—48.
8. **Ruohonen K.** Graph theory. URL: http://math.tut.fi/~ruohonen/GT_English.pdf
9. **Райзберг Б. А., Лозовский Л. Ю., Стародубцева Е. Б.** Современный экономический словарь. 6-е изд., перераб. и доп. М.:

ИНФРА-М, 2011. URL: http://www.consultant.ru/document/cons_doc_LAW_67315/

10. **Ларичев О. И.** Теория и методы принятия решений, а также Хроника событий в Волшебных странах: учебник. М.: Университетская книга, 2008. 392 с.

11. **Саати Т. Л.** Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1993. 320 с.

12. **Сəbrayılova Z. Q., Nobari S. M.** Personalın idarə olunması məsələlərində kriteriyaların vacibliyi haqqında informasiyanın emalı üsulları və ziddiyyətin aşkarlanması // *Informasiya texnologiyaları problemləri*, 2011. N. 2. P. 57–66. URL: www.jpit.az/storage/files/article/8a78f78a95641546667c863d448bfa7d.pdf

R. M. Alguliyev, Dr. of Tech. Sci., Prof., Director, e-mail: rasim@science.az,

G. Ch. Nabibayova, Head of Department, e-mail: gulnarara58@mail.ru

Institute of Information Technology of the Azerbaijan National Academy of Sciences, Baku, Azerbaijan

Methods for Determination of the Degree of Integration of Countries in e-Government Environment

In the paper, the necessity to study the integration processes between countries in e-government environment in the context of globalization is shown. To this end, the volume of information flows between countries' national domains on the Internet, the number of international meetings of employees of state organizations and the number of international agreements concluded between countries are offered as the indicators for determination of the degree of integration of countries. It is noted that these three criteria (indicators) we have chosen for our research have different importance, and therefore they have different impact on the integration. Therefore, we use weight coefficients to quantify them.

It is also noted that two of these criteria, namely: the international meetings of employees of state organizations and the international agreements concluded between the countries have sub-criteria that correspond to their types. Thus, we considered a two-level hierarchy of criteria (indicators) to solve this problem.

To calculate the weight coefficients of the indicators, a ranking method and an expert evaluation method are proposed. Using the ranking method, we consider a case where the rank of each of the three 1st level criteria, as well as ranks of 2nd level criteria are determined unambiguously by a group of experts in this field. Using the expert evaluation method, we considered a case where each expert from the group gives his/her own rank to each of the three 1st level criteria, as well as the 2nd level criteria.

Keywords: integration of countries, indicators of the integration of countries, country code Top-Level Domain (ccTLD), an international treaty, graph theory, integral indicator, weight coefficients, ranking method, expert evaluation method

References

1. **Prakash A., Jeffrey A.** Indicators of Economic Integration, *Global Governance*, 2000, vol. 6, no. 1, pp. 95–114, URL: http://www.jstor.org/stable/27800255?seq=1#page_scan_tab_contents

2. **Arribas I., Perez F., Tortosa-Ausina E.** *Measuring Globalization of International Trade: Theory and Evidence*. URL: <http://www.sciencedirect.com/science/article/pii/S0305750X08001447>

3. **De Lombaerde P., Dorrucchi E., Genna G., Mongelli F. P.** *Composite indexes and systems of indicators of regional integration. The Regional Integration Manual: Quantitative and Qualitative Methods*. Ed. by De Lombaerde P., Flôres R. G., Iapadre P. L., Schulz M. Routledge / Warwick studies in globalization, 2011, pp. 323–346.

4. **Sistema indikatorov evrazijskoj integracii EABR 2009** (The system of indicators Eurasian Integration 2009). Almaty: RUAN, 2010. 160 p. (in Russian).

5. **Mezhdunarodnyj dogovor** (International treaty). URL: ru.wikipedia.org/wiki/Международный_договор (in Russian).

6. **Alguliyev R., Aliguliyev R., Nabibayova G.** The Method of Measuring the Integration Degree of Countries on the Basis of International Relations, *Intelligent Systems and Applications*, 2015, no. 11, pp. 10–18.

7. **Nabibekova G.** Ob odnom podhode k izmereniju integracionnyh processov mezhdru stranami (One approach to the measurement of the integration processes between countries), *Telekommunikacii*, 2015, no. 3, pp. 44–48. (in Russian).

8. **Ruohonen K.** *Graph theory*. URL: http://math.tut.fi/~ruohonen/GT_English.pdf

9. **Rajzberg B. A., Lozovskij L. Ju., Starodubceva E. B.** *Sovremennij jekonomicheskij slovar'* (Modern Dictionary of Economics). 6-e izd., pererab. i dop. Moscow, INFRA-M, 2011. URL: http://www.consultant.ru/document/cons_doc_LAW_67315/ (in Russian).

10. **Larichev O. I.** *Teorija i metody prinjatija reshenij, a takzhe Hronika sobytij v Volshebnyh stranah: Uchebnik*. (Theory and methods of decision-making, as well as a chronicle of events in a magical land: Tutorial) Moscow, Universitetskaja kniga, 2008, 392 p. (in Russian).

11. **Saati T. L.** *Prinjatje reshenij. Metod analiza ierarhij*. (Decision making. The analytic hierarchy process). Moscow, Radio i svjaz', 1993. 320 p. (in Russian).

12. **Сəbrayılova Z. Q., Nobari S. M.** Personalın idarə olunması məsələlərində kriteriyaların vacibliyi haqqında informasiyanın emalı üsulları və ziddiyyətin aşkarlanması, *Informasiya texnologiyaları problemləri*, 2011, no. 2, pp. 57–66. URL: www.jpit.az/storage/files/article/8a78f78a95641546667c863d448bfa7d.pdf

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 62-50; 519.7; 519.8

В. И. Левин, д-р техн. наук, проф., e-mail: vilevin@mail.ru,
Пензенский государственный технологический университет

Моделирование систем с неопределенностью с помощью полиинтервальной математики

В последние десятилетия в военной и гражданской сферах все чаще встречаются новые технологии, связанные с изучением неопределенности. Эти технологии широко применяют в технике, экономике, социальной сфере. Для их поддержки нужны новые математические модели и методы. В связи с этим данная работа, посвященная разработке новой модели неопределенности (полиинтервал) и математических методов ее изучения, является актуальной. Цель работы заключается в детальной разработке новой математической модели неопределенности — полиинтервала, являющегося последовательностью конечного числа интервалов неопределенности, системы алгебраических операций над полиинтервалами и правил выполнения этих операций. Предложено распространить на изучение полиинтервалов известный в интервальной математике метод изучения интервалов, основанный на определении алгебраических операций над интервалами в виде теоретико-множественных обобщений соответствующих операций над вещественными числами. Новизна работы заключается в предложенной новой математической модели неопределенности систем в виде полиинтервалов, совместно с математическим аппаратом, позволяющим выполнять различные операции над полиинтервалами и тем самым дающим возможность выполнять математическое моделирование систем с неопределенностью. Детально разработана новая математическая модель неопределенности — полиинтервал. Определена система алгебраических операций над полиинтервалами и выведены правила их выполнения. Предложен алгоритм изучения систем с полиинтервальными параметрами.

Ключевые слова: интервал, полиинтервал, неопределенность, алгебра полиинтервалов, моделирование систем

Введение

Известно, что в период Второй Мировой войны в практику ведения военных действий в западных странах (США, Великобритании, Канады) было введено множество новых технологий: обнаружение воздушных целей с помощью радаров, управление огнем зенитной артиллерии, шифровка и дешифровка информации, атомное оружие и т. д. Все эти технологии в той или иной степени были связаны с изучением неопределенности, присущей любым военным действиям, и использовали соответствующие математические методы, в первую очередь, теорию вероятностей. После войны эти работы были продолжены и распространены на гражданскую сферу — экономику, технику, социум. При этом расширилось само понимание неопределенности, в которую теперь стали включать не только случайность возможных исходов, но и их неединственность или их незнание, дрейф переменных, семантическую неопределенность целей, многокритериальность при принятии решений, недоопределенность модели или структуры изучаемой системы и т. д. Новые подходы к описанию неопределенности изучаемых систем привели к появлению новых математических методов для их

изучения: теория нечетких множеств, многозначная логика, теория сверхслучайных процессов и др. Одним из наиболее популярных методов стала интервальная математика, занимающаяся изучением величин, определяемых с точностью до интервалов возможных значений [1, 2]. Но одиночные интервалы, являющиеся объектом изучения в интервальной математике, не охватывают всех ситуаций, встречающихся на практике. Например, неопределенный период времени, в течение которого возможно успешное проведение некоторой операции, может включать несколько последовательных временных интервалов, скажем $([4^{00}, 5^{30}], [21^{00}, 23^{00}], [24^{00}, 2^{00}])$. Аналогично, участок пространства, в рамках которого возможно наблюдение некоторых объектов, может включать в себя несколько последовательных угловых интервалов, например $([15^\circ, 21^\circ], [28^\circ, 35^\circ], [48^\circ, 53^\circ])$. Очевидным образом, на практике будут появляться и другие подобные примеры. Во всех таких примерах мы сталкиваемся с новыми неопределенными объектами, которые имеют вид последовательностей интервалов неопределенности. Каждый такой объект естественно назвать полиинтервалом. Настоящая работа полностью посвящена теории и возможным применениям полиинтервалов.

Постановка задачи

Распространенный подход к изучению неопределенных систем, известный под названием интервальной математики [1, 2], строится на базе понятия интервала, трактуемого как множество всех возможных значений неполностью определенной величины \tilde{a} , задаваемой лишь ее нижней a_1 и верхней a_2 границами. Величину \tilde{a} можно записать в виде следующего ограниченного интервала неопределенности:

$$\tilde{a} \equiv [a_1, a_2] = \{a | a_1 \leq a \leq a_2\}. \quad (1)$$

Здесь предполагается, что неизвестное "истинное" значение неопределенной величины \tilde{a} достоверно лежит в пределах интервала $[a_1, a_2]$, не выходя за его границы a_1 и a_2 . Причем все значения в пределах этого интервала считаются "равновозможными" в том смысле, что нет никаких оснований предпочитать одно значение другому. Заметим, что в данном случае понятие равновозможности не означает задание равномерного вероятностного или какого-либо иного распределения возможных значений внутри указанного интервала. Над интервалами вида (1) вводятся алгебраические операции, аналогичные соответствующим операциям над числами. Для этого используется теоретико-множественная конструкция

$$\tilde{a} \circ \tilde{b} = \{a \bullet b | a \in \tilde{a}, b \in \tilde{b}\}, \circ \tilde{a} = \{a | a \in \tilde{a}\}, \quad (2)$$

т. е. любая операция над интервалами \circ определяется на основе соответствующей операции над точными величинами \bullet , при условии, что конкретные значения этих величин пробегают все возможные значения из соответствующих интервалов. Из этого определения вытекают простые правила выполнения операций над интервалами:

$$\begin{aligned} [a_1, a_2] + [b_1, b_2] &= [a_1 + b_1, a_2 + b_2]; \\ [a_1, a_2] - [b_1, b_2] &= [a_1 - b_2, a_2 - b_1]; \\ k \cdot [a_1, a_2] &= \begin{cases} [ka_1, ka_2], & k > 0, \\ [ka_2, ka_1], & k < 0; \end{cases} \\ [a_1, a_2] \cdot [b_1, b_2] &= [\min_{i,j} (a_i \cdot b_j), \max_{i,j} (a_i \cdot b_j)]; \\ [a_1, a_2] / [b_1, b_2] &= [a_1, a_2] \cdot [1/b_2, 1/b_1], \\ &\text{при } 0 \notin [b_1, b_2]. \end{aligned} \quad (3)$$

Продолжим развитие интервальной математики и введем понятие полиинтервала как последовательности нескольких одиночных интервалов неопределенности:

$$\tilde{A} = (\tilde{a}, \tilde{b}, \dots, \tilde{d}), \quad (4)$$

где $\tilde{a}, \tilde{b}, \dots, \tilde{d}$ — одиночные интервалы вида (1).

Операции над полиинтервалами введем аналогично операциям над интервалами, т. е. с помощью теоретико-множественной конструкции типа (2):

$$\tilde{A} \circ \tilde{B} = \{a \bullet b | a \in \tilde{A}, b \in \tilde{B}\}, \circ \tilde{A} = \{a | a \in \tilde{A}\}. \quad (5)$$

Здесь \tilde{A} — полиинтервал вида (4); \tilde{B} — другой полиинтервал того же вида, но с другими составляющими его одиночными интервалами вида (1). Задача заключается в том, чтобы на базе определения (5) операций над полиинтервалами вывести правила выполнения указанных выше операций, аналогичные правилам (3) выполнения операций над интервалами.

Математический аппарат

Будем представлять полиинтервалы вида (4) в теоретико-множественных терминах таким образом:

$$\tilde{A} = \tilde{a} \cup \tilde{b} \cup \dots \cup \tilde{d}. \quad (6)$$

Пусть заданы два полиинтервала \tilde{A} и \tilde{B} следующего вида:

$$\tilde{A} = \bigcup_{i=1}^m \tilde{a}^i; \tilde{B} = \bigcup_{j=1}^n \tilde{b}^j, \quad (7)$$

где $\tilde{a}^i = [a_1^i, a_2^i]$, $i = \overline{1, m}$ и $\tilde{b}^j = [b_1^j, b_2^j]$, $j = \overline{1, n}$

— одиночные интервалы, составляющие \tilde{A} и \tilde{B} соответственно. Требуется выполнить операцию \circ над этими полиинтервалами. Согласно определению (5) имеем с учетом вида (7) полиинтервалов \tilde{A} и \tilde{B}

$$\tilde{A} \circ \tilde{B} = \left\{ a \bullet b \mid a \in \bigcup_{i=1}^m \tilde{a}^i, b \in \bigcup_{j=1}^n \tilde{b}^j \right\}. \quad (8)$$

На основании ассоциативного закона алгебры множеств выражение (8) можно представить как

$$\tilde{A} \circ \tilde{B} = \bigcup_{i=1}^m \bigcup_{j=1}^n \{a \bullet b \mid a \in \tilde{a}^i, b \in \tilde{b}^j\}. \quad (9)$$

Однако согласно определению (2) выражение в фигурных скобках формулы (9) равно $\tilde{a}^i \circ \tilde{b}^j$. Так что окончательно получаем

$$\begin{aligned} \tilde{A} \circ \tilde{B} &= \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \circ \tilde{b}^j) \text{ или в развернутом виде} \\ &\left(\bigcup_{i=1}^m \tilde{a}^i \right) \circ \left(\bigcup_{j=1}^n \tilde{b}^j \right) = \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \circ \tilde{b}^j). \end{aligned} \quad (10)$$

Выражение для операции \circ над одним полиинтервалом \tilde{A} вида (7) имеет вид, аналогичный (10):

$$\begin{aligned} \circ \tilde{A} &= \bigcup_{i=1}^m (\circ \tilde{a}^i) \text{ или в развернутом виде} \\ &\circ \left(\bigcup_{i=1}^m \tilde{a}^i \right) = \bigcup_{i=1}^m (\circ \tilde{a}^i). \end{aligned} \quad (11)$$

Формулы (10) и (11) сводят выполнение операций над полиинтервалами к выполнению тех же самых операций над одиночными интервалами. Поскольку для последних имеются формулы их конструктивного выполнения (3), то путем совместного применения формул (3), (10), (11) решается и поставленная задача конструктивного выполнения различных операций над полиинтервалами.

Решение задачи

Установим сначала вид выражения для конструктивного выполнения операции сложения полиинтервалов. Для этого подставим в исходную формулу (10) выражения сумм интервалов согласно формулам (3) и учтем, что в этом случае операция \circ есть сложение $+$. В результате получим искомую формулу в следующем виде:

$$\begin{aligned} \tilde{A} + \tilde{B} &= \bigcup_{i=1}^m [a_1^i, a_2^i] + \bigcup_{j=1}^n [b_1^j, b_2^j] = \\ &= \bigcup_{i=1}^m \bigcup_{j=1}^n [a_1^i + b_1^j, a_2^i + b_2^j]. \end{aligned} \quad (12)$$

Аналогично устанавливается формула для конструктивного выполнения операции вычитания полиинтервалов. Для этого подставляем в исходную формулу (10) выражения разностей интервалов согласно (3), учитывая при этом, что здесь операция \circ есть вычитание $-$. В результате мы получаем формулу

$$\begin{aligned} \tilde{A} + \tilde{B} &= \bigcup_{i=1}^m [a_1^i, a_2^i] - \bigcup_{j=1}^n [b_1^j, b_2^j] = \\ &= \bigcup_{i=1}^m \bigcup_{j=1}^n [a_1^i - b_1^j, a_2^i - b_2^j]. \end{aligned} \quad (13)$$

Формулу для конструктивного выполнения операции умножения полиинтервала на число находим аналогично. При этом используем выражение произведения интервала на число (3), а в качестве исходной используем не формулу (10), а формулу (11). В результате находим

$$k\tilde{A} = k \left(\bigcup_{i=1}^m [a_1^i, a_2^i] \right) = \begin{cases} \bigcup_{i=1}^m [ka_1^i, ka_2^i], & k > 0, \\ \bigcup_{i=1}^m [ka_2^i, ka_1^i], & k < 0. \end{cases} \quad (14)$$

Аналогично формулам (12), (13) для конструктивного выполнения операций сложения и вычитания полиинтервалов находим формулы для конструктивного выполнения операций умножения и деления полиинтервалов. При этом опираемся на правила умножения и деления интервалов (3), но в качестве исходной формулы снова используем

формулу (10). В результате получаем формулу умножения полиинтервалов в виде

$$\begin{aligned} \bigcup_{i=1}^m [a_1^i, a_2^i] \cdot \bigcup_{j=1}^n [b_1^j, b_2^j] &= \\ &= \bigcup_{i=1}^m \bigcup_{j=1}^n [a_1^i, a_2^i] \cdot [b_1^j, b_2^j] = \\ &= \bigcup_{i=1}^m \bigcup_{j=1}^n \left[\min(a_s^i b_q^j), \max(a_s^i b_q^j) \right] \end{aligned} \quad (15)$$

и формулу деления полиинтервалов в виде

$$\begin{aligned} \bigcup_{i=1}^m [a_1^i, a_2^i] / \bigcup_{j=1}^n [b_1^j, b_2^j] &= \\ &= \bigcup_{i=1}^m \bigcup_{j=1}^n [a_1^i, a_2^i] \cdot [1/b_2^j, 1/b_1^j] = \\ &= \bigcup_{i=1}^m \bigcup_{j=1}^n \left[\min(a_s^i / b_q^j), \max(a_s^i / b_q^j) \right] \end{aligned} \quad (16)$$

при $0 \notin [b_1^j, b_2^j], j = \overline{1, n}$.

Формулы (12)–(16) дают правила конструктивного выполнения всех введенных выше алгебраических операций над полиинтервалами путем сведения указанных операций к соответствующим хорошо известным операциям над одиночными интервалами.

Теперь алгоритм решения разнообразных задач, которые возникают при исследовании систем с полиинтервальными характеристиками, можно представить следующим образом.

Шаг 1. Построение математической модели, представляющей решение задачи как вычисление и анализ функции F аргументов-полиинтервалов.

Шаг 2. Составление по уже построенной модели блок-схемы алгоритма вычисления (анализа) функции F .

Шаг 3. Вычисление (анализ) по имеющейся блок-схеме алгоритма полиинтервальной функции F с использованием формул (12)–(16) выполнения различных операций над полиинтервалами. Заметим, что заключительной операцией во всех формулах является объединение интервалов, выполняемое известными методами [3].

Пример. Работник служит в двух фирмах: A и B . Причем в фирме A его месячная заработная плата в зависимости от заказов оценивается в размере $10\,000 \pm 1000$ руб. или $15\,000 \pm 1500$ руб. В фирме B его месячная зарплата оценивается (также в зависимости от заказов) в размере 3000 ± 500 руб. или же 8000 ± 1000 руб. Оценить суммарную месячную зарплату работника.

Решение. **Шаг 1.** В фирме A 1-ю зарплату можно представить в виде интервала $[a_1^1, a_2^1] = [9000, 11\,000]$, 2-ю — как интервал $[a_1^2, a_2^2] = [13\,500, 16\,500]$. Аналогично этому, в фирме B 1-ю зарплату

можно представить в виде интервала $[b_1^1, b_2^1] = [2500, 3500]$, а 2-ю — в виде интервала $[b_1^2, b_2^2] = [7000, 9000]$. Итак, месячную зарплату в фирмах A и B можно представить соответственно следующими полиинтервалами:

$$\tilde{A} = \bigcup_{i=1}^2 [a_1^i, a_2^i] = [9000, 11\ 000] \cup [13\ 500, 16\ 500],$$

$$\tilde{B} = \bigcup_{j=1}^2 [b_1^j, b_2^j] = [2500, 3500] \cup [7000, 9000].$$

Месячная суммарная зарплата работника \tilde{C} равна сумме его месячных зарплат в фирмах A и B , т. е. $\tilde{C} = \tilde{A} + \tilde{B}$. После подстановки значений полиинтервалов \tilde{A} и \tilde{B} получаем

$$\begin{aligned} \tilde{C} &= \bigcup_{i=1}^2 [a_1^i, a_2^i] + \bigcup_{j=1}^2 [b_1^j, b_2^j] = \\ &= ([9000, 11\ 000] \cup [13\ 500, 16\ 500]) + \\ &\quad + ([2500, 3500] \cup [7000, 9000]). \end{aligned}$$

Последняя формула и есть математическая модель решения задачи в виде вычисления суммы двух полиинтервалов.

Шаг 2. Блок-схема алгоритма вычисления функции-модели, полученной на шаге 1 алгоритма, очевидна и содержит всего одну ступень, на которой вычисляется сумма двух полиинтервалов.

Шаг 3. Вычисляем полиинтервальную функцию-модель, полученную на шаге 1. Эта функция — сумма двух полиинтервалов, содержащих каждый два интервала. По формуле (12) сложения полиинтервалов находим

$$\begin{aligned} \tilde{C} &= \bigcup_{i=1}^2 [a_1^i, a_2^i] + \bigcup_{j=1}^2 [b_1^j, b_2^j] = \\ &= [a_1^1 + b_1^1, a_2^1 + b_2^1] \cup [a_1^1 + b_1^2, a_2^1 + b_2^2] + \\ &\quad + [a_1^2 + b_1^1, a_2^2 + b_2^1] \cup [a_1^2 + b_1^2, a_2^2 + b_2^2], \end{aligned}$$

что после подстановки численных значений переменных a_k^i, b_s^j дает

$$\begin{aligned} \tilde{C} &= [9000 + 2500, 11\ 000 + 3500] \cup \\ &\quad \cup [9000 + 7000, 11\ 000 + 9000] \cup \\ &\quad \cup [13\ 500 + 2500, 16\ 500 + 3500] \cup \\ &\quad \cup [13\ 500 + 7000, 16\ 500 + 9000] = \\ &= [11\ 500, 14\ 500] \cup [16\ 000, 20\ 000] \cup \\ &\quad \cup [16\ 000, 20\ 000] \cup [20\ 500, 25\ 500] = \\ &= [11\ 500, 14\ 500] \cup [16\ 000, 20\ 000] \cup [20\ 500, 25\ 500]. \end{aligned}$$

Таким образом, суммарная месячная зарплата работника, в зависимости от заказов у фирм A и B , может лежать в интервалах $[11\ 500, 14\ 500]$ или $[16\ 000, 20\ 000]$ или $[20\ 500, 25\ 500]$ руб. или, в другой форме записи, составлять $13\ 000 \pm 1500$ или $18\ 000 \pm 2000$ или $23\ 000 \pm 2500$ руб.

Как показано ранее, дальнейшее развитие концепции интервальной неопределенности приводит к понятию полиинтервала, характеризующего более сложную неопределенность, имеющую вид последовательности интервалов неопределенности. Такая неопределенность характеризуется тем, что параметр системы не просто принимает какое-то, заранее неизвестное, значение внутри определенного заданного интервала, но еще сначала выбирает какой-нибудь, заранее неизвестный, интервал из нескольких заданных интервалов, внутри которого затем принимает какое-то, заранее неизвестное значение. Эта более сложная модель неопределенности встречается очень часто в экономике, технике и иных областях и потому заслуживает изучения и разработки. Логично осуществлять эту разработку, используя подходы интервальной математики [1, 2] и развивая их в направлении учета многоинтервальности. Подобно тому, как интервальная математика базируется на алгебре интервалов, полиинтервальная математика базируется на алгебре полиинтервалов. Однако в отличие от алгебры интервалов, в алгебре полиинтервалов не имеется простых зависимостей между сложностью (длиной) операндов и сложностью результата операции. Это вызвано большей сложностью неопределенных систем, описываемых алгеброй полиинтервалов.

Заключение

В работе сформулирована задача изучения новой модели неопределенности — так называемого полиинтервала, обобщающей известную модель неопределенности — интервал — на случай существования нескольких последовательных интервалов неопределенности. С помощью известной из интервальной математики теоретико-множественной конструкции, аналогично операциям над интервалами введены операции над полиинтервалами. Разработана методика сведения операций над интервалами к операциям над полиинтервалами. С ее помощью выведены формулы для конструктивного выполнения всех операций над полиинтервалами и построен соответствующий алгоритм. На примере из области экономики проиллюстрирована практическая польза разработанной теории и методов.

Список литературы

1. Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления. М.: Мир, 1987.
2. Левин В. И. Интервальная математика и исследование систем в условиях неопределенности. Пенза: Изд-во Пензенского технологического ин-та, 1998. 55 с.
3. Столл Р. Р. Множества. Логика. Аксиоматические теории. М.: Просвещение, 1968. 232 с.

Modeling of Systems with Uncertainty by Polyinterval Mathematics

In recent decades there are more and more new technologies in the military and civilian spheres which associated with studying of uncertainty. These technologies are widely used in engineering, economics, social sphere. To support their new mathematical models and methods are needed. In this regard, this article dedicated to the development of new model of uncertainty (polyinterval) and mathematical methods of studying this model is relevant. The purpose of article is in the detailed design of a new adequate mathematical model of uncertainty — polyinterval, which is a sequence of a finite number of intervals of uncertainty, system of algebraic operations on polyintervals and rules to perform these operations. To accomplish this goal we propose to extend to study polyintervals the method from the interval mathematics based on the determination of algebraic operations on intervals in form of set-theoretic generalizations of operations on real numbers. The novelty of the work lies in the proposed new mathematical model of uncertainty in form of systems of polyintervals, together with mathematical means allowing to perform various operations on polyintervals and thereby enabling them to perform mathematical modeling of uncertain systems. So, the article detailed developed a new mathematical model of uncertainty — polyinterval. The system of algebraic operations on polyintervals is determined and rules for their implementation are output. Algorithm of study of uncertain systems with polyinterval parameters is given.

Keywords: interval value, polyinterval value, uncertainty, algebra of polyinterval values, system modeling

References

1. Alefel'd G., Herberger Ju. *Vvedenie v interval'nye vychislenija*, Moscow, Mir, 1987 (in Russian).

2. Levin V. I. *Interval'naja matematika i issledovanie sistem v uslovijah neopredelennosti*, Penza, Izd-vo Penzenskogo tehnologicheskogo in-ta, 1998, 55 p. (in Russian).

3. Stoll R. R. *Mnozhestva. Logika. Aksiomaticheskie teorii*, Moscow, Prosveshhenie, 1968, 232 p. (in Russian).

УДК 517.97

Н. П. Деменков, канд. техн. наук, доц., e-mail: dnp@bmstu.ru,
Е. А. Микрин, д-р техн. наук, проф., e-mail: evgeny.mikrin@bmstu.ru,
И. А. Мочалов, д-р техн. наук, проф., e-mail: intelsyst@mail.ru,
МГТУ им. Н. Э. Баумана

Нечеткое преобразование Лапласа в задачах нечеткого математического моделирования. Часть II*

Решается нечеткое дифференциальное уравнение путем линейризации его по методу декомпозиции с последующим применением к нему нечеткого преобразования Лапласа. Методом нечеткого преобразования решена нечеткая двухточечная краевая задача при оптимизации простейшей динамической системы. Этим же методом решены нечеткие интегральные уравнения Вольтерра 2-го рода типа свертки с четкими и нечеткими ядрами. Предложенное преобразование обобщается на случай его применения к нечеткому дифференциальному оператору в частных производных второго порядка.

Ключевые слова: нечеткое преобразование Лапласа, метод декомпозиции, нечеткая краевая задача, нечеткое интегральное уравнение типа свертки

Введение

В теории систем важное место занимают нелинейные системы, которые во многих случаях можно представить в виде совокупной линейной (Л) и нелинейной (Н) частей [1]. Типичные структурные

схемы таких систем представлены на рис. 1, 2. Нелинейность в таких системах, как правило, аппроксимируется в виде степенных, экспоненциальных, интегральных или других форм. Например, в случае степенных форм появляются нелинейные системы Риккати (Riccati) и Дуффинга (Duffing) с показателями степеней $n = 2, 3$ соответственно в представлении нелинейностей.

* Часть I опубликована в журнале "Информационные технологии", т. 23, № 4, 2017 г.

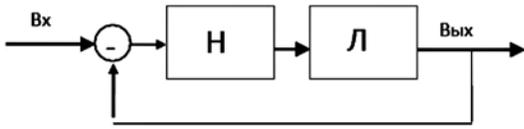


Рис. 1. Структурная схема колебательной сервосистемы с нелинейностью (Н) и линейностью (Л)

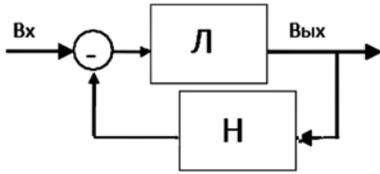


Рис. 2. Структурная схема системы с нелинейностью (Н) в цепи обратной связи и линейностью (Л) в прямой цепи при исследовании субгармонических колебаний методом гармонического баланса

Нечеткость применительно к дифференциальным уравнениям обычно связывают с работами Дубоиса (Dubois) и Праде (Prade) [2, 3]. В настоящее время результаты этих публикаций получили распространение при решении нечетких начальных и краевых задач, нечетких интегро-дифференциальных в частных производных уравнений Вольтерра (Volterra) типа свертки.

Ниже будет рассмотрено применение нечеткого преобразования Лапласа для решения нечеткой нелинейной начальной, двухточечной краевой задачи и интегро-дифференциального уравнения Вольтерра.

1. Декомпозиционный метод нечеткого преобразования Лапласа в нечетких нелинейных системах управления

1.1. Постановка задачи

В соответствии со структурной схемой рис. 2 имеем следующее четкое (традиционное) нелинейное дифференциальное уравнение первого порядка с нечетким начальным условием (нечеткая задача Риккати), заданным в параметрической форме:

$$\dot{y}(x) + c_0 y(x) + c_1 y^2(x) = f(x); \quad (1)$$

$$y_H(0) = y_0(r) = (\underline{\alpha}_0(r), \bar{\alpha}_0(r)), \quad \underline{\alpha}_0(r) = a_1 + (a_2 - a_1)r, \\ \bar{\alpha}_0(r) = a_3 + (a_2 - a_3)r, \quad r \in [0; 1], \quad (2)$$

где $y_H(0) = (a_1, a_2, a_3)$ — нечеткая переменная с треугольной функцией принадлежности, "н" — индекс нечеткости; $f(x)$ — заданная правая часть (1); c_0, c_1 — константы.

Задача состоит в нахождении нечеткого решения $y_H(x) = y(x, r) = (\underline{y}(x, r), \bar{y}(x, r))$ для нечеткой начальной задачи (1), (2).

1.2. Метод решения

Для решения задачи (1), (2) используется методика, предложенная в работах [4, 5], когда исходное уравнение представляется в виде Л и Н частей.

Нечеткое преобразование Лапласа, применительно к (1), (2) дает:

$$sL[y(x)] - y_0(r) + c_0 L[y(x)] + c_1 L[y^2(x)] = L[f(x)]. \quad (3)$$

Основные положения теории относительно L-преобразования изложены в части I статьи (см. "Информационные технологии", 2017, т. 23, № 4). Полагается, что

$$L[y(x)] = (l\underline{y}(x, r), l\bar{y}(x, r)).$$

Подставляя в (3) нечеткое начальное условие (2), получим:

$$sL[y(x)] - (\underline{\alpha}_0(r), \bar{\alpha}_0(r)) + c_0 L[y(x)] + c_1 L[y^2(x)] = L[f(x)],$$

откуда

$$L[y(x)] = s^{-1}(\underline{\alpha}_0(r), \bar{\alpha}_0(r)) - s^{-1}c_0 L[y(x)] - s^{-1}c_1 L[y^2(x)] + s^{-1}L[f(x)].$$

В нечетком виде будем иметь:

$$l\underline{y}(x, r) = s^{-1}\underline{\alpha}_0(r) - s^{-1}c_0 l\underline{y}(x) - s^{-1}c_1 l\underline{y}^2(x) + s^{-1}L[f(x)]; \quad (4)$$

$$l\bar{y}(x, r) = s^{-1}\bar{\alpha}_0(r) - s^{-1}c_0 l\bar{y}(x) - s^{-1}c_1 l\bar{y}^2(x) + s^{-1}L[f(x)]. \quad (5)$$

В технике декомпозиционного преобразования Лапласа имеет место следующее представление решения в виде бесконечного ряда:

$$y(x) = \sum_{n=0}^{\infty} y_n, \quad (6)$$

где y_n вычисляется рекуррентно.

Нелинейный оператор $h(y) = y^2$ является декомпозиционным, поэтому полагается, что

$$h(y) = y^2 = \sum_{n=0}^{\infty} A_n, \quad (7)$$

где A_n — полиномы Адомиана (Adomian) для y_0, \dots, y_n, \dots , которые вычисляются по формуле

$$A_n = \frac{1}{n!} \left\{ \frac{d^n h \left(\sum_{i=0}^{\infty} \lambda^i y_i \right)}{d\lambda^n} \right\}_{\lambda=0}, \quad n = 1, 2, \dots$$

Первые несколько полиномов имеют вид:

$$A_0 = h(y_0); A_1 = \dot{h}(y_0); A_2 = 0,5\ddot{h}(y_0)y_1^2 + \dot{h}(y_0)y_2;$$

$$A_3 = y_3\dot{h}(y_0) + \frac{1}{2}y_1y_2\ddot{h}(y_0) + \frac{1}{2}y_1\ddot{h}(y_0). \quad (8)$$

Например, для уравнения Риккати $h(y) = y^2$, поэтому

$$A_0 = y_0^2, A_1 = 2y_0y_1, A_2 = y_1^2 + 2y_0y_1,$$

$$A_3 = 2y_3y_0 + y_1y_2, \dots$$

Подставим (6), (7) в (4), (5). В результате получим:

$$l\left[\sum_{n=0}^{\infty} \underline{y}_n\right] = s^{-1}\underline{\alpha}_0(r) - s^{-1}c_0l\left[\sum_{n=0}^{\infty} \underline{y}_n\right] -$$

$$- s^{-1}c_1l\left[\sum_{n=0}^{\infty} \underline{A}_n\right] + s^{-1}l[f(x)]; \quad (9)$$

$$l\left[\sum_{n=0}^{\infty} \bar{y}_n\right] = s^{-1}\bar{\alpha}_0(r) - s^{-1}c_0l\left[\sum_{n=0}^{\infty} \bar{y}_n\right] -$$

$$- s^{-1}c_1l\left[\sum_{n=0}^{\infty} \bar{A}_n\right] + s^{-1}l[f(x)]. \quad (10)$$

Сравнивая левую и правую части уравнения (9), а затем и (10), получим следующие итерационные формулы:

$$l[\underline{y}_0] = \underline{d}_1 + d_2l[f(x)], \underline{d}_1 = \underline{\alpha}_0s^{-1}, d_2 = s^{-1},$$

$$l[\underline{y}_1] = -d_3l[\underline{y}_0] - d_4l[\underline{A}_0], d_3 = c_0s^{-1}, d_4 = c_1s^{-1},$$

$$\dots$$

$$l[\underline{y}_{n+1}] = -d_3l[\underline{y}_n] - d_4l[\underline{A}_n].$$

Подобные формулы получаются и для верхнего случая. При $\bar{d}_1 = \bar{\alpha}_0s^{-1}$ получим:

$$l[\bar{y}_0] = \bar{d}_1 + d_2l[f(x)]; l[\bar{y}_1] = -d_3l[\bar{y}_0] - d_4l[\bar{A}_0],$$

$$\dots, l[\bar{y}_{n+1}] = -d_3l[\bar{y}_n] - d_4l[\bar{A}_n].$$

Далее, представляя $f(x)$ в виде тейлоровского приближения в точке $x = x_0 = 0$, получим:

$$f(x) \cong \sum_{i=0}^k b_i x^i, b_i = \frac{1}{i!} f^{(i)}(x=0), i = \overline{0, k}.$$

Подставляя приближение $f(x)$ в итерационные формулы и применяя обратное преобразование Лапласа, получим нечеткое приближенное решение (1) в следующей форме:

$$y_0(r) = (\underline{y}_0(r), \bar{y}_0(r)); y_1(r) = (\underline{y}_1(r), \bar{y}_1(r));$$

$$y_2(r) = (\underline{y}_2(r), \bar{y}_2(r)); \dots$$

В общем случае, когда нелинейность имеет вид $h(y) = y^n$ (уравнение Бернулли), первые полиномы Адомиана A_i равны

$$A_0 = y_0^n, A_1 = ny_0^{n-1}y_1,$$

$$A_2 = ny_0^{n-1}y_2 + 0,5n(n-1)y_0^{n-2}y_1^2,$$

$$A_3 = ny_0^{n-1}y_3 + 0,5n(n-1)y_0y_1y_2 +$$

$$+ n(n-1)(n-2), \dots$$

При $n = 2$ эти формулы совпадают с полиномами A_i для уравнения Риккати, полученными ранее, а при $n = 3$ будем иметь соответствующие полиномы для уравнения Дуффинга.

Пример [5]. Имеем нечеткую начальную задачу для уравнения Дуффинга ($n = 3$):

$$\ddot{y}(x) + 3y(x) - 2y^3(x) = \cos x \cdot \sin 2x;$$

$$y_H(0) = (a_1 = -1/a_2 = 0/a_3 = 1);$$

$$\dot{y}_H(0) = (b_1 = 0/b_2 = 1/b_3 = 2),$$

где нечеткие начальные условия с треугольными функциями принадлежности в параметрической форме имеют вид:

$$y_H(0) = y_0(r) = (\underline{y}_0(r) = a_1 + (a_2 - a_1)r,$$

$$\bar{y}_0(r) = a_3 + (a_2 - a_3)r = (-1 + r; 1 - r);$$

$$\dot{y}_H(0) = (\underline{y}_0'(r) = b_1 + (b_2 - b_1)r;$$

$$\bar{y}_0'(r) = b_3 + (b_2 - b_3)r = (r; 2 - r); r \in [0; 1].$$

Решение. Тейлоровское приближение правой части исходного уравнения равно:

$$\hat{f}(x) = \cos x \cdot \sin 2x \cong (1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots)(2x - \frac{(2x)^3}{3!} +$$

$$+ \frac{(2x)^5}{5!} - \dots) \cong 2x - \frac{7}{3}x^3 + \frac{61}{60}x^5 - \frac{547}{2520}x^7.$$

Начальные полиномы Адомиана равны:

$$A_0 = y_0^3, A_1 = 3y_0^2y_1, A_2 = 3y_0y_1^2 + 3y_0^{n-2}y_1^2,$$

$$A_3 = 3y_0^2y_3 + 6y_0y_1y_2 + y_1^3.$$

Нечеткие прямое и обратное преобразования Лапласа для нижнего нулевого приближения \hat{y}_0 к нечеткому решению дает:

$$l[\hat{y}_0] \cong s^{-1}\underline{y}_0(r) + s^{-2}\dot{y}_0(r) + s^{-2}l[\hat{f}(x)] \Rightarrow \Gamma^{-1} \Rightarrow$$

$$\Rightarrow \hat{y}_0(x, r) = -1 + r + \frac{rx(2x^3)}{3!} - \frac{7 \cdot 3!x^5}{3 \cdot 5!} + \frac{61x^7}{60 \cdot 7!}.$$

Далее аналогично по той же методике определяются $\hat{y}_1, \hat{y}_2, \hat{y}_3$ и $\underline{\Phi}(r) = \sum_{i=0}^3 \hat{y}_i$. Точное решение

ние исходной задачи определяется при $r = 1$ и равно $y = \sin(x = \pi/2) = 1$. Вместе с тем $\underline{\Phi}(r = 1) = \overline{\Phi}(r = 1) = 1$, что косвенно указывает на корректность решения нечеткой начальной задачи Дуффинга.

2. Решение нечеткой двухточечной краевой задачи методом нечеткого преобразования Лапласа

Краевые задачи для динамических систем имеют широкую область применительно к проблемам управления. Они возникают при управлении прицельным бомбометанием, артиллерийским огнем, расчете параметров автоколебаний в поисковых системах оптимизации (САО), решении задач вариационного исчисления, приводящим к различного рода оптимальным регуляторам и т. д. [6—12].

Оптимизация функционалов, зависящих от искомой функции и ее производных, приводят к краевым задачам относительно дифференциальных уравнений второго порядка.

Обычно динамические параметры и краевые условия в этих задачах имеют неопределенности в своем задании. Они могут моделироваться нечеткостью, что приводит к нечетким краевым задачам. В линейном случае эти задачи решаются путем использования нечеткого преобразования Лапласа.

2.1. Постановка задачи

Необходимо найти нечеткое программное управление $u_H(t)$ для динамического объекта

$$\dot{x}_H = \varphi(x(t), u_H(t), t), \quad (11)$$

которое оптимизирует функционал качества

$$\text{ext}_{u_H} J(u_H) = \text{ext}_{u_H} \int_{t_0}^{t_k} f_0(x(t), u_H(t), t) dt + f_k(x(t_k), t_k) \quad (12)$$

с нечеткими граничными условиями

$$x(t = t_0) = x_{0H}; \quad x(t = t_k) = x_{kH}. \quad (13)$$

Здесь $f_0(\cdot)$ — нечеткий интегрант; $f_k(\cdot)$ — терминальный член; x, u_H — векторы состояния и нечеткого управления соответственно.

Оптимизационная задача (11)—(13) не имеет ограничений на управление, поэтому при применении принципа максимума появляется нечеткая двухточечная краевая задача для дифференциального уравнения второго порядка для x_{jH} координаты вектора x_H :

$$\begin{cases} \ddot{x}_H = f(x_H, \dot{x}_H, t); \\ x_H(t = t_0) = x_{0H}; \quad x_H(t = t_k) = x_{kH}. \end{cases} \quad (14)$$

Индекс "j" опущен для простоты обозначений.

2.2. Метод решения

Пусть в задаче (14) используется обобщенная производная по Хукухара (см. часть I статьи) для первой и второй производных. Это приводит к различным типам нечетких задач, число которых $m = 2^n$, где n — порядок исходного уравнения. Для уравнения (14) имеем $m = 2^2 = 4$ типа уравнений: (i, i), (i, ii), (ii, i), (ii, ii). Здесь символ на первом месте означает тип первой обобщенной производной, символ на втором месте после запятой — тип второй обобщенной производной.

Пусть (14) является нечетким линейным уравнением, тогда каждый тип краевой задачи может быть решен путем использования нечеткого преобразования Лапласа. Ниже приводится простейший пример решения нечеткой задачи (14) типа (i, i).

Пример. Имеем модель объекта управления

$$\dot{x}(t) = u(t), \quad x \in R^1, \quad u \in R^1, \quad t \in [0, 1]$$

и функционал качества управления

$$J = \int_{t_0}^{t_k} [u^2(t) + x^2(t)] dt \rightarrow \min$$

с нечеткими граничными условиями

$$x_H(t_0 = 0) = x_{0H} = 0_H, \quad x_H(t_k = 1) = x_{kH} = 0,5_H.$$

Необходимо найти нечеткую оптимальную пару $(x_H^*(t), u_H^*(t))$.

Здесь имеем нечеткую задачу Лагранжа. Из принципа максимума следует:

$$\begin{aligned} \ddot{x}_H(t) - x_H(t) &= 0, \quad x_H(t_0 = 0) = x_{0H} = 0_H, \\ x_H(t_k = 1) &= x_{kH} = 0,5_H. \end{aligned}$$

Нечеткие числа $0_H; 0,5_H$ в параметрической форме имеют вид

$$\begin{aligned} 0_H &= (\underline{0} = -1 + r, \overline{0} = 1 - r; \quad r \in [0, 1]); \\ 0,5_H &= (\underline{0,5} = 0,5r, \overline{0,5} = 1 - 0,5r; \quad r \in [0, 1]). \end{aligned}$$

Случай 1. (Краевая задача (i, i) типа). Будем использовать методику, изложенную в части I статьи. Имеем:

$$\begin{aligned} L\{\ddot{x}_H(t)\} - L\{x_H(t)\} &= 0 \Rightarrow \\ \Rightarrow [L\{x(t, r)\} &= \frac{s}{s^2 - 1} x(0, r)] + \frac{F}{s^2 - 1}, \end{aligned}$$

где

$$\underline{F} = \dot{x}(0, r).$$

Применяя обратное преобразование Лапласа Γ^{-1} , получим

$$\underline{x}(t, r) = 0,5(r - 1)(e^t + e^{-t}) + 0,5\underline{F}(e^t - e^{-t}).$$

Значение \underline{F} находим из второго нижнего граничного условия $\underline{x}(1, r) = 0,5r$:

$$0,5r = 0,5(r-1)(e^1 + e^{-1}) + 0,5\underline{F}(e^1 - e^{-1}),$$

откуда

$$\underline{F} = [r - (r-1)(e^1 + e^{-1})](e^1 - e^{-1})^{-1}.$$

В результате получим для нижнего уровня:

$$\underline{x}^*(t, r) = 0,5\{(r-1)(e^t + e^{-t}) + [r - (r-1)(e^1 + e^{-1})](e^1 - e^{-1})^{-1}(e^t - e^{-t})\}.$$

Аналогичные вычисления для верхнего уровня дают:

$$\bar{x}^*(t, r) = 0,5\{(1-r)(e^t + e^{-t}) + 1 - 0,5r - 0,5(1-r)(e^1 + e^{-1})[0,5(e^1 - e^{-1})^{-1}](e^t - e^{-t})\}.$$

Корректность вычислений $\underline{x}^*(t, r=1)$, $\bar{x}^*(t, r=1)$ осуществляется путем вычисления $\underline{x}^*(t=0, r)$, $\underline{x}^*(t=1, r)$, $\bar{x}^*(t=0, r)$, $\bar{x}^*(t=1, r)$ и сравнения их с заданными нечеткими граничными условиями.

Нечеткое управление $u_H^*(t)$ находится из модели объекта управления

$$\begin{aligned} u_H^*(t) = \dot{x}_t(t) \Rightarrow u_H^*(t) = u^*(t, r) = \\ = \left(\underline{u}^*(t, r) = \frac{\partial}{\partial t} \underline{x}(t, r), \bar{u}^*(t, r) = \frac{\partial}{\partial t} \bar{x}(t, r); r \in [0; 1] \right). \end{aligned}$$

Таким образом, найдена искомая нечеткая оптимальная пара $(x_H^*(t), u_H^*(t))$ нечеткого программного управления.

По аналогичной методике рассматриваются все остальные типы краевых задач (i, ii), (ii, i), (ii, ii).

3. Решение нечеткого интегро-дифференциального уравнения Вольтерра типа свертки методом нечеткого преобразования Лапласа

Интегральные уравнения широко применяются в управлении при описании элементов систем, в теории дифференциальных уравнений при доказательстве существования и единственности их решений, представлении различных типов интегральных преобразований и т. д.

Представляет практический интерес модификация традиционных интегральных уравнений в их нечеткие аналоги, что позволяет учитывать различного рода неопределенности в их задании. Это приводит к нечетким интегральным уравнениям Фредгольма—Вольтерра, в которых нечеткое неизвестное $x_H(t)$ находится под знаком интеграла:

$$x_H(t) = F \left(\int_0^t f(x_H, \tau) d\tau, \int_0^t g(x_H, \eta) d\eta \right), t \in [0, T] \subset R^1. \quad (15)$$

В работе [13] доказывается теорема существования и единственности решения уравнения (15) при наличии некоторых условий, налагаемых на оператор F .

Ниже рассматриваются различные модификации нечеткого интегрального уравнения Вольтерра 2-го рода типа свертки, которое решается методом нечеткого преобразования Лапласа.

3.1. Нечеткая свертка [14—16]. Для двух нечетких функций f_H и g_H нечеткая свертка определяется как

$$(f_H * g_H)(t) = \int_0^t f_H(\tau) g_H(t - \tau) d\tau, t > 0.$$

После замены $u = t - \tau$ получим

$$(f_H * g_H)(t) = \int_0^t g_H(u) f_H(t - u) du = (g_H * f_H)(t).$$

Для нечеткой свертки "*" выполняются следующие очевидные свойства:

$c(f_H * g_H) = cf_H * g_H = f_H * cg_H$, c — константа;

$f_H * (g_H * h_H) = (f_H * g_H) * h_H$ — ассоциативность.

Теорема (без доказательства). Если f_H и g_H — кусочно непрерывные нечеткие функции на $[0, \infty]$ и имеют экспоненциальный порядок роста, то

$$L\{f_H * g_H\} = L\{f_H(t)\}L\{g_H(t)\} = F_H(s)F_G(s), s > p.$$

3.2. Постановка задачи

Будем рассматривать три варианта постановки задачи.

Вариант первый. Задано нечеткое интегральное уравнение свертки с четким ядром.

Вариант второй. Задано нечеткое интегральное уравнение свертки с нечетким ядром.

Вариант третий. Задано нечеткое интегро-дифференциальное уравнение в частных производных типа свертки.

3.3. Метод решения

Для решения задач по трем вариантам будем использовать нечеткое преобразование Лапласа.

Вариант первый. Имеем

$$x_H(t) = f_H(t) + \int_0^t k(\tau - t)x_H(\tau) d\tau, t \in [0; T], T \rightarrow \infty, \quad (16)$$

где $k(\tau - t)$ — четкое ядро уравнения. Необходимо найти $x_H(t)$.

Взяв от обеих частей выражения (16) нечеткое преобразование Лапласа, получим:

$$L[x_H(t)] = L[f_H(t)] + L \left[\int_0^t k(\tau - t)x_H(\tau) d\tau \right].$$

Далее из определений нечеткой свертки и нечеткого преобразования Лапласа имеем:

$$\mathbb{L}\underline{x}(t, r) = \mathbb{L}\underline{f}(t, r) + \mathbb{L}k(t, r)\mathbb{L}\underline{x}(t, r); \quad (17)$$

$$\mathbb{L}\bar{x}(t, r) = \mathbb{L}\bar{f}(t, r) + \mathbb{L}k(t, r)\mathbb{L}\bar{x}(t, r). \quad (18)$$

Для компонент $\mathbb{L}k(t, r)\mathbb{L}\underline{x}(t, r)$, $\mathbb{L}k(t, r)\mathbb{L}\bar{x}(t, r)$ рассмотрим следующие случаи.

Случай 1. Если $k(\cdot)$ и $x(\cdot)$ положительны, то

$$\frac{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)}{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)} = \mathbb{L}k(\cdot)\mathbb{L}x(\cdot);$$

Случай 2. Если $k(\cdot)$ — положительна, а $x(\cdot)$ — отрицательна, то

$$\frac{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)}{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)} = \mathbb{L}k(\cdot)\mathbb{L}\bar{x}(\cdot);$$

Случай 3. Если $k(\cdot)$ — отрицательна, а $x(\cdot)$ — положительна, то

$$\frac{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)}{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)} = \mathbb{L}\bar{k}(\cdot)\mathbb{L}x(\cdot);$$

Случай 4. Если $k(\cdot)$ и $x(\cdot)$ отрицательны, то

$$\frac{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)}{\mathbb{L}k(\cdot)\mathbb{L}x(\cdot)} = \mathbb{L}\bar{k}(\cdot)\mathbb{L}\bar{x}(\cdot);$$

Например, для случая 1 будем иметь из (17), (18):

$$\begin{aligned} \mathbb{L}\underline{x}(\cdot) &= \mathbb{L}\underline{f}(\cdot) + \mathbb{L}k(\cdot)\mathbb{L}\underline{x}(\cdot) \Rightarrow \\ &\Rightarrow \mathbb{L}\underline{x}(\cdot) = \mathbb{L}\underline{f}(\cdot)(1 - \mathbb{L}k(\cdot))^{-1}; \\ \mathbb{L}\bar{x}(\cdot) &= \mathbb{L}\bar{f}(\cdot) + \mathbb{L}k(\cdot)\mathbb{L}\bar{x}(\cdot) \Rightarrow \\ &\Rightarrow \mathbb{L}\bar{x}(\cdot) = \mathbb{L}\bar{f}(\cdot)(1 - \mathbb{L}k(\cdot))^{-1}, \end{aligned}$$

откуда после применения обратного преобразования Лапласа получим:

$$\begin{aligned} \underline{x}(t, r) &= l^{-1}\{\mathbb{L}\underline{f}(x, r)(1 - \mathbb{L}k(x, r))^{-1}\}; \\ \bar{x}(t, r) &= l^{-1}\{\mathbb{L}\bar{f}(x, r)(1 - \mathbb{L}k(x, r))^{-1}\}. \end{aligned}$$

Формулы для случаев 2, 3, 4 могут быть получены аналогичным способом, как в случае 1.

Пример. Имеем:

$$x_H(t) = f_H(t) + \int_0^t \sin(t - \tau)x_H(\tau)d\tau,$$

$$f_H(t) = \underline{f}(t, r) = re^{-t}; \quad \bar{f}(t, r) = (2 - r)e^{-t},$$

$k(t) = \sin(t - \tau)$ — четкое ядро уравнения.

Применяя преобразование Лапласа к обеим частям уравнения, получим

$$\begin{aligned} L[x(\cdot)] &= L[f_H(\cdot)] + L[\sin t]L[x(\cdot)] \Leftrightarrow \\ &\Leftrightarrow \begin{cases} \mathbb{L}\underline{x}(\cdot) = \mathbb{L}re^{-t} + \mathbb{L}[\sin t]\mathbb{L}\underline{x}(\cdot) \\ \mathbb{L}\bar{x}(\cdot) = \mathbb{L}(2 - r)e^{-t} + \mathbb{L}[\sin t]\mathbb{L}\bar{x}(\cdot), \end{cases} \end{aligned}$$

откуда

$$\mathbb{L}\underline{x}(\cdot) = r\left(\frac{2}{s+1} + \frac{1}{s^2} - \frac{1}{s}\right);$$

$$\mathbb{L}\bar{x}(\cdot) = (2 - r)\left(\frac{2}{s+1} + \frac{1}{s^2} - \frac{1}{s}\right).$$

После применения обратного преобразования Лапласа получим

$$\underline{x}(t, r) = r(2e^{-t} + t - 1);$$

$$\bar{x}(t, r) = (2 - r)(2e^{-t} + t - 1), \quad r \in [0; 1].$$

Вариант второй. Имеем уравнение

$$\begin{aligned} x(t) &= (r, 2 - r)t + \int_0^t (r + 2, 4 - r)\cos(t - \tau)x(\tau)d\tau, \\ t &\in [0; T], \quad T \rightarrow \infty, \end{aligned}$$

где $k(t) = (r + 2, 4 - r)\cos(t - \tau)$ — нечеткое ядро уравнения.

Расчеты с применением операторов l и l^{-1} дают:

$$\begin{aligned} \underline{x}(t, r) &= r[1 + 2(r + 2) \times \\ &\times (r^2 + 4r)e^{(0,5r+1)t}\text{sh}(0,5(r+4r)^{0,5}t)]; \end{aligned}$$

$$\begin{aligned} \bar{x}(t, r) &= (2 - r)[1 - 2(r - 4) \times \\ &\times (r^2 - 8r + 12)e^{-(0,5r+2)t}\text{sh}(0,5(r^2 - 8r + 12)^{0,5}t)]. \end{aligned}$$

Вариант третий. Имеем уравнение

$$xu_x(x, t) =$$

$$= u_{tt}(x, t) + (x\sin t)(r - 1, 1 - r) + \int_0^t \sin(t - \tau)u(x, \tau)d\tau$$

с нечеткими начальными условиями:

$$u(x, t = 0, r) = (\underline{u}(x, t = 0, r) = 0; \bar{u}(x, t = 0, r) = 0);$$

$$u_t(x, t = 0, r) = (\underline{u}_t(x, t = 0, r) = (r - 1)x;$$

$$\bar{u}_t(x, t = 0, r) = (1 - r)x)$$

и нечетким граничным условием:

$$u(x = 1, t, r) = (\underline{u}(x = 1, t, r) = (r - 1)t;$$

$$\bar{u}(x = 1, t, r) = (1 - r)t),$$

где

$$u_c(\cdot) = \frac{\partial u}{\partial t}, \quad u_{tt}(\cdot) = \frac{\partial^2 u}{\partial t^2}.$$

Применение L -преобразования по переменной t дает:

$$xL[u_x] = L[u_{tt}] + x(r - 1, 1 - r)L[\sin t] + L[\sin t]L[u],$$

откуда получаем

$$\begin{aligned} x\dot{U}(x, s) &= s^2U(x, s) \ominus sU(x, t = 0) \ominus \\ &\ominus x(r - 1, 1 - r)\frac{1}{s^2 + 1} + \frac{1}{s^2 + 1}U(x, s). \end{aligned}$$

Применение L -преобразования к граничным условиям дает

$$U(x = 1, s, r) = (r - 1, 1 - r) \frac{1}{s^2}.$$

Используя параметрическую форму представления нечетких переменных, получим:

$$x \frac{d}{dx} \underline{U}(x, s, r) = s^2 \underline{U}(x, s, r) - (r - 1)x + \\ + x(r - 1) \frac{1}{s^2 + 1} + \underline{U}(x, s, r) \frac{1}{s^2 + 1};$$

$$x \frac{d}{dx} \overline{U}(x, s, r) = s^2 \overline{U}(x, s, r) - (1 - r)x + \\ + x(1 - r) \frac{1}{s + 1} + \overline{U}(x, s, r) \frac{1}{s + 1}.$$

Для переменной нижнего уровня $\underline{U}(x, s, r)$ имеем:

$$\frac{d}{dx} \underline{U}(x, s, r) - (s^4 + s^2 + 1)(s^2 + 1) \underline{U}(x, s, r) + \\ + (r - 1) \frac{s^2}{s^2 + 1} = 0;$$

Его решение дает:

$$\underline{U}(x, s, r) = (r - 1)sx + C \exp \left[\frac{s^4 + s^2 + 1}{s^2 + 1} \right],$$

где C — константа интегрирования, которая находится из нечеткого граничного условия:

$$\underline{U}(x = 1, s, r) = (r - 1) \frac{1}{s^2}.$$

Подставив граничное условие, получим $C = 0$ и далее соответственно

$$\underline{U}(x, s, r) = (r - 1) \frac{x}{s^2}.$$

После применения оператора Γ^{-1} будем иметь:

$$\underline{u}(x, t, r) = (r - 1)xt.$$

Аналогичные вычисления для нахождения $\overline{u}(x, t, r)$ дают:

$$\overline{u}(x, t, r) = (1 - r)xt.$$

Аналогичным способом могут быть получены решения нечетких линейных дифференциальных уравнений в частных производных второго порядка параболического и гиперболического типов, которые описывают физические процессы во времени и пространстве, например, тепловые, волновые и др.

Заключение

Решены нечеткие нелинейные дифференциальные уравнения, для которых в явной форме можно выделить линейные и нелинейные части, например, нечеткие уравнения Риккати, Дуффинга, Бернулли, которые часто встречаются в задачах оптимального и классического управления. Для их решения используется линеаризация по методу декомпозиции Адомиана с последующим применением нечеткого преобразования Лапласа. Приведен числовой пример по решению дифференциального уравнения Дуффинга с нечеткими начальными условиями.

Получено решение двухточечной краевой задачи для нечеткого дифференциального уравнения второго порядка. Одно из краевых условий используется для нахождения значения начальной производной в нечетком преобразовании Лапласа, с помощью которого решается исходная нечеткая задача. Приведен числовой пример решения нечеткой краевой задачи по методу нечеткого преобразования Лапласа.

Решено нечеткое интегральное уравнение Вольтерра 2-го рода типа свертки для случаев, когда ядро уравнения является четким и нечетким. Задача обобщена на случай наличия в нечетком интегральном уравнении нечеткой дифференциальной составляющей в частных производных. Это позволяет распространить методику применения нечеткого преобразования для случая нечетких линейных уравнений в частных производных второго порядка параболического и гиперболического типов.

Список литературы

1. Сю Д., Мейер А. Современная теория автоматического управления и ее применения. М.: Машиностроение, 1972. 544 с.
2. Dubois D., Prade H. Towards fuzzy differential calculus // Fuzzy sets and systems. 1982. Vol. 8, N. 3. P. 225—233.
3. Dubois D., Prade H. On several definitions of the differential of fuzzy mapping // Fuzzy sets and systems. 1987. Vol. 24, N. 1. P. 117—233.
4. Adomian C. Overview of the decomposition method in applied mathematics // Journal of Mathematical analyses and applications. 1988. Vol. 135, N. 2. P. 501—504.
5. Ahmad N., Mamat M., Kavikumar J. and etc. Solving fuzzy Duffing equation by the Laplace transform decomposition // Applied mathematical sciences. 2012. Vol. 6, N. 59. P. 2935—2944.
6. Современная математика для инженеров / Под ред. Э. Ф. Бекенбаха. М.: ИИЛ, 1959. 500 с.
7. Казакевич В.В., Родов А. Б. Системы автоматической оптимизации. М.: Энергия, 1977. 288 с.
8. Бесекерский В. А., Попов Е. П. Теория систем автоматического регулирования. М.: Наука, 1975. 768 с.
9. Пантелеев А. В., Бортаковский А. С. Теория управления в примерах и задачах. М.: Высшая школа, 2003. 583 с.
10. Пантелеев А. В. Вариационное исчисление в примерах и задачах. М.: Высшая школа, 2006. 272 с.
11. Цлаф Л. Я. Вариационное исчисление и интегральные уравнения. М.: Наука, 1969. 192 с.

12. Эльсгольд Л. Э. Дифференциальные уравнения и вариационное исчисление. М.: Наука, 1969. 425 с.

13. Park J. Y., Jeong J. U. On the existence and uniqueness of solutions of fuzzy Volterra-Fredholm integral equations // *Fuzzy sets and systems*. 2000. Vol. 115, N. 1. P. 425–431.

14. Salahshour S. and etc. Solving fuzzy integral equations of the second kind by fuzzy Laplace transform method // *International Journal industrial mathematics*, 2012. Vol. 4, N. 1. P. 21–29.

15. Das M., Talukdar D. Method for solving fuzzy integral differential equation by using fuzzy Laplace transformation // *International Journal of scientific technology research*. 2014. Vol. 3, N. 5. P. 291–295.

16. Ullah S., Farooq M., Ahmad L. and etc. Application of fuzzy Laplace transforms for solving fuzzy partial Volterra integro-differential equations // *General Mathematics (Math GM)*. 2014, N. 8. P. 1–11.

N. P. Demenkov, Ph. D., Associate Professor, e-mail: dnp@bmstu.ru,

E. A. Mikrin, D. Sc., Professor, e-mail: eugeny.mikrin@bmstu.ru,

I. A. Mochalov, D. Sc., Professor, e-mail: intelsyst@mail.ru,

Bauman Moscow State Technical University

Fuzzy Transformation of Laplace in Tasks of Fuzzy Mathematical Modelling. Part II

The purpose of part II of the present article consists in further synthesis of representations of fuzzy transformation of Laplace, its properties and its application for the solution of fuzzy nonlinear tasks.

The fuzzy differential equation is solved by its linearization by a decomposition method with the subsequent application to it fuzzy transformation of Laplace. The example of an fuzzy point-to-point regional task is solved by method of fuzzy transformation by optimization of the elementary dynamic system.

The fuzzy nonlinear differential equations for which in an obvious form it is possible to allocate linear and nonlinear parts, for example, the fuzzy equations of Rikkati of Duffings, Bernoullis meet in tasks of optimum and classical control also very often. These equations are solved by use of linearization by a method of decomposition of Adomian with the subsequent application of fuzzy transformation of Laplace. The numerical example according to the solution of the differential equation of Duffings with fuzzy entry conditions is given.

The solution of a point-to-point regional task for the fuzzy differential equation of the 2nd order is received. One of regional conditions is used for finding of value of an initial derivative in fuzzy transformation of Laplace by means of which the initial fuzzy problem is solved. The numerical example of the solution of an fuzzy regional task of a method of fuzzy transformation of Laplace is given.

The fuzzy integrated equation of Voltaire's equation of the 2nd sort for cases when the kernel of the equation is accurate and fuzzy is solved. The task is generalized on a case of existence in the indistinct integrated equation of an fuzzy differential component in private derivatives. It allows to extend a technique of application of fuzzy transformation for a case of the fuzzy linear equations in private derivatives of the 2nd order of parabolic and hyperbolic types.

Keywords: fuzzy transformation of Laplace, derivative on Hukukhara, Voltaire's equation of the 2nd sort, the equation of Duffings, a decomposition method, an fuzzy boundary task, the fuzzy integrated equation like convolution

References

1. Sju D., Mejer A. *Sovremennaja teorija avtomaticheskogo upravlenija i ee primenenija* (Modern theory of automatic control and its application). Moscow, Mashinostroenie, 1972. 544 p. (in Russian).

2. Dubois D., Prade H. Towards fuzzy differential calculus, *Fuzzy sets and systems*, 1982, vol. 1, no. 8, pp. 225–233.

3. Dubois D., Prade H. On several definitions of the differential of fuzzy mapping, *Fuzzy sets and systems*, 1987, vol. 24, no. 1, pp. 117–233.

4. Adomian C. Overview of the decomposition method in applied mathematics, *Journal of Mathematical analyses and applications*, 1988, vol. 135, no. 2, pp. 501–504.

5. Ahmad N., Mamat M., Kavikumar J., and etc. Solving fuzzy Duffing equation by the Laplace transform decomposition, *Applied mathematical sciences*, 2012, vol. 6, no. 59, pp. 2935–2944.

6. *Sovremennaja matematika dlja inzhenerov*. Pod red. Je. F. Bekenbaha (Modern mathematics for engineers. Under the editorship of E. F. Bekenbakh), Moscow, IIL, 1959. 500 p. (in Russian).

7. Kazakevich V. V., Rodov A. B. *Sistemy avtomaticheskoy optimizacii* (Systems of automatic optimization). Moscow, Jenergija, 1977. 288 p. (in Russian).

8. Besekerskij V. A., Popov E. P. *Teorija sistem avtomaticheskogo regulirovanija* (Theory of systems of automatic control). Moscow, Nauka, 1975. 768 p. (in Russian).

9. Panteleev A. V., Bortakovskij A. S. *Teorija upravlenija v primerah i zadachah* (The theory of control in examples and tasks). Moscow, Vysshaja shkola, 2003. 583 p. (in Russian).

10. Panteleev A. V. *Variacionnoe ischislenie v primerah i zadachah* (Calculus of variations in examples and tasks). Moscow, Vysshaja shkola, 2006. 272 p. (in Russian).

11. Claf L. Ja. *Variacionnoe ischislenie i integral'nye uravnenija* (Calculus of variations and integrated equations), Moscow, Nauka, 1969. 192 p. (in Russian).

12. Jel'sgol'c L. Je. *Differencial'nye uravnenija i variacionnoe ischislenie* (Differential equations and calculus of variations). Moscow, Nauka, 1969. 425 p. (in Russian).

13. Park J. Y., Jeong J. U. On the existence and uniqueness of solutions of fuzzy Volterra-Fredholm integral equations, *Fuzzy sets and systems*, 2000, vol. 115, pp. 425–431.

14. Salahshour S. and etc. Solving fuzzy integral equations of the second kind by fuzzy Laplace transform method, *International Journal industrial mathematics*, 2012, vol. 4, no. 1, pp. 21–29.

15. Das M., Talukdar D. Method for solving fuzzy integral differential equation by using fuzzy Laplace transformation, *International Journal of scientific technology research*, may 2014, vol. 3, is. 5, pp. 291–295.

16. Ullah S., Farooq M., Ahmad L. and etc. Application of fuzzy Laplace transforms for solving fuzzy partial Volterra integro-differential equations, *Math GM*, may 2014, no. 8, pp. 1–11.

Ю. В. Литвин, канд. экон. наук, директор центра, e-mail: litvinj@simplecs.ru,
Научно-исследовательский институт экономики и организации управления
в газовой промышленности (ООО "НИИгазэкономика", ОАО "Газпром"), г. Москва,
К. С. Ноздрин, нач. научно-техн. центра, e-mail: nozdrinks@yandex.ru,
А. А. Полков, вед. инженер, e-mail: andr.polkov@yandex.ru,
Научно-исследовательский институт автоматической аппаратуры
им. академика В. С. Семенихина" (АО "НИИАА"), г. Москва

Система виртуального прототипирования и моделирования информационно-управляющих систем реального времени

Рассматривается новый подход к оценке вероятностно-временных характеристик (ВВХ) создаваемых информационно-управляющих систем реального времени (ИУС РВ) на основе виртуальных прототипов, имитационных и комбинированных имитационно-регрессионных моделей. Технология применения разработанного подхода демонстрируется конкретным примером оценки влияния распараллеливания процессов ИУС РВ на ВВХ системы.

Ключевые слова: концептуальная модель, виртуальный прототип, комбинированная модель, технология прототипирования и моделирования, имитация, информационно-управляющая система, система реального времени

Введение

Исходя из растущей сложности создаваемых и используемых информационно-управляющих вычислительных систем различной отраслевой направленности, возрастает потребность в развитии инструментария по обоснованию принимаемых решений на стадиях их разработки и эксплуатации. Под информационно-управляющими системами реального времени (ИУС РВ) в данной работе мы будем понимать системы, построенные в виде комплексов автоматизации, предоставляющих пользователям, ведущим управление производственными объектами, информационные и вычислительные ресурсы в пределах заданных временных ограничений (мягкий режим реального времени). Примером ИУС РВ может служить система управления нефтегазовым объектом, осуществляющим поиск, разведку, бурение и разработку нефтегазовых месторождений в режиме реального времени [1, 2]. Проблема состоит в том, что задачи принятия решений приходится ставить и решать в условиях высокой неопределенности и неполноты информации об исследуемой системе и ее окружении. В то же время, откладывая анализ таких решений на более позднее время, когда появятся дополнительные данные о проектируемой системе, есть риск столкнуться с существенными изменениями ранее принятых решений, дополнительными затратами времени и денег на их реализацию. В данной работе, в отличие от традиционного использования формальных методов оценки показателей функционирования вычислительных и коммуникационных систем, предлагается новый комплексный подход к обоснованию эффективности и оценке рисков принимаемых проектных решений, более полно

учитывающий наличие распределенной структуры систем, случайностей и неопределенностей. Разработка и использование данного подхода при создании средств исследования сложных ИУС РВ продиктована новым уровнем технологического развития, совершенствованием систем компьютерного проектирования и накопленной практикой использования для этих целей моделей различных видов [3, 5]. Проект создания ИУС РВ, как и других систем, обычно делится на стадии, на каждой из которых принимаются и оцениваются те или иные решения. Наряду с модельным обоснованием проектных решений в последнее время существенно возрос интерес к использованию виртуального прототипирования [4]. Использование прототипа создаваемых систем позволяет уже на ранних этапах подключить заказчика к выработке детальных требований к создаваемой системе, а разработчикам провести апробирование принимаемых конструкторских решений и осуществить сбор достоверной информации о вычислительном процессе. Эволюционный тип прототипа путем постепенного наращивания в нем функций дает возможность создать, в конце концов, полную ИУС РВ. Однако использование прототипирования решает только часть вопросов проектирования и оценки принимаемых решений. Одновременно требуется использовать и другие инструменты анализа режимов и условий работы будущей системы, причем это следует делать на возможно более ранних стадиях проектирования. Такими инструментами служат комплексные комбинированные модели [5, 6]. В настоящей работе представлена разработанная авторами система виртуального прототипирования и моделирования (СВПМ), поддерживающая процессы проектирования ИУС РВ на различных стадиях их жиз-

ненного цикла. При этом благодаря включению в процесс оценки проектных решений виртуальных прототипов (ВП) и моделей, разрешен ряд вопросов более полного и точного анализа таких систем.

Рассмотрим состав и основные структурные уровни СВПМ, отражающие в обобщенном виде использование заложенных в нее возможностей. Далее на конкретном примере будет продемонстрировано применение указанной технологии в решении конкретного вопроса создаваемой системы ИУС РВ.

Структура системы виртуального прототипирования и моделирования

Обобщенная структура СВПМ, представленная на рис. 1, выполняет следующие функции:

- управление процессами создания и использования динамических виртуальных прототипов и математических моделей ИУС РВ, а также организация работы СВПМ;
- поддержка процессов разработки концептуальных моделей ИУС РВ (определение целей исследования, формирование показателей и критериев, выработка требований, построение концептуальной модели ИУС РВ);
- разработка и использование виртуальных прототипов (создание ВП, отработка интерфейсов, обоснование решений, измерение вычислительного процесса);
- разработка комбинированных моделей отдельных компонентов ИУС РВ, а также распределенных вычислительных систем (разработка имитационных, аналитических, регрессионных моделей и их комбинаций);
- разработка и использование различных видов имитаторов ИУС РВ и других вычислительных систем;
- интеграция и управление процессами построения комплексных систем исследования функционирования ИУС РВ (объединение в единую технологию прототипирования и комбинированного моделирования на различных иерархических уровнях ИУС РВ);
- измерение вычислительного процесса, реализуемого прикладными программами и операционной системой;
- ведение базы знаний и данных СПВМ (знания о методах и моделях ИУС РВ, технологии их применения, базы исходных данных и результатов исследований и др.);
- средства поддержки процессов СПВМ (стандартные пакеты планирования экспериментов, SPSS, AnyLogic и др.).

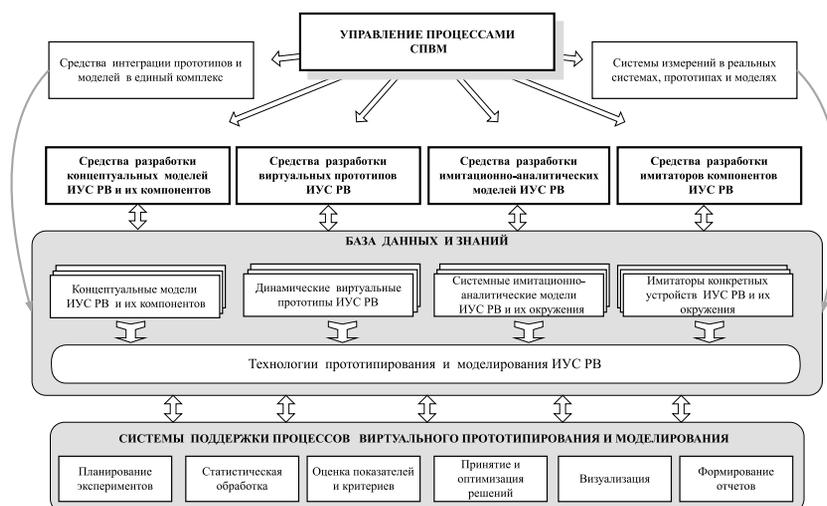


Рис. 1. Система виртуального прототипирования и моделирования

В связи с многоэтапностью и многоплановостью процесса проведения исследований и обоснования решений создаваемой ИУС РВ элементы соответствующей технологии виртуального прототипирования и математического моделирования лучше всего рассмотреть на конкретном примере.

Пример проектирования ИУС РВ на начальной стадии жизненного цикла

Этапы технологии виртуального прототипирования и математического моделирования представлены на рис. 2. Рассмотрим применение развиваемого подхода на конкретном примере оценки решения по организации параллельного выполнения прикладных программ системы.

1. *Концептуальный уровень.* В данном примере будем рассматривать упрощенную клиент-серверную

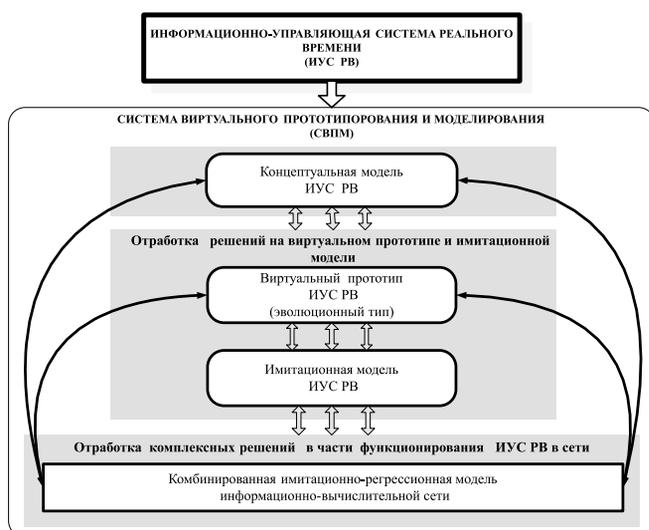


Рис. 2. Этапы технологии виртуального прототипирования и математического моделирования ИУС РВ

систему (содержащую два терминальных узла и сервер приложений) с ограниченным составом реализуемых на сервере функций. ИУС РВ по запросам пользователей должна выполнять информационно-аналитические расчеты в реальном масштабе времени [8]. Программная часть системы реализуется на сервере приложений (СП), входящим в состав комплекса средств автоматизации (КСА), состоящего из web-сервера, сервера приложений, сервера баз данных и локальных терминальных узлов, объединенных в локальную вычислительную сеть. Удаленные пользователи взаимодействуют с КСА через территориальные сети передачи данных. В рассматриваемом примере СП состоит из многоядерного процессора, основной и внешней памяти. Поступающие запросы пользователей обрабатываются создаваемым программным обеспечением [7]. Запросы поступают в СП в случайные моменты времени, присоединяются к очереди к центральному процессору и ожидают выполнения. В данном примере считаем, что в систему поступает два класса запросов, отличающихся потребностью в ресурсах, временем выполнения и назначенными приоритетами. Для упрощения концептуальной модели детали алгоритмов диспетчеризации на данном шаге не рассматриваются. Время от времени выполняемые запросы обращаются к данным, расположенным на локальных накопителях СП или в сервере баз данных. С помощью технологии прототипирования и моделирования будем решать следующую задачу: определение влияния на производительность и вероятностно-временные характеристики (ВВХ) ИУС РВ сегментации разрабатываемых прикладных программ, обеспечивающей необходимый уровень распараллеливания вычислительного процесса на многоядерном процессоре СП.

2. *Уровень виртуального прототипирования.* Следующим шагом применения рассматриваемой технологии является создание ВП, отражающего наиболее важные функции ИУС РВ, выполняемые на вычислительных ресурсах СП. При этом виртуальный прототип должен ответить на вопросы, возникающие при разработке программ, и сформировать требования к ним, а также обосновать решения по интерфейсам взаимодействия с другими подсистемами и пользователями. Создание прототипа включает:

- разработку алгоритмов реализации основных функций СП;
- кодирование функций обработки запросов;
- разработку внутренних и внешних интерфейсов;
- включение измерителей в состав разрабатываемых программ и др.

Применение ВП ИУС РВ на начальной стадии проектирования позволяет решить следующие задачи:

- уже на ранней стадии создания ИУС РВ заказчиком и пользователями совместно с програм-

мистами необходимы обработка структурных вопросов интерфейсов и оценка показателей реактивности и производительности создаваемых компонентов;

- обеспечение выбора уровня сегментации программ и определение баланса между закладываемыми возможностями прикладных программ и настраиваемыми параметрами управления операционной системы с точки зрения реактивности системы и использования ресурсов;
- проведение измерений вычислительного процесса в целях получения реальных данных функционирования прототипа для последующего моделирования.

В результате измерения прототипа были получены результаты, приведенные в табл. 1.

Другие характеристики, полученные в результате прототипирования, не приведены.

3. *Уровень имитационного моделирования ВП.* Выбор имитационной модели в качестве инструмента анализа ВП объясняется тем, что с ее помощью можно отразить основные процессы выполнения программ в прототипе, а также организацию вычислительного процесса с распараллеливанием потоков и приоритетным обслуживанием, которые не поддаются аналитическому описанию. Элементы диспетчеризации процессов, включен-

Таблица 1

Средние значения времени выполнения фрагментов двух программ, полученные по результатам измерений прототипа

Наименование функций	Среднее время выполнения, с	Комментарий
Выполнение сегментов программ 1 и 2 для запросов класса 1	0,16	Без параллельного выполнения запросов ЦП С параллельным выполнением запросов на ЦП (2 ядра) С параллельным выполнением запросов на ЦП (4 ядра)
	0,08	
	0,04	
Выполнение сегментов программ 1 и 2 для запросов класса 2	0,13	Без параллельного выполнения запросов ЦП С параллельным выполнением запросов на ЦП (2 ядра) С параллельным выполнением запросов на ЦП (4 ядра)
	0,065	
	0,0325	
Работа с внешней памятью запросов класса 1	0,4	Среднее время приведено для всех запросов
Работа с внешней памятью запросов класса 2	0,2	Среднее время приведено для всех запросов
Время подготовки запросов пользователями (запросы класса 1)	10	
Время подготовки запросов пользователями (запросы класса 2)	15	

ных в модель, отражают механизмы соответствующей операционной системы [7]. Структура модели СП, представленной в виде замкнутой сети массового обслуживания (СеМО), приведена на рис. 3. В качестве исходных данных используются результаты измерений прототипа и работы ОС. Имитационная модель СП реализована в среде системы моделирования AnyLogic.

Приведем следующие результаты моделирования процессов обслуживания разных классов запросов при различных нагрузках на СП:

- выполнена оценка влияния параметров создаваемого ПО (варианты разделения программ на параллельно выполняемые модули, число ядер процессора СП, интенсивность обращений к внешней памяти и др.) на ВВХ для пользователей, генерирующих запросы разных классов;
- определены динамические рабочие характеристики программ в критических ситуациях;
- определена производительность и требуемое число ядер процессора для обслуживания заданного потока запросов, выполнена оценка требуемого резерва производительности.

В результате моделирования получено большое число различных оценок для программ и вычислительного процесса, характеризующих принимаемые проектные решения. Приведем для иллюстрации следующие усредненные показатели для данного вида систем — ВВХ для двух классов запросов ИУС РВ и распараллеливания программ (табл. 2).

Таким образом, организация параллельного выполнения сегментов программ существенно улучшает вероятностно-временные характеристики выполнения запросов обоих классов. Так, среднее время выполнения запросов класса 1 уменьшается с 2,359 до 1.205 с), аналогично снижается также среднее квадратическое отклонение (СКО) примерно на 51 %. Подобные тенденции наблюдаются и для запросов класса 2. Однако для полной оценки ВВХ пользователей необходимо учесть также задержки прохождения запросов через другие модули и коммуникационные сети, которые нагружены выполнением других работ.

Уровень комбинированного моделирования. Данные, полученные на прототипе и имитационной модели, являются исходными для комплексного ана-

лиза ИУС РВ, погруженной в более широкую вычислительную сеть. В примере предполагается, что проектируемая система погружена в вычислительную сеть, в которой выполняются также работы других подсистем (рис. 4). Комплексный анализ работы ИУС РВ в составе такой вычислительной

Таблица 2

Влияние распараллеливания выполнения процессов на время ответа ИУС РВ

Источники запросов	Процессы, потоки	Общая нагрузка на ЦП	Среднее время ответа, с	СКО времени ответа, с
Терминальный модуль 1 (7 терминалов) Терминальный модуль 2 (5 терминалов)	Без распараллеливания	0,992	2,359 10,166	1,667 7,604
Терминальный модуль 1 (7 терминалов) Терминальный модуль 2 (5 терминалов)	С распараллеливанием	0,501	1,205 4,014	1,384 5,626

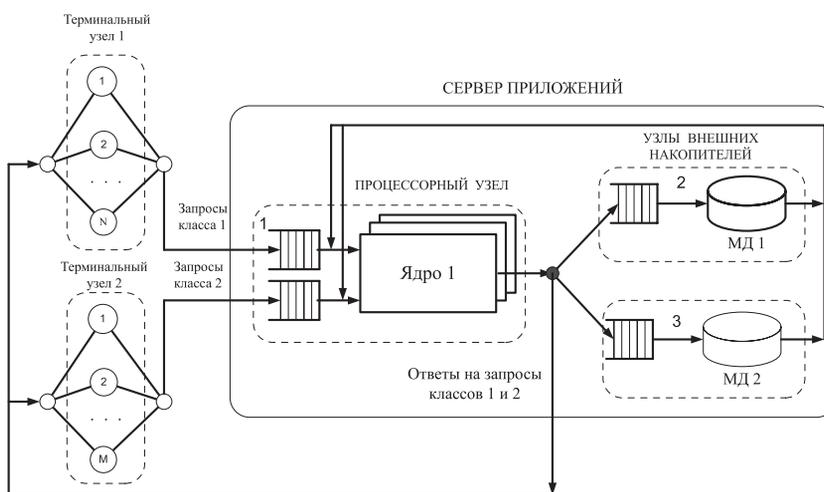


Рис. 3. Сетевая модель прототипа серверного узла

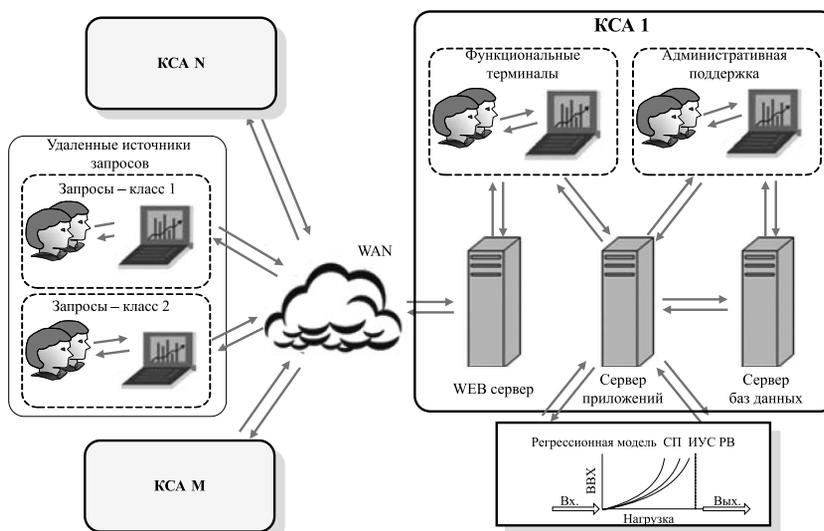


Рис. 4. Структура комбинированной имитационно-регрессионной модели информационно-вычислительной сети, включающей ИУС РВ

сети необходим для определения выполнения требований заказчика с учетом различных нагрузок сети, а также при возникновении "узких мест" или отказов. Объем требуемой для моделирования исходной информации при этом существенно возрастает. В качестве инструмента исследования использована комбинированная двухуровневая модель, в которой первый уровень представлен имитационной моделью, отражающей глобальную вычислительную сеть, состоящую из комплексов средств автоматизации, терминальных узлов и сети передачи данных (WAN). Второй уровень представлен регрессионной моделью сервера приложений ИУС РВ, которая была разработана по данным имитационного моделирования разработанного прототипа (см. выше).

В результате получен ряд оценок, позволяющий сделать выводы о работе ИУС РВ в составе территориально распределенной информационно-вычислительной сети и скорректировать при необходимости принятые решения по ее построению. Приведем для иллюстрации важные для такого рода систем оценки времени ответа для запросов первого класса при фиксированном числе клиентов терминального модуля (пять клиентов) и стабильной нагрузке вычислительной сети (табл. 3).

Расчет проведем для варианта, где число терминалов класса 1 равно 7, а класса 2 — 5 (см. табл. 2). По результатам моделирования СП построены регрессионные модели времени выполнения запросов классов 1 и 2 в зависимости от соответствующей нагрузки. Остальная часть сети моделировалась путем имитации (Any Logic). Нагрузка, создаваемая на сеть другими потоками информации, здесь не приведена. При поступлении запроса в СП время его обработки определяли из уравнения регрессии. Сравнивая результаты оценок, приведенные в табл. 2 и 3, видим, что время реакции ИУС РВ, погруженную в территориальную вычислительную сеть, существенным образом зависит также от характеристик и нагрузки на сеть. Время ответа для запросов класса 2 существенно зависит от потока

запросов класса 1 (нагрузки на СП). Помимо усредненных характеристик на комбинированной модели получены также значения квантилей времени ответа при вероятностях 0,95 и 0,99.

При проектировании системы необходимо исследовать ее работу с помощью виртуальных прототипов ИУС РВ, локального моделирования и моделирования всей системы в целом с учетом ее окружения, что и обеспечивает предложенная технология.

Приведенная технология позволяет получить большой объем других данных, который может оказаться полезным при проектировании ИУС РВ и других информационных систем.

Заключение

Разработчики ИУС РВ в течение длительного времени значительные усилия прикладывали к поиску методов и моделей, позволяющих оценивать эффективность принимаемых проектных решений на ранних стадиях жизненного цикла этих систем. Накоплен большой объем знаний и инструментов вероятностных оценок при неполной информации. Тем не менее развитие компьютерных систем и сетей ставит новые задачи их анализа и синтеза путем создания новых технологий совместной оценки разнородных решений, что невозможно без комплексного использования соответствующего инструментария. В данной работе предложен вариант формирования такой технологии путем совместного использования разработанных методов, моделей и возможностей цифрового прототипирования создаваемых систем на различных этапах проектирования. Дальнейшее развитие технологии видится в использовании более простых аналитических и экспериментальных методов к построению комбинированных моделей для выполнения оценок систем на разных стадиях их жизненного цикла.

Список литературы

1. Еремин Н. А. Управление разработкой интеллектуальных месторождений нефти и газа. М.: Изд-во РГУ нефти и газа имени И. М. Губкина, 2011. 200 с.
2. Еремин Ал. Н., Еремин Ан. Н., Еремин Н. А. Управление разработкой интеллектуальных месторождений нефти и газа. М.: Изд-во РГУ нефти и газа имени И. М. Губкина, 2012. 165 с.
3. Авен О. И., Коган Я. А. Управление вычислительным процессом в ЭВМ. М.: Энергия, 1978. 240 с.
4. Bingjian Liu. Integration of physical and virtual prototyping. Loughborough University, 2011. 252 p.
5. Авен О. И., Гурин Н. Н., Коган Я. А. Оценка качества и оптимизация вычислительных систем. М.: Наука, 1982. 464 с.
6. Литвин В. Г., Аладышев В. П., Винниченко А. И. Анализ производительности мультипрограммных ЭВМ. М.: Финансы и статистика, 1984. 160 с.
7. Таненбаум Э. Современные операционные системы. СПб.: Питер, 2010. 1120 с.
8. Литвин В. Г., Чернышова Е. В., Литвин Ю. В. Оценка показателей работы вычислительной системы реального времени, содержащей узлы с параллельным выполнением процессов // Информационные технологии и вычислительные системы. 2014. № 2. С. 16—25.

Таблица 3

ВВХ времени выполнения запросов в ИУС РВ, погруженной в сеть

Число терминалов классов 1; 2	Класс запросов	Среднее время ответа, с	СКО времени ответа, с	Время ответа с вероятностью 0,95, с	Время ответа с вероятностью 0,99, с
3; 5	Класс 1 Класс 2	0,647 0,690	1,276 1,374	3,1 3,8	6,1 6,1
5; 5	Класс 1 Класс 2	1,325 2,085	2,890 4,720	6,3 11,9	12,6 20,8
7; 5	Класс 1 Класс 2	2,722 4,727	6,089 10,533	12,5 24,3	24,8 48,4

Iu. V. Litvin, Ph.D, Director of the Center, e-mail: litvinj@simplecs.ru
The Limited Liability Company "Scientific-Research Institute of Economics
and Management in Gas Industry" (LLC "NIIgazeconomika", JSC "Gazprom"), Moscow,
K. S. Nozdrin, Head of the Scientific-Technical Center, e-mail: nozdrinks@yandex.ru,
A. A. Polkov, Leading Engineer, e-mail: andr.polkov@yandex.ru
Joint Stock Company "Research Institute for Automated Apparatus
named after Academician V. S. Semenikhin" (JSC "NIIAA"), Moscow

System of Virtual Prototyping and Modeling Management Information Systems of Real Time

New approach to assessment of probable time response characteristics created management information real-time systems (IMS RT), on the basis of the virtual prototypes, combined imitative and analytical and regression models is considered. There is a large number of examples systems belonging to the considered class. The main problem creation of similar systems consists that it is necessary to make decisions in the conditions of high uncertainty and incompleteness of information. Postponing the analysis for later time when information appears, it is necessary spend a lot of time for a redesign. Within the considered technology three basic approaches are integrated: conceptual simulation, the virtual prototyping and combined models.

Conceptual simulation is the important building tool IMS RT. However only recently special attention, in many respects thanks to the UML and IDEF languages is paid to the this approach. The support system of processes of creation of IMS RT offered in article essentially leans on the UML language.

The second element of the offered technology is use the virtual prototypes (VP). Development and the research VP already at early design stages allows to define detail requirements to the IMS RT separate components and to coordinate external interfaces with the customer, and also to fulfill the made project decisions. Dipping allows estimating at the environment VP how the developed system will behave in interaction with other systems. It is essential to help with their decision can render combined models.

The used combined models can have different character, for example, simulation and analytical models, simulation and regression and hybrid discrete and continuous models.

Use of the considered technology is shown by a simple specific example of the solution one often originating questions during creation of IMS RT.

Keywords: *conceptual model, virtual prototype, combined simulation-regression models, technology prototyping and modeling, simulation, information control system, real-time system*

References

1. **Eremin N. A.** *Upravlenie razrabotkoi intellektual'nykh mestorozhdenii nefiti i gaza*, Moscow, RGU nefiti i gaza imeni I. M. Gubkina, 2011, 200 p. (Eremin N. A. Development of intellectual oil and gas fields, Moscow, Gubkin Russian State University of Oil and Gas, 2011, 200 p.) (in Russian).
2. **Eremin Al. N., Eremin An. N., Eremin N. A.** *Upravlenie razrabotkoi intellektual'nykh mestorozhdenii nefiti i gaza*, Moscow, RGU nefiti i gaza imeni I. M. Gubkina, 2012, 165 p. (Eremin Al. N., Eremin An. N., Eremin N. A. Development of intellectual oil and gas fields, Moscow, Gubkin Russian State University of Oil and Gas, 2012. 165 p.) (in Russian).
3. **Aven O. I., Kogan Ya. A.** *Upravlenie vychislitel'nykh protsessov v EVM*, Moscow, Energiya, 1978, 240 p. (Aven O. I., Kogan Y. A. Control of the computing process in the computer, Moscow, Energy, 1978, 240 p.) (in Russian).
4. **Bingjian Liu.** *Integration of physical and virtual prototyping*, Loughborough University, 2011, 252 p.
5. **Aven O. I., Gurin N. N., Kogan Ya. A.** *Otsenka kachestva i optimizatsiya vychislitel'nykh sistem*, Moscow, Nauka, 1982, 464 p.

(Aven O. I., Gurin N. N., Kogan, Y. A. quality Assessment and optimization of computing systems, Moscow, Nauka, 1982, 464 p.) (in Russian).

6. **Litvin V. G., Aladyshev V. P., Vinnichenko A. I.** *Analiz proizvoditel'nosti mul'ti programmykh EVM*, Moscow, Finansy i statistika, 1984, 160 p. (Litvin, V. G. and V. P. Aladyshev, Vinnichenko, A. I., performance Analysis of multi programming, a computer, Moscow, Finance and statistics, 1984, 160 p.) (in Russian).

7. **Tanenbaum E.** *Sovremennyye operatsionnyye sistemy*, Sankt-Peterburg, Piter, 2010, 1120 p. (Tanenbaum A. Modern operating systems, Sankt-Peterburg, Peter, 2010, 1120 p.) (in Russian).

8. **Litvin V. G., Chernyshova E. V., Litvin Yu. V.** *Otsenka pokazatelei raboty vychislitel'noi sistemy real'nogo vremeni, soderzhashchei uzly s parallel'nym vypolnieniem protsessov*, *Informatsionnye tekhnologii i vychislitel'nye sistemy*, 2014, no. 2, pp. 16–25. (Litvin V. G., Chernyshova E. V., Litvin Yu. V. Assessment of the performance of computing real-time systems containing nodes with parallel execution of processes // Information technologies and computing systems, 2014, no. 2, pp. 16–25) (in Russian).

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.67

Чан Ван Нгиа, аспирант, e-mail: nghiamosmpt@gmail.com
Московский физико-технический институт (государственный университет)

Разработка параллельной архитектуры кодеров БЧХ и LDPC в стандартах систем цифрового телевизионного вещания второго поколения с регулируемыми параметрами кодирования на базе ПЛИС

В стандартах систем цифрового телевизионного вещания второго поколения, таких как DVB-T2, DVB-S2, DVB-C2, и др., использована система эффективного канального кодирования для высокой помехоустойчивости при передаче данных по неидеальным каналам связи с ограниченной пропускной способностью за счет последовательной конкатенации кода БЧХ (Боуза—Чоудхури—Хоквингема) и кода с низкой плотностью проверок на четность (LDPC). Требования к высокой скорости обработки информации, большой блок данных и многопараметрическое кодирование представляют сложную задачу для реализации эффективной аппаратной архитектуры. В данной работе предложен способ распараллеливания кодеров БЧХ и LDPC с регулируемой корректирующей способностью, поддерживающий все возможные конфигурации кодов БЧХ + LDPC.

Ключевые слова: помехоустойчивое кодирование, кодер БЧХ, кодер LDPC, поле Галуа, DVB, ПЛИС, DVB-T2 модулятор

Введение

Системы помехоустойчивого кодирования — кодирования с опережающей коррекцией ошибок (*forward error correction*) играют решающую роль в современных цифровых системах связи и запоминающих устройствах. В частности, в стандартах систем цифрового телевизионного вещания (DVB) второго поколения используется сочетание внешнего кодирования БЧХ и внутреннего кодирования LDPC [1–4], которые обеспечивают согласование источника информации с каналом связи и имеют возможность вплотную приблизиться к границе Шеннона. В стандартах DVB используются два формата данных (64 800 битов для основных кадров и 16 200 битов для сокращенных кадров) и 11 различных кодовых скоростей (1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9 и 9/10). Для каждой кодовой скорости корректирующая способность разная, так как используются разные порождающие полиномы (для кодера БЧХ) и проверочные матрицы (для кодера LDPC). Следовательно, устройство, обеспечивающее все возможные варианты кодирования, достаточно сложно для аппаратной реализации. Поэтому целью настоящей работы является разработка параллельных реконфигурируемых кодеров БЧХ и LDPC для удовлетворения различных требований к корректирующей способности и устранения "узких мест".

1. Новая параллельная архитектура кодеров БЧХ

Кодирование БЧХ, описанное в работах [1–4], осуществляется следующим образом:

$$\frac{m(x)x^n}{g(x)} = q(x) + \frac{r(x)}{g(x)}, \quad (1)$$

где $m(x) = m_{\text{кБЧХ}-1}x^{\text{кБЧХ}-1} + m_{\text{кБЧХ}-2}x^{\text{кБЧХ}-2} + \dots + m_1x + m_0$ — полином входного сообщения, представляющий $M = (m_{\text{кБЧХ}-1}, m_{\text{кБЧХ}-2}, \dots, m_1, m_0)$ информационных битов; $g(x) = x^n + g_{n-1}x^{n-1} + \dots + g_2x^2 + g_1x + 1$ — порождающий полином степени n над полем Галуа; $r(x)$ — остаток от деления $m(x)x^n$ на $g(x)$; $q(x)$ — частное от деления $m(x)x^n$ на $g(x)$.

Сформированное на выходе кодовое слово представляется в виде полинома:

$$c(x) = m(x)x^n + r(x). \quad (2)$$

Коды БЧХ относятся к циклическому кодированию, поэтому аппаратная реализация кодера БЧХ (реализация выражений (1) и (2)) выполняется с помощью регистра сдвига длины n с линейной обратной связью (рис. 1), определенного порождающим полиномом $g(x)$. Все операции сложения проводятся по правилам арифметики по модулю 2

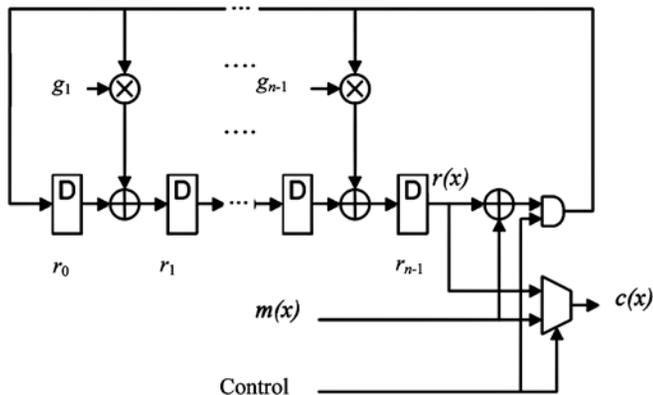


Рис. 1. Традиционная схема кодера БЧХ

(XOR). Операции умножения выполняются с помощью элемента "И" (AND).

Первые $k_{\text{БЧХ}}$ тактов работы данные сообщения поступают на вход регистра сдвига для вычисления остатка $r(x)$, и одновременно проходят на выход. Последующие n тактов происходит выгрузка остатка $r(x)$ из регистра сдвига.

Архитектура регистра сдвига имеет ограничения по производительности вследствие последовательной архитектуры. Видно, что если кодер БЧХ может обрабатывать одновременно w бит, то число тактов его работы уменьшается в w раз. Для реализации параллельного алгоритма кодера БЧХ необходимо рассмотреть зависимость состояния регистра сдвига остатка в момент $T + w$ от состояния регистра остатка в момент T и от w входных информационных битов. По схеме рис. 1 состояние регистра сдвига остатка в моменты T и $T + 1$ и $(T + 1)$ -й входной информационной бит $m(T + 1)$ связаны соотношениями:

$$\begin{cases} r_{n-1}(T+1) = r_{n-2}(T) + g_{n-1}[r_{n-1}(T) + m(T+1)] \\ r_{n-2}(T+1) = r_{n-3}(T) + g_{n-2}[r_{n-1}(T) + m(T+1)] \\ \dots\dots\dots \\ r_1(T+1) = r_0(T) + g_1[r_{n-1}(T) + m(T+1)] \\ r_0(T+1) = [r_{n-1}(T) + m(T+1)]. \end{cases} \quad (3)$$

Приведенные выше уравнения можно переписать в виде матрицы:

$$\begin{bmatrix} r_{n-1}(T+1) \\ r_{n-2}(T+1) \\ \dots\dots\dots \\ r_1(T+1) \\ r_0(T+1) \end{bmatrix} = \begin{bmatrix} g_{n-1} & 1 & 0 & 0 & \dots & 0 \\ g_{n-2} & 0 & 1 & 0 & \dots & 0 \\ \dots\dots\dots \\ g_1 & 0 & 0 & 0 & \dots & 1 \\ g_0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \times \begin{bmatrix} r_{n-1}(T) + m(T+1) \\ r_{n-2}(T) + 0 \\ \dots\dots\dots \\ r_1(T) + 0 \\ r_0(T) + 0 \end{bmatrix}. \quad (4)$$

Для упрощения выражения (4), обозначим:

$$\begin{cases} R(T+1) = [r_{n-1}(T+1)r_{n-2}(T+1) \dots \\ \dots r_1(T+1)r_0(T+1)]^T, \\ R(T) = [r_{n-1}(T)r_{n-2}(T) \dots r_1(T)r_0(T)]^T \\ M(T+1) = [m(T+1)0 \dots 0 0]^T \end{cases}$$

$$\text{и} \quad F = \begin{bmatrix} G I_{n-1} \\ 0 \end{bmatrix} = \begin{bmatrix} g_{n-1} & 1 & 0 & \dots & 0 \\ g_{n-2} & 0 & 1 & \dots & 0 \\ \dots\dots\dots \\ g_1 & 0 & 0 & \dots & 1 \\ g_0 & 0 & 0 & \dots & 0 \end{bmatrix},$$

где $G = [g_{n-1} g_{n-2} \dots g_1 g_0]^T$ — коэффициенты порождающего полинома $g(x)$; I_{n-1} — единичная матрица размерностью $(n-1) \times (n-1)$.

Выражение (4) можно записать в виде:

$$\begin{aligned} R(T+1) &= F[R(T) + M(T+1)] = \\ &= FR(T) + m(T+1)G. \end{aligned} \quad (5)$$

Состояние регистра сдвига остатка в момент $T + w$ получено рекурсивным методом:

$$R(T+w) = F^w[R(T) + M(T+w)], \quad (6)$$

где $M(T+w) = [m(T+1)m(T+2)\dots m(T+w)0\dots 0 0]^T$ — вектор группы по w входных информационных битов;

$$F^w = \begin{bmatrix} F^{w-1} \times G | F^{w-2} \times G | \dots | F \times G | \text{Первые } (n-w+1) \text{ столбцов матрицы } F \end{bmatrix}. \quad (7)$$

Для любой матрицы A размерностью $(n \times n)$ при умножении на матрицу F получим

$$A \times F = [A \times G | \text{Первые } (n-1) \text{ столбцов матрицы } A]. \quad (8)$$

Для доказательства выражений (6) и (7) рассмотрим их при $w = 2$. Из выражений (5) и (8) получим

$$\begin{aligned} R(T+2) &= F \times R(T+1) + m(T+2)G = \\ &= F^2[R(T) + M(T+1)] + m(T+2)G; \end{aligned} \quad (9)$$

$$F^2 = F \times F = \begin{bmatrix} FF \times G | \text{Первые } (n-1) \text{ столбцов матрицы } F \end{bmatrix}; \quad (10)$$

$$\begin{aligned} & m(T+2)G = \\ &= \begin{bmatrix} \text{Первый столбец} \\ \text{с любыми значениями} \end{bmatrix} | \begin{bmatrix} G | (n-2) \text{ столбцов} \\ \text{с любыми значениями} \end{bmatrix} \times \\ & \times [0 \ m(T+2) \ 0 \ 0 \ \dots \ 0]^T = F^2 M^{(2)}(T+2), \end{aligned}$$

где $M^{(2)}(T+2) = [0 \ m(T+2) \ 0 \ 0 \ \dots \ 0]^T$ — вектор размерности n , второй элемент которого равен информационному биту $m(T+2)$, остальные — нулям.

Выражение (9) можно переписать:

$$\begin{aligned} R(T+2) &= F^2[R(T) + M(T+1) + M^{(2)}(T+2)] = \\ &= F^2[R(T) + M(T+2)], \end{aligned} \quad (11)$$

где $M(T+2) = [m(T+1) \ m(T+2) \ 0 \dots \ 0 \ 0]^T$ — вектор группы по два входных информационных бита:

$$M(T+2) = [m(T+1) \ 0 \ 0 \dots \ 0]^T + [0 \ m(T+2) \ 0 \ 0 \dots \ 0]^T = M(T+1) + M^{(2)}(T+2).$$

Из выражений (10) и (11) видно, что выражения (6) и (7) верны при $w = 2$. Допустим, что выражения (6) и (7) также верны при $w = k$, рассмотрим эти выражения при $w = k + 1$. Имеем:

$$F^k = \left[F^{k-1} \times G \mid F^{k-2} \times G \mid \dots \mid F \times G \mid \begin{array}{l} \text{Первые } (n-k+1) \\ \text{столбцов матрицы } F \end{array} \right];$$

$$F^{k+1} = \left[F^k \times G \mid \begin{array}{l} \text{Первые } (n-1) \\ \text{столбцов } F^k \end{array} \right] =$$

$$= \left[F^k \times G \mid F^{k-1} \times G \mid \dots \mid F \times G \mid \begin{array}{l} \text{Первые } (n-k) \\ \text{столбцов матрицы } F \end{array} \right]. \quad (12)$$

Из выражения (12) видим, что матрица F^{k+1} содержит в $(k+1)$ -м столбце вектор G . Поэтому

$$m(T+k+1)G =$$

$$= \left[\begin{array}{l} k \text{ столбцов с любыми} \\ \text{значениями} \end{array} \mid G \mid \begin{array}{l} (n-k+1) \text{ столбцов} \\ \text{с любыми значениями} \end{array} \right] \times$$

$$\times [0 \dots 0 \ m(T+k+1) \ 0 \dots 0]^T =$$

$$= F^{k+1} M^{(k+1)}(T+k+1), \quad (13)$$

где $M^{(k+1)}(T+k+1) = [0 \dots 0 \ m(T+k+1) \ 0 \dots 0]^T$ — вектор размерности n , $(k+1)$ -й элемент которого равен информационному биту $m(T+k+1)$, остальные элементы равны нулям.

Для $w = k + 1$ имеем:

$$R(T+k+1) = F \times R(T+k) + m(T+k+1)G =$$

$$= F^{k+1}(R(T) + M(T+k)) + m(T+k+1)G. \quad (14)$$

Подставляем (12) и (13) в выражение (14):

$$R(T+k+1) =$$

$$= F^{k+1}[R(T) + M(T+k) + M^{(k+1)}(T+k+1)] =$$

$$= F^{k+1}[R(T) + M(T+k+1)]. \quad (15)$$

Из выражений (12) и (15) видно, что выражения (6) и (7) также верны для $w = k + 1$. По методу математической индукции выражения (6) и (7) применены для $w \leq n$.

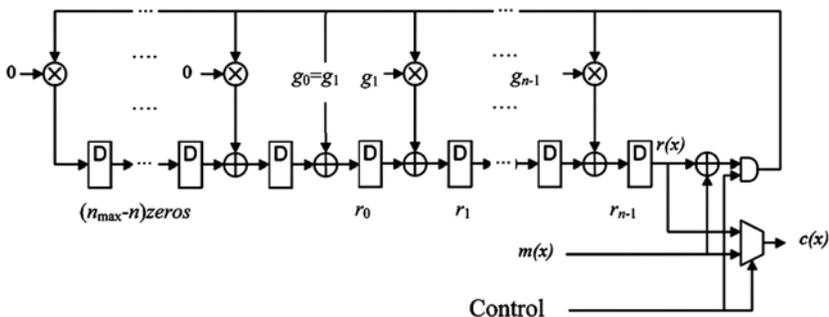


Рис. 2. Кодер БЧХ с регулируемой корректирующей способностью

В стандартах DVB кодер БЧХ выполняется в поле Галуа $GF(2^{16})$ (для обычных кадров) и $GF(2^{14})$ (для сокращенных кадров) и имеет способность корректировать $t = \{8, 10, 12\}$ ошибок. Для изменения корректирующей способности необходимо изменить порождающий полином (следует изменить матрицу F^w), что ведет к изменению аппаратной архитектуры. Для реализации схемы кодера БЧХ с реконфигурируемым порождающим полиномом, хранящимся в перезаписываемых регистрах:

- ✓ Размерность $n \times n$ матрицы F , длина регистра остатка и размерность вектора G расширены до максимальной степени n_{max} порождающего полинома, использованной в стандартах DVB.
- ✓ Для порождающего полинома $g(x)$ меньшей степени ($n < n_{max}$) значащий коэффициент при самой старшей степени полинома должен быть на месте старшего регистра, а $(n_{max} - n)$ добавленных младших коэффициентов необходимо обнулить. На рис. 2 показано, что n бит остатка не изменяются. Таким образом, использование нового порождающего полинома $g_{new}(x) = g(x)x^{(n_{max}-n)}$ дает те же результаты, что использование полинома $g(x)$.

2. Параллельная архитектура кодера LDPC с регулируемой кодовой скоростью

В последнее время кодеры LDPC получили широкое распространение в современных стандартах передачи данных и цифрового вещания за счет своей высокой эффективности. Конструкция кодеров LDPC представляет большую проблему вследствие своей огромной размерности. Было разработано достаточно большое число разнообразных алгоритмов построения проверочных матриц H . В стандартах DVB [1–5] использован нерегулярный код повторения накопления благодаря использованию компактных форм хранения самих проверочных матриц. Нерегулярный код, как правило, изображен графом Таннера и может быть представлен в виде двух подматриц:

$$H = [H_u \ H_p] =$$

$$= \left[\begin{array}{cccc|cc} h_{11} & h_{12} & \dots & h_{1K} & 1 & \\ h_{21} & h_{22} & \dots & h_{2K} & & 1 \\ \dots & & & & H_u & \dots \\ h_{M1} & h_{M2} & \dots & h_{MK} & & H_p \\ & & & & & 1 \end{array} \right] \quad (16)$$

$$\left\{ \begin{array}{l} [c_1 + (i-1)q] \bmod (n-k) \\ [c_2 + (i-1)q] \bmod (n-k) \\ \dots \\ [c_{wl} + (i-1)q] \bmod (n-k). \end{array} \right. \quad (17)$$

Для построения матрицы H_u информационные узлы делятся на q групп, каждая из которых имеет $Q = 360$ бит.

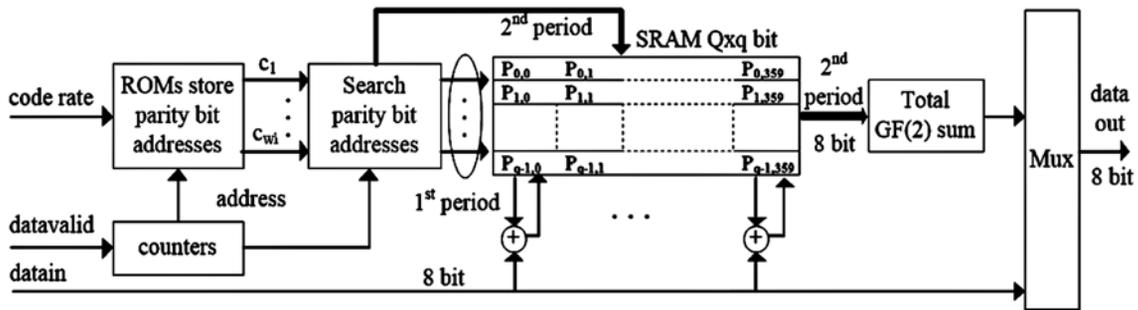


Рис. 3. Архитектура кодера LDPC

При этом нужно только указать проверочные узлы (с индексами c_1, c_2, \dots, c_{w_i}), соединенные с первым информационным узлом в рассматриваемой группе, тогда индексы проверочных узлов любого i -го информационного узла в группе определяются по формулам (17), где k — число входных информационных битов; n — длина кодового слова; $q = (n - k)/Q$.

Однако для различных кодовых скоростей и размеров блоков число групп в матрице отличается. Для реализации параллельной архитектуры кодера LDPC с регулируемой кодовой скоростью и размером блока предлагаемый способ представлен на рис. 3, где память SRAM спроектирована для максимального значения q ($q_{\max} = 135$), используемого в стандартах DVB.

Проверочные узлы для первого битового узла каждой из групп для всех кодовых скоростей сохранены в постоянной памяти ROM. Память SRAM размерности ($Q \times q_{\max}$) бит хранит проверочные биты в вычислении. Работа кодера LDPC состоит из двух этапов. На первом этапе на основе индексов (c_1, c_2, \dots, c_{w_i}) проверочных узлов, соответствующих каждой из q групп Q информационных битов и значения счетчика (*counter*) входных информационных байтов, считывается w_i групп по восемь проверочных битов, которые суммируются на информационном байте по модулю 2. Результаты сложения перезаписываются в SRAM. Второй этап выполняется после поступления q информационных групп в кодер LDPC. На этом этапе восемь проверочных битов последовательно считываются по столбцам q первых строк и поступают в сумматор (Total GF(2) sum) для вычисления по формуле

$$\begin{bmatrix} p_k \\ p_{k+1} \\ \dots \\ p_{k+7} \end{bmatrix} = \begin{bmatrix} p_k \\ p_{k+1} \\ \dots \\ p_{k+7} \end{bmatrix} \oplus \begin{bmatrix} p_{k-1} \\ \dots \\ p_{k+6} \end{bmatrix}. \quad (18)$$

В результате проверочные биты ($p_k, p_{k+1}, \dots, p_{k+7}$) передаются на выход кодера LDPC и бит p_{k+7} будет использован для следующего вычисления.

3. Экспериментальные результаты

Предложенная архитектура кодеров БЧХ и LDPC протестирована в DVB-T2 модуляторе [6], реализованном на чипе Xilinx Kintex-7 XC7K325T платы NetFPGA-1G-CML Development Kit и на плате AD9789 Evaluation Board (рис. 4). Двоичная последовательность коэффициентов порождающего полинома в шестнадцатеричной системе показана в табл. 1.

Кодеры БЧХ и LDPC распараллелены по 8-разрядной шине данных, поэтому мы используем матрицу F^8 для вычисления кода БЧХ. Последовательность элементов восьми первых столбцов матрицы F^8 показана в табл. 2.

Для моделирования неидеального канала связи аддитивный белый гауссовский шум генерируется в чипе ПЛИС. Радиочастотные сигналы без адди-

Таблица 1
Последовательность коэффициентов порождающего полинома в стандартах DVB

Коэффициенты порождающего полинома	Степени $g(x)$
1C07255F712797BD19FC6D7504F9662B	128
60150CEDFC2A331F6A785703EFD12301B8BB6591	160
4E260E83845C511C50CF2CD8DC350889034785F7660255E7	192
4062DBEA9869B262CD23A39069528FE7D7D11905A5	168

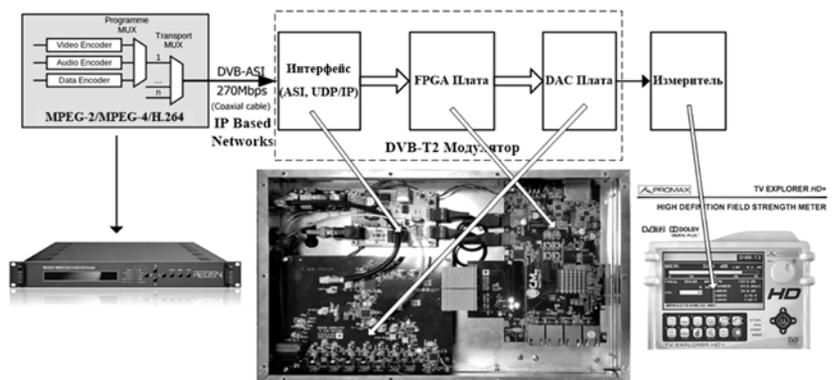


Рис. 4. Экспериментальный DVB-T2 модулятор

Последовательность элементов восьми первых столбцов матрицы F^8

Элементы восьми первых столбцов матрицы F^8	Степень
B85CAE57937180C0E070B8DCEE7703B964B25994CA658A459A4D1E0F3F272B2D2E17B3E1C8E472392412897C3E9FF74319341A8D7E3FA76B8DFEFFF479B758241188CC66309BC5EAFEFECF5F97F3C1582C964B9DF6FBC5DA6D0E073BA56AB562312090482412093C9E4F9FF74399F4FAFD46A3E9CC66332128140A857A3D269371	128
4EA79D0080402090C8E47239D269FA7D70B85C2E97858C46235F61FEFFB1160B4BEB3B53E7BD90C8E472395229DAEDB85CAED7A51C8EC72D58AC562BDB23DF21DEEFB912898A45EC763B53E7BD90C864B2D9A2D1A6D3A71DC060B058AC562B5B63FFB196CBAB9B03CFA99ACDA8D4EAF5B4DAED389C4EA79D80C0E0F0F8FC7E3FD12693878D8844221146235FE13E1F416EB795048241EE77F5349A4DE8743A9D	160
DAED2C964B7F6568349A4D7CBE5F7560B0D86CB65B7761EA7560B0582C168B9F155028944A25C8643299964BFFA508844221CAE528140A05582C964BFFA50884C2E1AA55F078BCDEEFAD8CC6E32B4FFD2412095EAF8D9C4E27497E3FC5381C0E8799168B1F5570B85CAED731C2E12A15502894CAE52894CA65E8743A1DD46A35C0E070B8DCEEF7A10A85188C46A38B9F9590C8E4F2796633C3BB0759F6FBA7095E2F4D7CBEDFB58040A0D0E8F47A3DC4E2F1A2517239C6636B6FED2C160BDFB5	192
AA5580C0E0F0F87CBEDFC5482492498EC7C9CE6719261323BB7711A2D1C2E15AADFC7E3FB570B85C2E97E1DA6D1C0E87E95E2F3DB45AAD7CBEDFC548A4D2691E8FED5C2E17A1FA7D94CAE5D8ECF6FBD7C1CAE5582C160BAFFD542A95E070B85C2E1721BADD44A2D14221BA5D0482C1CA6598CC66B3F3D3C3BCFC4D2613233B373132192613A3FBD7410A0528944AA578BC5EAFDD542A152090482412092E1721BA5D040201AA55	168

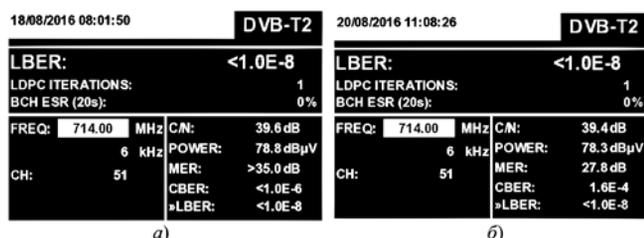


Рис. 5. Измерительные результаты на телевизионном анализаторе TV EXPLORER HD+

тивного белого шума и с шумом на выходе DVB-T2 модулятора рассматриваются на телевизионном анализаторе PROMAX TV EXPLORER HD+ и приведены на рис. 5, а и рис. 5, б соответственно.

В случае отсутствия шума СBER (вероятность ошибки для цифрового сигнала перед коррекцией ошибок) и LBER (коэффициент битовых ошибок для цифрового сигнала после коррекции ошибок) меньше $1 \times E-6$ и $1 \times E-8$, соответственно и MER (коэффициент ошибок модуляции) больше 35 дБ. Эти значения являются граничными значениями анализатора TV EXPLORER HD+. В случае присутствия шума СBER (вероятность ошибки) повышается (примерно $E-3 \div E-6$) и MER уменьшается, но $LBER < 1 \times E-8$. Это значит, что битовые ошибки, возникшие вследствие шума, откорректированы полностью. Результаты экспериментов на ПЛИС показали правильность предложенного метода проектирования БЧХ- и LDPC-кодеров и их эффективность. Очевидно, что эти кодеры имеют полную совместимость с декодерами на приемной стороне и могут заменить традиционные схемы кодирования [7], используемые в стандартах DVB и т. п.

Заключение

В работе представлен новый способ построения параллельных реконфигурируемых кодеров БЧХ и

LDPC, основанный на традиционной схеме кодеров. Их реконфигурируемость (параметризация) универсальна, и дает возможность оптимизации аппаратной реализации. Параллельная реализация этих кодеров приводит к выигрышу по быстродействию и к большей пропускной способности. Результаты экспериментов на ПЛИС показали верность предложенного метода и легкость интегрирования в системах стандартов DVB и т. п.

Список литературы

1. **European Standard** ETSI EN 302 307-1 V1.4.1 (2014-11). Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broad-band satellite applications. URL: <https://www.dvb.org/standards/dvb-s2>
2. **European Standard** ETSI EN 302 755 V1.4.1 (2015-07): Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system. URL: <https://www.dvb.org/standards/dvb-t2>
3. **European Standard**. ETSI EN 302 755 V1.3.1 (2015-10): Frame structure channel coding and modulation for a second generation digital transmission system for cable systems. URL: <https://www.dvb.org/standards/dvb-c2>
4. **Дворкович В. П., Дворкович А. В.** Цифровые видеотрансмиссионные системы (теория и практика). М.: Техносфера, 2012. 1008 с.
5. **Gomes M., Falcao G., Sengo A., Ferreira V., Silva V., Falcao M.** High Throughput Encoder Architecture for DVB-S2 LDPC-IRA Codes // IEEE International Conference on Microelectronics. 29–31 Dec. 2007. P. 271–274.
6. **Чан Ван Нгиа, Ле Ван Ки, Нгуен Хьюй Хоанг.** Реализация передатчика системы наземного цифрового телевидения второго поколения DVB-T2 на FPGA // Сборник трудов 17-й Международной конференции "Цифровая обработка сигналов и ее применение — DSPA-2015". Т. 2. 25–27 марта 2015 г. Москва. М.: ИПУ РАН. С. 690–693.
7. **Чан Ван Нгиа, Ле Ван Ки, Чан Минь Хай, Ле Тхи Чанг Линь.** Реализация модулей кодирования коррекции ошибок и демультимплексирования битов по ячейкам системы наземного цифрового телевидения второго поколения DVB-T2 на FPGA // Сборник трудов 17-й Международной конференции "Цифровая обработка сигналов и ее применение — DSPA-2015". Т. 2. 25–27 марта 2015 г., г. Москва. М.: ИПУ РАН. С. 687–690.

Development of the Parallel BCH and LDPC Encoders Architecture for the Second Generation Digital Video Broadcasting Standards with Adjustable Encoding Parameters on FPGA

In the second generation digital video broadcast standards, such as DVB-T2, DVB-S2, DVB-C2, etc., is applied a powerful channel coding scheme to transmit data on the non-ideal communication channels with limited bandwidth due to the serial concatenation of BCH (Bose-Chaudhuri-Hocquenghen) and Low-Density-Parity-Check (LDPC) codes. The high-speed requirements, large data block lengths and multi-parametric encoding present complex challenges in the efficient implementation of hardware architecture. This paper proposes a new approach to parallel implementation of BCH and LDPC encoders with adjustable encoding parameters, supporting all the different BCH + LDPC code configurations. The proposed solution is fully backward compatible to legacy decoder on the receiving side.

Keywords: forward error correction code, BCH encode, LDPC encode, Galois field, DVB, FPGA, DVB-T2 modulator

References

1. **European Standard**, ETSI EN 302 307-1 V1.4.1, 2014-11, Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broad-band satellite applications. URL: <https://www.dvb.org/standards/dvb-s2>
2. **European Standard**, ETSI EN 302 755 V1.4.1 (2015-07): Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system. URL: <https://www.dvb.org/standards/dvb-t2>
3. **European Standard**, ETSI EN 302 755 V1.3.1 (2015-10): Frame structure channel coding and modulation for a second generation digital transmission system for cable systems. URL: <https://www.dvb.org/standards/dvb-c2>
4. **Dvorkovich V. P., Dvorkovich A. V.** *Cifrovye videoinformacionnye sistemy (teorija i praktika)*, Moscow, Tehnosfera, 2012, 1008 p. (in Russian).
5. **Gomes M., Falcao G., Sengo A., Ferreira V., Silva V., Falcao M.** High Throughput Encoder Architecture for DVB-S2 LDPC-IRA Codes, *IEEE International Conference on Microelectronics*, 29–31 Dec. 2007, pp. 271–274.
6. **Tran Van Nghia, Le Van Ky, Nguyen Huy Hoang.** Realizacija peredatchika sistemy nazemnogo cifrovogo televidenija vtorogo pokolenija DVB-T2 na FPGA, Proc. 17-ja Mezhdunarodnaja konferencija Cifrovaja obrabotka signalov i ee primenenie — DSPA-2015, Moscow, 25–27 marta 2015, IPU RAN, Moscow, vol. 2, pp. 690–693 (in Russian).
7. **Tran Van Nghia, Le Van Ky, Tran Minh Hai, Le Thi Trang Linh.** Realizacija modulej kodirovanija korekcii oshibok i demul'tipleksirovanija bitov po jachejkam sistemy nazemnogo cifrovogo televidenija vtorogo pokolenija DVB-T2 na FPGA, Proc. of 17-ja Mezhdunarodnaja konferencija Cifrovaja obrabotka signalov i ee primenenie — DSPA-2015, Moscow, 25–27 marta 2015, IPU RAN, Moscow, vol. 2, pp. 687–690.

УДК 519.674

А. П. Кудряшов, мл. науч. сотр., e-mail: kudryashovA@dvo.ru,
Институт автоматизации и процессов управления ДВО РАН, г. Владивосток,
И. В. Соловьев, аспирант, e-mail: igorek.solovyev@mail.ru,
Институт прикладной математики ДВО РАН, г. Владивосток

Реконструкция городской обстановки с учетом рельефа местности с использованием топографического плана и сервисов Google Maps

Описывается подход для реконструкции сцены городской обстановки на основе данных топографического плана. Предлагаются алгоритмы выделения и распознавания контуров зданий, надписей и других объектов. Описывается метод построения трехмерного рельефа сцены с географической привязкой на основе сервисов Google Maps. Также предлагается метод автоматического текстурирования зданий на основании набора фотоснимков местности. Приводятся вычислительные эксперименты с использованием реальных топографических планов г. Владивостока.

Ключевые слова: трехмерная реконструкция, топографический план, географические координаты, текстурирование, городская обстановка

Введение

Трехмерная реконструкция объектов реального мира является актуальной задачей компьютерной графики и машинного зрения. Полученные объекты широко используются в научных, служебных и развлекательных целях. Одна из важных практических задач — реконструкция городского пространства. Такие трехмерные модели позволяют ориентироваться на местности, учитывать существующую композицию объектов при проектировании новых, а также для туристической и развлекательной сфер деятельности.

Городское пространство состоит из множества классов объектов, которые являются как его обязательными составляющими, так и дополняющими для придания сцене большей реалистичности. В зависимости от вида прикладной задачи состав классов может варьироваться, однако к общему списку классов объектов относятся рельеф, здания, деревья, кусты, светофоры, уличные фонари, автомобили, люди и др.

Уровни детализации моделируемой сцены и отдельных ее объектов различаются для разных прикладных задач. Так, например, здания могут быть смоделированы вплоть до воссоздания своей точной трехмерной формы или же представлять собой некоторый геометрический примитив. Текстуры зданий могут являться частью фотоизображения реального объекта или иметь некий условный вид для определенного типа строений. Во всех этих

случаях требуется соблюдать баланс между детализацией сцены и временем ее реконструкции.

Ручное моделирование трехмерной сцены городского пространства с высокой степенью детализации требует огромных затрат времени и сил. Автоматизация реконструкции позволяет значительно сократить время, но обладает рядом тех или иных недостатков.

На сегодняшний день существует ряд методов, направленных на решение задачи трехмерной реконструкции объектов городской обстановки с использованием фотоснимков в качестве исходных данных, в основе которых лежат алгоритмы выделения геометрических примитивов [1, 2]. Другим подходом является реконструкция городской сцены по топографическому плану, который, как правило, содержит достаточное количество полезной для моделирования информации. При таком подходе требуется выделить контуры, принадлежащие зданиям, условно изображенных на топоплане, распознать служебную информацию о них, а также найти все дополнительные объекты, такие как светофоры, уличные фонари, деревья, клумбы и т. д. Далее эти данные можно преобразовать в трехмерное представление и нанести текстуры для большей реалистичности. Наличие данных о географических координатах объектов позволяет получить набор дополнительных данных о сцене, например рельефе, с использованием различных облачных сервисов. Предлагаемый метод является развитием более ранних работ авторов [3, 4].

Существующие методы реконструкции по топографическому плану

Существует ряд отечественных и зарубежных работ, посвященных методам выделения информации с топографических планов. В статье [5] авторы предлагают метод, в котором распознаются контуры, однако не распознается вложенная служебная информация, что не позволяет точно идентифицировать здание и определить его тип и этажность. В статье [6] авторы описывают метод реконструкции зданий по топографическому плану без автоматизированного алгоритма. Построение трехмерной модели городской сцены осуществляется с помощью сторонних коммерческих продуктов, и на каждом этапе реконструкции требуется участие оператора.

Автоматизированный алгоритм реконструкции городской сцены предлагают авторы в статье [7]. Метод основан на использовании системы LIDAR для лазерного сканирования трехмерного пространства и сопоставлении полученного облака точек с топографическим планом на основании элементов тензорного анализа. Метод не предусматривает автоматического текстурирования моделей, к тому же требует наличия сложного измерительного прибора. В статье [8] авторы предлагают похожий подход к реконструкции на основании тех же данных. В последнем случае происходит поиск простого трехмерного геометрического примитива, который подходит под данные на топографическом плане и под данные об облаке точек, полученных с помощью LIDAR. Метод также не проводит текстурирование трехмерной модели и не учитывает рельеф местности.

Исходные данные и постановка задачи

Реконструкция городского пространства с высокой детализацией — это комплексная задача, которая решается путем объединения решений ряда подзадач. Каждая такая подзадача имеет свой набор исходных данных, поэтому для детальной реконструкции выберем компоненты, приведенные ниже.

Топографический план — черно-белое растровое изображение (рис. 1). Информация на топографическом плане содержит данные о геометрической форме оснований зданий, числе этажей в них, типе строений (кирпичные, жилые, нежилые, смешанные и т. п.), расположении таких объектов, как деревья, кусты, светофоры, уличные фонари.

На изображении также присутствует информация о рельефе территории в виде данных о высоте в различных точках и изолиниях. Однако из-за тесного переплетения данных о рельефе с другими сведениями, не представляется возможным провести реконструкцию рельефа с достаточной степенью достоверности. Поэтому для воссоздания рельефа будет использоваться другой источник.

Геолокационные данные об объектах. Для реконструкции рельефа территории необходимо иметь данные хотя бы о двух парах географических координат, которые сопоставлены с точками топографического плана. На основании этих данных могут быть определены географические координаты любой точки топографического плана. Для нахождения высоты над уровнем моря используется сервис службы Google Maps Elevation API.

Фотоснимки зданий. Для нанесения реалистичных текстур на здания используют их фотоснимки. Некоторые современные фотоаппараты обладают

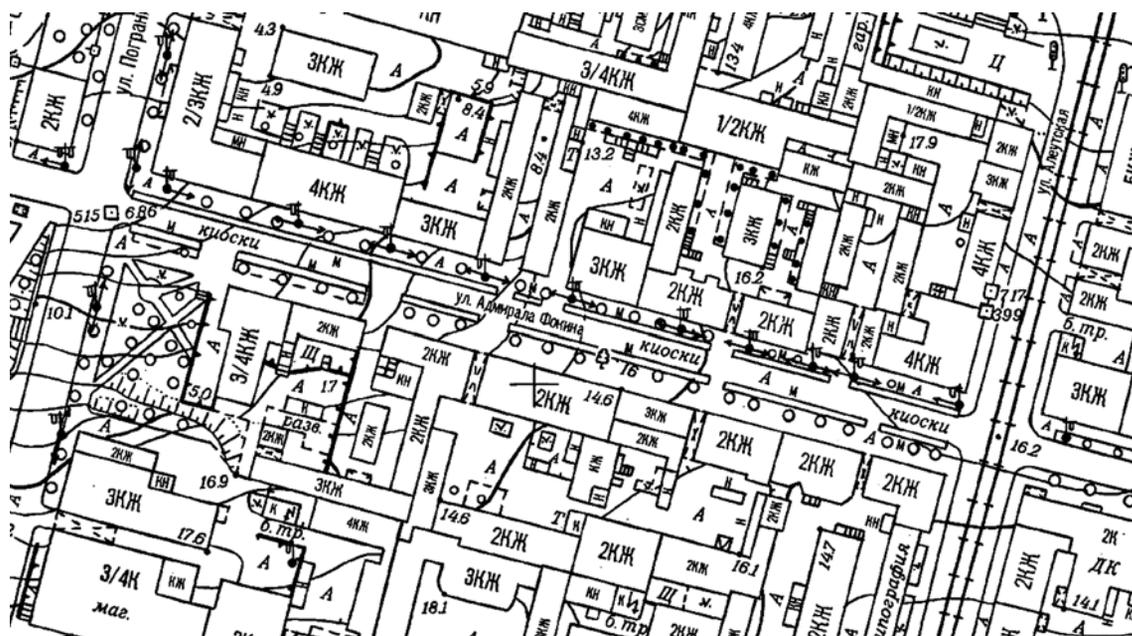


Рис. 1. Топографический план

рядом полезных функций, например, они способны получать географические координаты места съемки и направление камеры по электронному компасу. На основании совокупности этих данных будет проводиться текстурирование зданий.

Таким образом, при наличии всех описанных выше исходных данных реконструированная сцена может включать следующие объекты: текстурированные здания, рельеф, деревья, кусты, уличные фонари, светофоры.

Выделение объектов на топографическом плане

На топографическом плане здания представлены в виде замкнутой ломаной линии, которая соответствует геометрической форме основания. Как правило, здания имеют небольшое число стен. На практике установлено, что число зданий с 13 и более сторонами крайне мало. То же самое можно сказать и о зданиях с тремя стенами. Таким образом, допустимое для здания число стен будет в диапазоне между 4 и 12. В большинстве случаев угол между стенами равен 90°.

Для нахождения контуров зданий на топографическом плане используют модификацию волнового алгоритма [9]. Он позволяет с высокой точностью на бинарном изображении выделить контуры объектов. Данные контуры представляют собой плотную последовательность точек. Эта последовательность фильтруется путем исключения избыточных точек. Так, например, удаляются точки, которые с заданной точностью лежат на одной прямой. Затем происходит выравнивание, поскольку большинство углов здания равны 90°. В результате останется только минимальный набор точек, который характеризует отдельный контур.

Внутри контура каждого здания должна находиться дополнительная информация о строении. Эта информация может содержать данные о числе этажей и типе здания. Типы зданий характеризуются материалом постройки и заселенностью. Возможны следующие виды материалов зданий: кирпичное (обозначается буквой "К"), металлическое ("М"), смешанное ("СМ"). По заселенности здания могут быть жилыми ("Ж") и нежилыми ("Н"). Информация внутри здания является комбинацией этих типов. Например, двухэтажное кирпичное жилое здание обозначается "2КЖ". Отсутствие подобной информации внутри контура обычно означает, что этот объект не является зданием. Поиск внутренней информации также осуществляется с помощью волнового алгоритма.

Распознавание служебной информации происходит путем сравнения ее отдельных символов с эталонными знаками, которые встречаются на топографических планах. Чтобы распознать служебную информацию, ее нужно сперва привести к горизонтальному положению, а для этого следует найти угол наклона надписи, который всегда имеет

такое же направление, как и одна из стен здания. Либо, если надпись состоит из нескольких букв, то угол наклона можно найти, проведя линию, проходящую через центры прямоугольников, описанных вокруг каждой буквы. Далее берется каждый символ из повернутой надписи и сравнивается с каждым эталонным знаком (рис. 2).

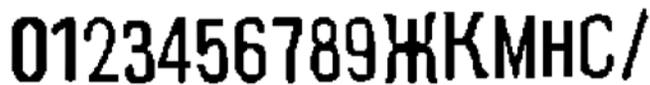


Рис. 2. Эталонные символы

Дополнительной проверкой корректности распознанных данных является синтаксический разбор найденной последовательности символов. Он основан на том, что перед надписью может быть число, означающее число этажей. Если этого числа нет, то у здания один этаж. Синтаксический контроль позволяет исправить ошибочно распознанные надписи. Так, буква "С" часто распознается как "0", а "Н" как "К", и наоборот. Например, текст вида "0МН" или "НЖ" будет преобразован соответственно в "СМН" и "КЖ".

После поиска зданий осуществляется поиск остальных объектов, таких как деревья, кусты, светофоры и уличные фонари (рис. 3).

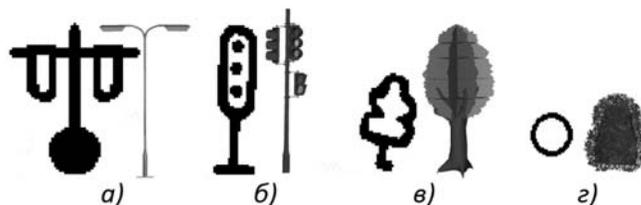


Рис. 3. Символы на топографическом плане и соответствующие им 3D-модели:

а — фонарь; б — светофор; в — дерево; г — клумба

Данные объекты находятся по своим алгоритмам, каждый из которых основан на определении геометрических особенностей контуров этих объектов и сравнении участка топографического плана с эталонным изображением такого объекта. Подробное описание приведено в работе [4].

Моделирование рельефа

Добавление к сцене рельефа придаст ей значительную долю реалистичности. Для этого сначала необходимо установить географическое положение объектов. Обычно топографический план имеет геолокацию, но если таких данных нет, то можно воспользоваться сервисом Google Maps. Для этого достаточно установить географические координаты двух точек на топографическом плане. На основании этих данных возможно определить геогра-

фические координаты любой точки на топографическом плане с помощью сервиса Google Maps API. Пусть A, B — точки Земли с известными географическими координатами для соответствующих им точек a и b на топографическом плане (далее, опорные точки) (рис. 4, см. третью сторону обложки).

Тогда географические координаты любой точки C с известными координатами c_x и c_y на топографическом плане будут определяться следующим образом:

$$C_{longitude} = \frac{c_x - a_x}{b_x - a_x} (B_{longitude} - A_{longitude}) + A_{longitude}; \quad (1)$$

$$C_{latitude} = \frac{c_y - a_y}{b_y - a_y} (B_{latitude} - A_{latitude}) + A_{latitude}. \quad (2)$$

Для того чтобы географические координаты были определены с высокой точностью, требуется соблюдение следующих условий для выбора опорных точек:

- точки не должны находиться на одной горизонтальной или вертикальной прямой и тем более совпадать;
- расстояние между точками должно быть большим, насколько это возможно.

Обозначим ширину и высоту топографического плана соответственно W и H . Разобьем его область на отдельные равные прямоугольные участки. Пусть таких участков будет mn , где m — их число по горизонтали, а n — по вертикали. Для каждой точки топографического плана $p_{i,j}$, где $i = 0, \dots, m$ и $j = 0, \dots, n$, определим географические координаты точек $P_{i,j}$ с помощью (1) и (2). Число точек на каждой вертикали и горизонтали будет на одну больше, чем число участков.

Сервис Google Maps Elevation API позволяет получать информацию о высоте над уровнем моря для массива точек в одном запросе. Число точек в массиве ограничено только длиной запроса к сервису, который не должен превышать 2000 знаков. Чтобы увеличить число возможных точек в массиве, в сервисе предусмотрена возможность "сжимания" записи точек в более компактный вид [10]. Запрос по получению высот для массива точек будет выглядеть следующим образом:

<https://maps.googleapis.com/maps/api/elevation/xml?locations=40.700000,-120.950000|43.252000,-126.453000>

После кодирования запрос будет следующим:

https://maps.googleapis.com/maps/api/elevation/xml?locations=enc:_flwFn`faV_mqNvxq`@

В результате обработки всех точек строится трехмерная триангуляционная поверхность.

На основании опорных точек возможно получить текстуру поверхности рельефа. Для этого используется сервис Google Maps API. Сервис позволяет получить изображение местности со спутника

по географическим координатам точки. За один запрос максимально возможное по размеру изображение можно получить с размерами 640×640 пикселей. На топографическом плане строится сетка из точек. Для каждой точки будет получаться изображение. Число точек по горизонтали и вертикали определяются выражениями

$$K_W = \left[\frac{W}{640} \right]; K_H = \left[\frac{H}{640} \right], \quad (3)$$

где скобки означают получение следующего целого числа после деления.

Для определения ширины и высоты каждого изображения используется формула

$$w = \left[\frac{W}{K_w} \right], h = \left[\frac{H}{K_h} \right], \quad (4)$$

где скобки обозначают округление до целого.

Для каждой опорной точки $t_{i,j}$, где $i = 1, \dots, K_w$ и $j = 1, \dots, K_h$, находятся географические координаты с помощью формул (1) и (2). Для каждой найденной таким образом географической точки $T_{i,j}$ выполняется запрос для получения изображения.

Таким образом, на основании двух точек на топографическом плане построен текстурированный рельеф (рис. 5, см. третью сторону обложки). После формирования рельефа, построенные трехмерные модели зданий и дополнительных объектов должны быть соответственно смещены по оси Z .

Текстурирование зданий

Нанесение текстур на здание — наиболее трудоемкий и затратный по времени этап реконструкции сцены. Для нанесения текстуры на одну стену здания требуется сфотографировать эту стену и провести коррекцию перспективы. Число зданий и соответственно стен в масштабной сцене может достигать нескольких десятков. Ручное текстурирование каждого здания занимает очень большое время, поэтому предлагается использовать базовый набор стандартных текстур (рис. 6, см. третью сторону обложки), каждая из которых характеризует тот или иной тип здания.

Текстура здания должна состояться из композиции текстуры из базового набора (рис. 7). По

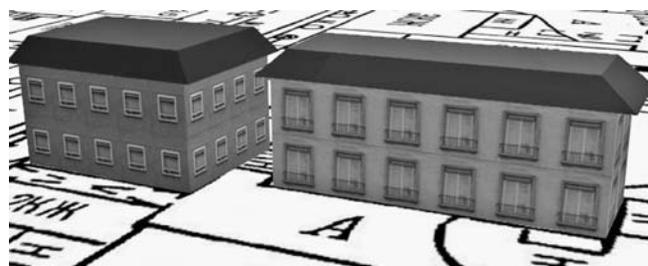


Рис. 7. Здания со стандартными текстурами

вертикали базовая текстура должна повторяться столько раз, сколько у здания этажей. По горизонтали базовая текстура должна повторяться следующее число раз:

$$k = \left[\frac{\text{Высота текстуры} \cdot \text{Длина стены}}{\text{Ширина текстуры} \cdot \text{Высота этажа}} \right]. \quad (5)$$

Текстура для здания выбирается исходя из данных о заселенности и материала здания. Информация об этом присутствует после выделения зданий на топографическом плане.

Дополнительно предлагается проводить автоматическое текстурирование зданий фотореалистичными изображениями при наличии следующих данных:

- географического положения и направления камеры;
- координат углов стены на фотоснимке для коррекции перспективы;
- географических координат всех объектов сцены.

Современные телефоны и планшеты позволяют получать фотоснимки с записью географических координат, определяемых с помощью систем GPS или ГЛОНАСС.

Нахождение углов стены на фотоснимке — довольно сложная и нетривиальная задача, поэтому выделение стены на фотоснимке осуществляют вручную. Для этого разработано специальное программное обеспечение для мобильных устройств (рис. 8, см. третью сторону обложки). На основании выделенных четырех точек стены корректируется перспектива на фотоснимке.

Для каждого фотоснимка на модели необходимо определить здание и стену, на которую должна быть нанесена текстура. Для этого из точки съемки проводится луч по направлению угла съемки. Ближайшая плоскость стены, которую пересечет этот луч, и будет искомым (рис. 9).

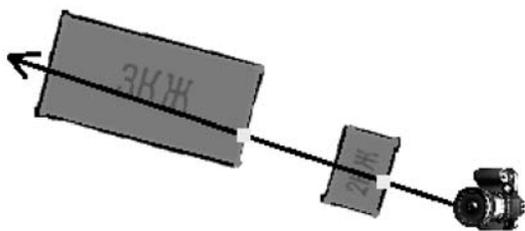


Рис. 9. Определение ближайшей стены

Пример реконструкции

Для демонстрации работы алгоритма реконструкции городской сцены был выбран реальный топографический план одной из улиц города Владивостока (рис. 10, см. четвертую сторону обложки). Для данной сцены было сделано 30 фотоснимков стен зданий, их местоположение и направление также показано на рис. 10. По известным геогра-

фическим координатам были получены данные о высотах над уровнем моря и построен рельеф местности. Также были получены спутниковые снимки для подложки. Реальная сцена содержит 115 зданий. На топографическом плане правильно распознано 108 зданий. Процент верно распознанной внутренней информации внутри зданий составляет 94 %. Число верных автоматически найденных объектов сцены составляет 84 %. Время, затраченное на реконструкцию, составило около 6 мин.

Общий вид трехмерной сцены после реконструкции и нанесения текстур показан на рис. 11 (см. четвертую сторону обложки).

Заключение

На основании совокупности начальных данных, таких как топографический план, географические координаты нескольких точек на нем и фотоснимки зданий, были получены результаты, которые наглядно показывают надежность и точность предлагаемого метода реконструкции.

Автоматизация практически всего процесса реконструкции позволяет проводить моделирование масштабных городских сцен за короткое время. Основное время занимает работа оператора при подготовке и обработке фотоснимков городского пространства.

Список литературы

1. **Vezhnevets V., Konushin A., Ignatenko A.** Interactive image-based urban modeling // PIA-2007, 2007.
2. **Knopp J., Prasad M., Gool L. V.** Scene cut: Class-specific object detection and segmentation in 3D scenes // Proceedings of international conference on 3D imaging, modeling, processing, visualization and transmission. 2011. P. 180—187.
3. **Кудряшов А. П., Соловьев И. В.** Распознавание контуров зданий на топографическом плане для реконструкций городских сцен // Вестник компьютерных и информационных технологий. 2015. № 2. С. 3—8.
4. **Кудряшов А. П., Соловьев И. В.** Реконструкция трехмерной модели городского пространства на основе топографического плана // Вестник Амурского государственного университета. 2016. № 73. С. 58—66.
5. **Чернов А. В., Чупшев Н. В.** Автоматическое распознавание контуров зданий на картографических изображениях // Компьютерная оптика. 2007. № 31. С. 101—103.
6. **Гуненко И. В., Салтыкова Ю. С., Середович В. А., Комиссаров Д. В.** Технология построения трехмерных моделей городов на основе топографических планов // Интерэкспо Гео-Сибирь. 2006. Т. 1. № 2.
7. **Lin B. C., You R. J.** Tensor-based quality prediction for building model reconstruction from LIDAR data and topographic map // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXIX-B7. 2012. P. 437—441, doi: 10.5194/isprsarchives-XXXIX-B7-437-2012.
8. **Wang S.** Floating model for building reconstruction from topographic maps and LIDAR data // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2008, Vol. XXXVII. Part B3b. P. 681—685.
9. **Бобков В. А., Соловьев И. В.** Алгоритм распознавания контуров на топлане // Одиннадцатый Международный Форум студентов, аспирантов и молодых ученых стран Азиатско-Тихоокеанского региона: материалы конференции. Май 2012 г. Владивосток. 2012. С. 044.
10. <https://developers.google.com/maps/documentation/utilities/polylinealgorithm?hl=ru> (дата обращения 11.12.2016).

A. P. Kudrjashov, Junior Researcher, e-mail: kudryashovA@mail.ru,
Institute of Applied Mathematics, Far Eastern Branch of RAS, Vladivostok,
I. V. Solov'iov, Graduate Student, e-mail: igorek.solovyev@mail.ru,
Institute of Applied Mathematics, Far Eastern Branch of RAS, Vladivostok

Reconstruction of the City Environment Based on Terrain Using the Topographic Map and Services Google Maps

In this paper we propose a method for reconstruction of urban space on the basis of topographic maps and geographic coordinates. Our algorithm allocates map information on buildings and creates a 3D-terrain model. Additional data allow to construct relief. Texturing buildings carried an automatic method based on real photos. The method of reconstruction is based on use of wave algorithm of Lee. Modeling of a scene is based on extracting of service information on a topographic map. The whole process is fully automatic using data of a topographic map algorithm automatically chooses texture of the building and puts it on model. We demonstrate the algorithm on real maps of Vladivostok. The topographic map is used as initial data. Result of work of algorithm — the textured 3D-scene consisting of models of buildings, relief and additional objects. Our method allows to recreate the simplified models of scenes. However, these models may be used in applied tasks.

Keywords: three-dimensional reconstruction, topographic map, geographic coordinates, texturing, urban setting

References

1. **Vezhnevets V., Konushin A., Ignatenko A.** Interactive image-based urban modeling, *PIA-2007*. 2007.
2. **Knopp J., Prasad M., Gool L. V.** Scene cut: Class-specific object detection and segmentation in 3D scenes, *Proceedings of international conference on 3D imaging, modeling, processing, visualization and transmission*, 2011, pp. 180–187
3. **Kudrjashov A. P., Solov'jov I. V.** Raspoznavanie konturov zdaniy na topograficheskom plane dlja rekonstrukcij gorodskih scen (Recognition contours of buildings on the topographic plan for the reconstruction of urban scenes), *Vestnik komp'yuternyh i informacionnyh tehnologij*, 2015, no. 2, pp. 3–8 (in Russian).
4. **Kudrjashov A. P., Solov'jov I. V.** Rekonstrukcija trehmernoj modeli gorodskogo prostranstva na osnove topograficheskogo plana (Reconstruction of three-dimensional model of urban space based on topographic plan), *Vestnik Amurskogo gosudarstvennogo universiteta*, 2016, no. 73, pp. 58–66 (in Russian).
5. **Chernov A. V., Chupshev N. V.** Avtomaticheskoe raspoznavanie konturov zdaniy na kartograficheskikh izobrazhenijah (Automatic recognition of contours of buildings on the map images), *Komp'yuternaja optika*, 2007, no. 31, pp. 101–103 (in Russian).
6. **Gunenko I. V., Saltykova Ju. S., Seredovich V. A., Komissarov D. V.** Tehnologija postroenija trehmernyh modelej gorodov na osnove topograficheskikh planov (The technology of constructing three-dimensional models of cities on the basis of topographical plans), *Interjekspo Geo-Sibir'*, 2006, vol. 1, no. 2 (in Russian).
7. **Lin B. C., You R. J.** Tensor-based quality prediction for building model reconstruction from LIDAR data and topographic map // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXIX-B7, 2012, pp. 437–441, doi: 10.5194/isprsarchives-XXXIX-B7-437-2012.
8. **Wang S.** Floating model for building reconstruction from topographic maps and LIDAR data, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2008, vol. XXXVII, part B3b, pp. 681–685.
9. **Bobkov V. A., Solov'jov I. V.** Algoritm raspoznavanija konturov na topoplane (Recognition Algorithm contours on topographic maps), *Odinnadcatyj Mezhdunarodnyj Forum studentov, aspirantov i molodyh uchjonyh stran Aziatsko-Tihookeanskogo regiona: materialy konferencii*, Vladivostok: May 2012. P. 044 (in Russian).
10. **Encoded Polyline Algorithm Format.** <https://developers.google.com/maps/documentation/utilities/polylinealgorithm?hl=ru> (date of access 11.12.2016)

УДК 004.91

И. С. Печенко, науч. сотр., e-mail: ivan.pechenko@intel.com,
О. В. Венгер, ст. науч. сотр., e-mail: oleg.v.venger@intel.com,
Д. А. Плоткин, науч. сотр., e-mail: dmitry.plotkin@intel.com,
ЗАО "Интел А/О"

ACES Tables: среда для создания и использования табличных спецификаций систем на кристалле

Ввиду сложности современных систем на кристалле крайне важными этапами их проектирования становятся анализ и верификация. Зачастую верификация систем начинается сравнительно поздно, после создания RTL-кода. Однако в последнее время ведется активная разработка методов создания архитектурных спецификаций систем на кристалле, которые могут быть использованы для анализа и верификации этих систем, что позволит начинать верификацию намного раньше. Описывается один из таких методов, предоставляющий возможность создания структурных спецификаций систем на кристалле в табличном виде.

Ключевые слова: система на кристалле, процесс проектирования, верификация, архитектурная спецификация, таблица, объектно-ориентированный подход, шаблон, импорт данных, экспорт данных

Введение

Сложность проектирования современных систем на кристалле чрезвычайно высока: они состоят из десятков функциональных компонентов, как аппаратных, так и программных. Ввиду чрезвычайной сложности систем на кристалле в процессе их проектирования на первый план выходит верификация различных компонентов системы и связей между ними. Причем такая верификация должна начинаться как можно раньше в цикле разработки системы, так как ошибки, выявленные на поздних стадиях разработки, "стоят" компании-разработчику дороже всего [1–4]. Однако традиционный процесс проектирования предполагает, что верификация компонентов системы на кристалле начинается после готовности RTL-кода этих компонентов. Основная причина такого положения состоит в том, что имеющиеся на ранних стадиях проектирования архитектурные и микроархитектурные спецификации в большинстве случаев состоят из документов, написанных на естественных языках и, соответственно, имеют все недостатки естественных языков, такие как неточность, двусмысленность, неполнота. В таком виде архитектурные спецификации систем совершенно не приспособлены к машинной обработке и формальной верификации. Вместе с тем, описание в формате регистровых передач (Register Transfer Level, RTL) имеет два ключевых недостатка: оно появляется довольно поздно в процессе разработки и часто является слишком

сложным для полной верификации систем, таких как системы на кристалле. Для верификации систем на кристалле требуется более абстрактная модель, однако создание таких моделей требует серьезных дополнительных усилий разработчиков системы.

В идеальном случае для начала верификации системы на более ранних этапах ее проектирования необходимо использовать архитектурные спецификации для автоматической генерации исполняемых моделей системы.

Табличные спецификации систем на кристалле

Система на кристалле может быть представлена как сеть, состоящая из связанных между собой компонентов (IP-блоков). Архитектурная спецификация такой системы может быть условно разделена на две части: структурную и поведенческую. Структурная спецификация системы на кристалле представляет собой описание строения системы, перечень входящих в нее блоков и описание устройства каждого из них. Поведенческая спецификация — это описание поведения системы в различных ситуациях, а также протоколы взаимодействия различных ее компонентов.

На первых стадиях проектирования систем на кристалле их архитектурная спецификация обычно представляет собой неформализованное текстовое описание системы. При этом наиболее подходящей и распространенной формой представления ранних поведенческих спецификаций являются диаграм-

мы, а наиболее распространенная форма представления структурных спецификаций — таблицы. Позже, с увеличением детализации описания системы, в архитектурной спецификации системы начинают использоваться формальные и частично формализованные языки представления данных. В фокусе этой работы находятся структурные аспекты спецификации системы на кристалле и их табличное представление.

В качестве примеров элементов структурной спецификации систем на кристалле можно назвать интерфейсы IP-блоков, сообщения, которыми они обмениваются, наборы контрольных регистров этих IP-блоков, систему синхронизации и т.д.

Интерфейсы IP-блоков описывают способы взаимодействия компонентов системы, обмена информацией между ними. Спецификация интерфейсов является отправной точкой для проектирования функциональных блоков и соединений между ними и позволяет проводить высокоуровневое моделирование взаимодействия IP-блоков, не вдаваясь в детали их устройства. В узком смысле каждый интерфейс описывает, какие сигналы может отправлять и принимать данный компонент системы, а также различные характеристики таких сигналов. Связь интерфейсов и IP-блоков неоднозначна: как один блок может иметь несколько интерфейсов, так и один интерфейс может охватывать несколько блоков. Описание интерфейса как списка сигналов интуитивным образом представляется в виде таблицы, каждая запись в которой представляет собой один сигнал.

Сообщения, которыми обмениваются IP-блоки, представляют собой двоичные сигналы определенной длины, в которых в определенном порядке зашифрована необходимая для взаимодействия этих блоков информация. Табличное описание этих сообщений представляет собой побитовую или побайтовую конфигурацию сигнала и обозначенное определенным образом назначение каждого блока сообщения. Рис. 1 демонстрирует пример таблицы сообщений. В качестве материала для примеров была использована спецификация OpenSPARC T2 [5].

Как и описания интерфейсов, естественным образом представляются в виде таблицы описания контрольных регистров. Контрольными называются регистры, которые управляют поведением процессора или другого блока вычислительной системы.

Message definition					
Title	Chassis Software messages				
Group	N+2	N+1	N	8	7
Request message	-	-	-	checksum	payload
Response message	-	-	checksum	payload	completion code
Encapsulated request message	-	stop	checksum	payload	cmd
Encapsulated response message	stop	checksum	payload	completion code	cmd

Рис. 1. Пример таблицы сообщений (архитектура OpenSPARC T2)

Register Definitions			
Comments	Summary of OpenSPARC T2 Ancillary State Registers		
Header			
Macros			
Register Type	Register Address	Description	Access Type
Y	00	Y Register	RW
Reserved	01	Any access causes a illegal_instruction trap	
CCR	02	Condition Code register	RW
ASI	03	ASI register	RW
TICK	04	TICK register	RO
PC	05	Program counter	RW
FPRS	06	Floating-Point Registers Status register	RW
Reserved	07-0E	Any access causes an illegal_instruction trap	
(MEMBAR, STBAR, SIR)	0F	Instruction opcodes only, not an actual ASR.	
PCR	09	Performance counter control register	RW
PIC	10	Performance instrumentation counter	RW
Reserved	11	Any access causes an illegal_instruction trap	
GSR	12	General Status register	RW
SOFTINT_SET	13	Set bit in Soft Interrupt register	WO
SOFTINT_CLR	14	Clear bit in Soft Interrupt register	WO
SOFTINT	15	Soft Interrupt register	RW
TICK_CMPR	16	TICK Compare register	RW
STICK	17	System Tick register	RW
STICK_CMPR	18	System TICK Compare register	RW

Рис. 2. Пример таблицы контрольных регистров (архитектура OpenSPARC T2)

Примерами задач, решаемых контрольными регистрами, являются контроль прерываний, управление адресацией памяти, управление подкачкой страниц и многое другое. При создании спецификации регистров для них указывается множество параметров, таких как разрядность регистров, поля в их составе, поведение регистров при сбросе, характеристики доступа к регистрам, возможности доступа к ним при валидации системы и т. д. Пример таблицы контрольных регистров демонстрирует рис. 2.

Еще одним видом спецификации, представляемой в табличной форме, является описание системы синхронизации. В системе на кристалле обычно используется множество тактовых сигналов с различными частотами. Чаще всего эти сигналы создаются программируемыми микросхемами генерации, что позволяет регулировать их частоту в процессе работы системы. Архитектурное описание тактовых сигналов содержит возможные частотные характеристики сигнала, а также различные способы управления сигналом и возможности доступа к нему при валидации системы.

Средства работы с таблицами

Для устранения разрыва между созданием архитектурных спецификаций и началом верификации системы необходима возможность использования этих спецификаций для автоматического создания исполняемых моделей. Это накладывает дополнительные ограничения на архитектурные спецификации.

Во-первых, для их создания требуется формальное или частично формализованное представление. Для структурных спецификаций системы таблицы являются одним из часто используемых вариантов такого представления.

Во-вторых, архитектурные спецификации должны удовлетворять требованиям точности, полноты

и непротиворечивости. Если говорить о табличной форме спецификаций, то важной задачей при решении задачи подготовки их к трансляции в формальные модели является обеспечение возможности создания табличных спецификаций, свободных от ошибок, типичных для такого формата данных.

В литературе описаны два основных подхода к проблеме получения корректных табличных данных. Один из них [6—9] предполагает полуавтоматический анализ и валидацию таблиц. Дело в том, что формальный анализ табличных данных при отсутствии информации о структуре таблицы невозможен в большинстве случаев, поэтому все методы валидации табличных данных, не полагающиеся на знания структуры таблиц и предметной области, не полностью автоматизированы и требуют непосредственного участия пользователя.

Второй подход к получению корректных табличных данных [10—15] называется объектно-ориентированным. Этот подход предполагает соответствие синтаксиса и семантики таблиц заранее определенной модели. Процесс получения таких таблиц состоит из двух этапов: сначала создаются их шаблоны, а затем отдельные экземпляры, соответствующие этим шаблонам. С помощью построения шаблонов с определенными свойствами такие таблицы можно реализовать свободными от многих типовых для табличных данных ошибок, например, некорректных ссылок, типов и пробелов в записях.

В литературе встречается несколько примеров подобного подхода к созданию таблиц. Одним из таких примеров является работа [10], представляющая ViTSL (Visual Template Specification Language) — язык описания шаблонов таблиц. Для генерации таблиц авторы работы используют среду Gencel [16], расширение для программы Microsoft Excel, реализующее объектно-ориентированный подход к созданию таблиц.

Вторым известным описанным в литературе примером применения данного подхода является система создания таблиц ClassSheets [13]. Ряд работ [11, 12, 14, 15] описывает различные расширения для системы ClassSheets. В них особенно подробно описывается важнейшая функция, реализованная в системе ClassSheets, — поддержка изменений шаблонов после создания таблиц на их основе. В этом случае для восстановления соответствия таблиц измененным шаблонам необходимо применять к таблицам операторы, описывающие трансформацию их шаблонов.

Предлагаемое решение: ACES Tables

Для решения задачи разработки архитектурных спецификаций, подходящих для использования в целях верификации системы, в компании Intel была разработана интегрированная платформа ACES (Architectural, Consistent, Efficient Specifications), которая объединяет в себе функции создания, ве-

рификации и дальнейшего использования архитектурных спецификаций вычислительных систем. Наличие такой платформы открывает широкие возможности для совместного использования спецификаций различных аспектов системы в целях создания более полных моделей системы, ее анализа и валидации. Так как зачастую основной платформой для создания архитектурных спецификаций является Microsoft Office, платформа ACES реализована как набор надстроек над различными программами пакета Microsoft Office, такими как Microsoft Excel и Microsoft Visio, а также связывающая эти надстройки инфраструктура. Платформа позволяет пользователю при работе в привычной среде создавать более качественные спецификации и предоставляет средства для машинной обработки и экстракции данных из документов Microsoft Office.

Данная работа представляет часть платформы ACES, предназначенную для работы с табличными спецификациями систем на кристалле. Для обеспечения корректности табличных спецификаций используется объектно-ориентированный подход к созданию таблиц. При соблюдении условий соответствия таблиц определенному шаблону табличный формат становится свободен от недостатков естественных языков и при этом, в отличие от формальных языков, не требует экспертных знаний для работы с ним, поэтому прекрасно подходит для создания структурной спецификации системы.

Предложенная в данной работе система ACES Tables реализована в виде расширения для Microsoft Excel. Таблица в системе ACES Tables представляет собой расширение стандартной таблицы Microsoft Excel. Помимо основного блока — массива записей с определенным набором полей — она включает в себя заголовок — область для записи данных, общих для всей таблицы, таких как название таблицы, версия и т. д.

Шаблон таблицы в системе ACES Tables состоит из двух частей. Первая часть шаблона представляет собой шаблон внешнего вида таблицы, в котором определяется набор столбцов, а также графические характеристики таблицы, такие как цвет, ширина столбцов и т. д. Вторая часть шаблона таблицы — это описание дополнительных характеристик таблицы в виде текста в формате XML. Набор дополнительных характеристик может включать в себя: тип данных, формат, начальные значения и требования уникальности значений для столбцов таблицы; список выражений и предикатов для автоматической проверки целостности и непротиворечивости данных в таблице; ссылки на возможные источники для подкачки данных в таблицу; описания перекрестных ссылок между таблицами.

Как правило, для спецификации определенного аспекта архитектуры необходим модуль, включающий в себя набор из нескольких шаблонов и специфические для каждого набора средства автома-

тизации их использования. ACES Tables поддерживает работу с произвольным числом таких модулей, включая модули, созданные пользователем. Такая организация программного комплекса обеспечивает его расширяемость, предоставляет возможности для более гибкого использования его для спецификации различных аспектов архитектуры систем на кристалле.

Возможности ACES Tables

Система управления табличными спецификациями ACES Table Spec предоставляет возможности по частичной автоматизации процесса создания, валидации и использования структурных спецификаций систем на кристалле, основанные на:

- различных возможностях импорта и экспорта данных как в форматах, поддерживаемых ACES, так и в других форматах. Последнее осуществляется с помощью подключаемых к платформе ACES модулей;
- механизме валидации табличных данных;
- специальных функциях, разработанных для улучшения визуальной привлекательности, удобства использования и предупреждения нарушения целостности таблиц, таких как добавление и удаление записей, регулирование доступа к полям таблицы и т. д.;
- механизме автоматического обновления таблиц при изменениях их шаблонов.

Создание табличных спецификаций в системе ACES Tables осуществляется в два этапа: сначала по существующему шаблону создается экземпляр пустой таблицы, затем таблица заполняется необходимыми данными. Так как во многих случаях спецификации создаются с использованием документации из разработанных ранее проектов, перед системой разработки спецификаций встает задача импорта данных в различных форматах описания архитектуры систем на кристалле в таблицы ACES. Источниками таких данных могут быть различные базы данных регистров, а также готовые спецификации IP-блоков и их связей. Примером источника данных для таблиц ACES, работа с которым поддерживается в системе, являются спецификации регистров на языке SystemRDL [17]. В качестве промежуточного формата представления данных в системе ACES Tables используется формат ACES XML. Задача трансляции из различных форматов описания архитектуры систем на кристалле в ACES XML решается с помощью подключения дополнительных модулей трансляции к платформе ACES. Данные же в формате ACES XML могут быть преобразованы в табличный формат с использованием стандартного конвертера ACES Tables.

Еще одним важнейшим аспектом работы с табличными спецификациями, который реализован в системе ACES Tables, является валидация данных. Хотя объектно-ориентированный подход направ-

лен на разработку корректных по построению таблиц, некоторые данные в них все равно необходимо подвергать процессу валидации. Для этого система ACES Tables предлагает решение, основанное на описанных в формате xml дополнительных характеристиках шаблонов таблиц. В числе прочего эти дополнительные характеристики включают список выражений и предикатов для автоматической проверки целостности и непротиворечивости данных в таблице. Подобный вариант валидации табличных данных, использующий проверку заданных пользователем утверждений, описан в работе [18]. В отличие от описанного в [18] механизма, валидация данных в системе ACES Tables предоставляет возможность валидировать не только конкретные записи таблицы, но и добавлять требования к существованию или единственности определенных записей.

Ниже приведен пример синтаксиса утверждения для валидации данных в системе ACES Tables:

```
<assert>
  <expr>v1! = v2</expr>
  <var>
    <name>v1</name>
    <table_type>RegisterTable</table_type>
    <column>Name</column>
    <row>[1:-1]</row>
  </var>
  <var>
    <name>v2</name>
    <column>Name</column>
    <row>[1:-1]</row>
  </var>
</assert>
```

В этом выражении проверяется, что в текущей таблице нет ни одной строки, в которой поле Name совпадает с полем Name хотя бы в одной строке таблицы типа RegisterTable.

Кроме условий соответствия всех найденных по определенному признаку строк определенному выражению, в системе ACES Tables реализована возможность проверки предикатов, определяющих существование и уникальность определенных записей в таблице (предикаты Exists и Unique). Таким образом, внутренний механизм валидации таблиц в системе ACES Tables имеет достаточно широкую область применения.

Важной частью системы автоматизации работы с табличными спецификациями являются функции улучшения визуальной привлекательности и удобства использования таблиц. ACES Tables предлагает множество инструментов для удобства работы с таблицами. К примеру, для каждого шаблона таблиц имеется возможность настроить список скрытых полей и список полей с доступом только для чтения. Имеется также возможность настроить цветное выделение синтаксиса данных в ячейках таблицы.

Одной из важнейших проблем в процессе использования табличных спецификаций, чей синтаксис определен с помощью заранее подготовленных шаблонов, является поддержка обновления этих таблиц при изменении их шаблонов. В рамках системы ACES Tables реализована возможность такой поддержки. При любом изменении шаблона таблицы, будь то добавление или удаление столбцов или изменение текста дополнительных характеристик шаблона, меняется его версия. После этого для всех таблиц, созданных на основе этого шаблона, появляется возможность применить операцию Upgrade, в результате которой таблица будет подвергнута таким же изменениям.

Последний этап использования спецификаций систем на кристалле, созданных с помощью ACES Tables, состоит в автоматическом экспорте данных. Начальным форматом данных является модель данных Microsoft Excel, конечный формат данных зависит от приложения. Чтобы облегчить процесс трансляции, ACES Tables предоставляет конвертер из модели данных Microsoft Excel в промежуточное XML-представление. Задача дальнейшей трансляции решается с помощью подключения к платформе ACES дополнительных модулей.

Заключение

Основной проблемой существующего процесса создания и использования архитектурных спецификаций систем на кристалле является разрыв между созданием архитектурных спецификаций и началом верификации системы. Этот разрыв возникает из-за недостаточного уровня формализации архитектурных спецификаций и отсутствия возможности их автоматической трансляции в исполняемые модели. В данной работе была представлена система для создания и использования архитектурных спецификаций систем на кристалле в табличном виде ACES Tables, созданная для сокращения этого разрыва. Представленная система реализована на базе программы для работы с таблицами Microsoft Excel. Опираясь на объектно-ориентированный подход к созданию таблиц, данная система обеспечивает поддержку всего процесса создания, валидации и использования структурных спецификаций систем на кристалле.

Важной особенностью описанной системы является ее расширяемость. Для каждого аспекта структурной спецификации может быть определен набор шаблонов таблиц и сопутствующая автоматизация его использования, а для различных форматов представления данных могут быть подключены модули, реализующие трансляцию данных из этих форматов в таблицы ACES Tables. Таким образом, представленная система открывает широкие возможности автоматической трансляции архитектурных спецификаций из формальных языков в табличный вид и обратно, что позволяет со-

кратить разрыв между созданием архитектурных спецификаций и верификацией системы.

Основным направлением дальнейшей работы над описанной системой является расширение библиотеки шаблонов таблиц и модулей трансляции в различные форматы представления данных, которое приведет к расширению области применения данной системы.

Список литературы

1. Lee Y., Kim N., Kim D., Lee D., In H. Customer Requirements Elicitation based on Social Network Service // *KSTT Transactions On Internet And Information Systems*. October 2011. Vol. 5, N. 10. P. 1733—1750.
2. Keller T. Contextual Requirements Elicitation // *Seminar in Requirements Engineering*. Spring 2011. Department of Informatics, University of Zurich.
3. Dhungana D., Seyff N., Graf F. Research Preview: Supporting End-user Requirements Elicitation Using Product Line Variability Models // *Requirements Engineering Foundation for Software Quality*. March 2011. Vol. 6606. P. 66—71.
4. Neetu K. S., Pillai A. S. A survey on global requirements elicitation issues and proposed research framework // *Software Engineering and Service Science (ICSESS)*. 2013, 4th IEEE International Conference. 2013. P. 554—557.
5. Спецификация архитектуры OpenSparc-T2. URL: <http://www.oracle.com/technetwork/systems/opensparc/opensparc-t2-page-1446157.html> (дата обращения: 01.07.2016).
6. Abraham R., Erwig M. Goal-directed debugging of spreadsheets // *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium*. 2005. P. 37—44.
7. Abraham R., Erwig M. GoalDebug: A spreadsheet debugger for end users // *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society. 2007. P. 251—260.
8. Fisher M., Rothermel G., Creelan T., Burnett M. Scaling a Dataflow Testing Methodology to the MultiparadigmWorld of Commercial Spreadsheets // *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium*. 2006. P. 13—22.
9. Abraham R., Erwig M. Mutation operators for spreadsheets // *Software Engineering, IEEE Transactions*. 2009. Vol. 35, N. 1. P. 94—108.
10. Abraham R., Erwig M., Kollmansberger S., Seifert E. Visual specifications of correct spreadsheets // *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium*. 2005. P. 189—196.
11. Mendes J. Coupled evolution of model-driven spreadsheets // *Software Engineering (ICSE), 2012 34th International Conference*. 2012. P. 1616—1618.
12. Cunha J., Fernandes J. P., Mendes J., Saraiva J. MDSheet: A framework for model-driven spreadsheet engineering // *Proceedings of the 34th International Conference on Software Engineering*. 2012. P. 1395—1398.
13. Engels G., Erwig M. ClassSheets: automatic generation of spreadsheet applications from object-oriented specifications // *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*. 2005. P. 124—133.
14. Cunha J., Fernandes J. P., Mendes J., Pacheco H., Saraiva J. Bidirectional Transformation of Model-Driven Spreadsheets // *International Conference on Theory and Practice of Model Transformations*. 2012. Vol. 12. P. 105—120.
15. Stevens P. Bidirectional model transformations in QVT: Semantic issues and open questions // *Model Driven Engineering Languages and Systems*. 2007. P. 1—15.
16. Erwig M., Abraham R., Kollmansberger S. and Cooperstein I. Gencel: a program generator for correct spreadsheets // *Journal of Functional Programming*. 2006. Vol. 16, N. 03. P. 293—325.
17. Описание языка SystemRDL. URL: <http://accellera.org/activities/working-groups/systemrdl> (дата обращения: 01.07.2016).
18. Burnett M., Cook C., Pendse O., Rothermel G., Summet J., Wallace C. End-user software engineering with assertions in the spreadsheet paradigm. In *Proceedings of the 25th international conference on Software engineering*. May 2003. P. 93—103.

I. S. Pechenko, Research Associate, e-mail: ivan.pechenko@intel.com,
O. V. Venger, Senior Researcher, e-mail: oleg.v.venger@intel.com,
D. A. Plotkin, Research Associate, e-mail: dmitry.plotkin@intel.com,
ZAO Intel A/O

ACES Tables: a Framework for Creation and Usage of SoC Architectural Specifications in Table Form

Growing complexity of SoC design process leads to growing the importance of design verification process. However, traditional design process assumes that verification begins only after RTL coding phase. There are several reasons for it, and the main reason is impossibility of automatic processing and formal verification of most architectural specifications.

SoC architectural specification on the early design phases is an informal textual description of the system, but includes some structured elements such as spreadsheets and diagrams. An essential part of the spec then is presented in tabular form.

Architectural specifications can be used for formal system verification only if: they can be automatically processed, their quality, completeness and unambiguity are under tight control. Another important requirement is convenience of capturing these architectural specifications.

To implement these requirements we developed ACES (Architectural, Consistent, Efficient Specifications) system that supports the creation and use of structured components of SoC specifications (spreadsheets and diagrams). Here in this work we describe ACES capabilities for working with spreadsheets implemented in ACES Tables subsystem. ACES Tables is an extension for Microsoft Excel that implements object-oriented approach to table creation when each table is created based on its template. Table templates determine table layout, rules for checking data in the table, data relations and data swapping rules. For each architecture aspect ACES needs a module that consists of specific table templates and some automation.

ACES Tables subsystem provides capabilities for automatization of the process of creation and usage of SoC architectural specifications. These capabilities include:

- Basic data import and export capabilities using ACES XML intermediate format.
- Data validation engine.
- Some functions for convenient usage of ACES tables: records addition and deletion mechanism, columns hiding and access control mechanisms and other.
- Capability for tables upgrade after template changes.

One of the most important features of ACES Tables is its extensibility. It supports adding new templates and automation for new architectural artifacts and new modules for translation from and to new data formats. Thus, ACES Tables system has many capabilities for automatic specifications data processing and can allow using architectural specifications in system verification process.

Keywords: System on Chip, design process, verification, architectural specification, table, object-oriented approach, template, data import, data export

References

1. Lee Y. K., Kim N. H., Kim D., Lee D. H., In H. P. Customer Requirements Elicitation based on Social Network Service, *KSTT Transactions On Internet And Information Systems*, October 2011, vol. 5, no. 10, pp. 1733—1750.
2. Keller T. Contextual Requirements Elicitation. *Seminar in Requirements Engineering*, Spring 2011. Department of Informatics, University of Zurich.
3. Dhungana D., Seyff N., Graf F. Research Preview: Supporting End-user Requirements Elicitation Using Product Line Variability Models, *Requirements Engineering Foundation for Software Quality*, March 2011, vol. 6606, pp. 66—71.
4. Neetu K. S., Pillai A. S. A survey on global requirements elicitation issues and proposed research framework, *Software Engineering and Service Science (ICSESS). 2013. 4th IEEE International Conference*, 2013, pp. 554—557.
5. OpenSparc-T2 architectural specification. URL: <http://www.oracle.com/technetwork/systems/opensparc/opensparc-t2-page-1446157.html> (date of appeal: 01.07.2016) (in Russian).
6. Abraham R., Erwig M. Goal-directed debugging of spreadsheets. *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium*, 2005, pp. 37—44.
7. Abraham R., Erwig M. GoalDebug: A spreadsheet debugger for end users, *Proceedings of the 29th international conference on Software Engineering, IEEE Computer Society*, 2007, pp. 251—260.
8. Fisher M., Rothermel G., Creelan T., Burnett M. Scaling a Dataflow Testing Methodology to the MultiparadigmWorld of Commercial Spreadsheets, *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium*, 2006, pp. 13—22.
9. Abraham R., Erwig M. Mutation operators for spreadsheets, *Software Engineering, IEEE Transactions*, 2009, vol. 35, no. 1, pp. 94—108.
10. Abraham R., Erwig M., Kollmansberger S., Seifert E. Visual specifications of correct spreadsheets, *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium*, 2005, pp. 189—196.
11. Mendes J. Coupled evolution of model-driven spreadsheets. *Software Engineering (ICSE), 2012 34th International Conference*, 2012, pp. 1616—1618.
12. Cunha J., Fernandes J. P., Mendes J., Saraiva J. MDSheet: A framework for model-driven spreadsheet engineering. *Proceedings of the 34th International Conference on Software Engineering*, 2012, pp. 1395—1398.
13. Engels G., Erwig M. ClassSheets: automatic generation of spreadsheet applications from object-oriented specifications. *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*, 2005, pp. 124—133.
14. Cunha J., Fernandes J. P., Mendes J., Pacheco H., Saraiva J. Bidirectional Transformation of Model-Driven Spreadsheets. *International Conference on Theory and Practice of Model Transformations*, 2012, vol. 12, pp. 105—120.
15. Stevens P. Bidirectional model transformations in QVT: Semantic issues and open questions, *Model Driven Engineering Languages and Systems*, 2007, pp. 1—15.
16. Erwig M., Abraham R., Kollmansberger S. and Cooperstein I. Gencil: a program generator for correct spreadsheets, *Journal of Functional Programming*, 2006, vol. 16, no. 03, pp. 293—325.
17. Description of SystemRDL language. URL: <http://accelera.org/activities/working-groups/systemrdl> (date of appeal: 01.07.2016) (in Russian).
18. Burnett M., Cook C., Pendse O., Rothermel G., Summet J., Wallace C., 2003, May. End-user software engineering with assertions in the spreadsheet paradigm, *In Proceedings of the 25th international conference on Software engineering*, May 2003, pp. 93—103.

УДК 004.3

А. Б. Барский, д-р техн. наук, проф., e-mail: arkbarsk@mail.ru,
Московский государственный университет путей сообщения (МИИТ)

Нейросетевой метод ускоренного принятия решений операционной системой суперкомпьютера EPIC-архитектуры

Предлагается реализация систем принятия решений (СПР) в ОС суперкомпьютера EPIC-архитектуры, управляемого в каждом такте, с помощью однослойных логических нейронных сетей (ЛНС), допускающих распараллеливание и конвейеризацию действий без ветвления. В модели нейрокомпьютера используется принцип "зацепления" конвейеров для выполнения основной операции — скалярного умножения векторов при расчете значений функции активации нейронов. Сложение результатов умножения элементов векторов, получаемых на двух конвейерах умножения, производится по схеме двоичного дерева, где окончательный результат формируется в последнем регистре буфера быстрых регистров — в корневой вершине дерева. Такая схема "свертки" массива не прерывает конвейер сложения для ожидания промежуточных результатов. Рассматривается возможность "зацепления" четырех конвейеров умножения и двух конвейеров сложения для повышения производительности нейрокомпьютера.

Нейрокомпьютер может быть включен в пакет прикладных программ для универсального применения во многих системах распознавания, управления и принятия решений.

Ключевые слова: суперкомпьютер, EPIC-архитектура, логическая нейронная сеть, матрица связей, нейрокомпьютер, "зацепление" конвейеров

Введение

Идея применения нейросетевых технологий [1] для достижения сверхвысокой производительности вычислительной системы (ВС) привлекла внимание ввиду широкого параллелизма нетрудоемких вычислений при реализации принципа ассоциативного мышления человека [2] для решения трудноформализуемых задач и задач большой сложности.

EPIC-архитектура (*Explicitly Parallel Instruction Computing*) компьютера, как правило, в составе многопроцессорной ВС, использует принцип *программно управления каждым тактом машины* [3, 4]. Суперкомпьютер такой архитектуры, используемый для управления в реальном времени, призван удовлетворять требованиям сверхвысокой производительности. Его основные особенности:

1. Распараллеливание на уровне выполнения команд в многофункциональном арифметическо-логическом устройстве (АЛУ), состоящем из конвейерных исполнительных устройств (ИУ).

2. Возможность параллельного ветвления при обработке конвейера команд в двух случаях: при обработке команд условного перехода и при выполнении конструкций вида *if-then-else* в составе арифметических операторов.

Чтобы снизить влияние эффекта остановки конвейера команд, в EPIC-архитектуре по предвари-

тельной команде за несколько тактов до ветвления запускается дополнительный конвейер. Конвейеры начинают работать в симулятивном режиме, не допуская "порчи" данных. Интенсивное ветвление выполняемой программы способно привести к лавинообразному запуску дополнительных, аппаратно поддержанных конвейеров. Очевидно число таких конвейеров ограничено. Поэтому оптимально то программирование, при котором предпочтение отдается параллельному выполнению условных арифметических операторов (допускающих рекурсию), хотя это и приводит к избыточности оборудования, используемого при вычислениях. С помощью условных переходов следует поддерживать общую структуру программы, для обеспечения взаимодействия достаточно крупных работ. Следует также широко применять циклы типа "арифметическая прогрессия", где направление единственного перехода (выхода) предсказуемо заранее. Ведь целью оптимизации программирования суперкомпьютера является достижение минимального времени выполнения программы.

Таким образом, складывается культура программирования суперкомпьютера EPIC-архитектуры.

Управление ресурсами суперкомпьютера и многопроцессорной ВС, представляющее основу функций ОС, в общем случае сводится к анализу некоторой

ситуации — вектора измеренных или ожидаемых характеристик устройств или вычислительного процесса и к выработке решений по планированию дальнейшей эксплуатации вычислительных ресурсов, управлению вычислительным процессом, устранению неисправностей, визуализации и т. д.

В суперкомпьютерах широко используются модули ассоциативной памяти (АП) для кодирования и быстрой выборки отношений "причина — следствие". АП эффективна при организации виртуальной памяти, при обработке "глобалов", для хранения меток перехода, в системах контроля, в арбитрах коммутаторов, при решении конфликтов многоканального обмена и пр. Однако применение АП ограничено из-за существенных недостатков: 1) они не работают по нечетким данным; 2) они не используют взвешенные данные.

Любая система принятия решений (СПР) является универсальной и полной, если она предполагает нечеткие исходные данные и способна в процессе модернизации и развития учитывать различные факторы, и даже их значения, со своим весом.

Можно выделить ряд задач, решаемых ОС и являющихся основой создания многих СПР.

1. Принятие решения о реконфигурации вычислительной системы на основе вектора состояния ее средств.

2. Выбор оптимальной стратегии обслуживания очереди "к процессору" на основе текущего вектора характеристик этой очереди.

3. Выбор оптимальной стратегии управления многоканальным обслуживанием многопользовательского режима на основе λ -характеристик, приоритетного ряда и других показателей потоков заявок.

4. Обработка результатов контроля и диагностики вычислительных средств.

5. Разрешение конфликта при обмене по внутренним магистралям и в компьютерной сети в целом.

6. Адаптивный выбор пошагового смещения информационных пакетов в компьютерной сети.

7. Принятие решений по статистической оценке динамики загрузки модулей.

8. Экспертное управление надежностью средств системы, включающей разнородные устройства и объекты, оперирующих объектами разных типов, измерений и т. д.

9. Выбор оптимальной стратегии распараллеливания на основе характеристик потока заданий и данных и многие другие задачи.

Все указанные задачи при попытке найти точное оптимальное решение относятся к классу трудноформализуемых задач большой сложности, опирающихся на перебор. Практически используются эвристические методы решения, в основу которых берутся достаточно нетрудоемкие решающие правила. С конца 70-х — начала 80-х годов прошлого века особые надежды стали возлагать на применение методов искусственного интеллекта (ИИ) и, прежде всего, на нейросетевые технологии. В значи-

тельной степени это было связано с предполагаемой возможностью получить "быстрые" алгоритмы решения некоторых задач. Появилось понятие "эквивалентной" производительности при решении таких задач по сравнению с производительностью реальной.

Однако полную универсальность нейросетевые технологии обрели [5, 6] с построением и развитием *логических нейронных сетей* (ЛНС). ЛНС — универсальный аппарат моделирования ассоциативного мышления человека, определивший класс численных методов решения информационно-логических задач и задач управления.

1. Выбор моделей логических нейронных сетей

Нейрон (нейроподобный элемент) — прибор, выполняющий пороговую *функцию активации*.

Взаимосвязанные своими входами и выходами нейроны образуют *нейронную сеть*.

Функционально-ориентированная нейронная сеть, отображающая причинно-следственные связи или систему принятия решений, обладает (а) входным слоем рецепторов, возбуждаемым извне или обратными связями, и (б) выходным слоем, где максимальное возбуждение нейрона является или указывает на решение задачи.

Логическая нейронная сеть реализует систему распознавания, управления или принятия решений и строится по логическому описанию системой булевых функций в соответствии с требованием полноты и непротиворечивости. ЛНС сводится к однослойной, где одни и те же решения могут повторяться. ЛНС реализует базу знаний (БЗ) и является обученной.

На рецепторный слой ЛНС подаются *значения достоверности высказываний о принадлежности данных факторам факторного пространства событий*.

Нейроны выходного слоя не формируют численное решение, а лишь *указывают* на возможные принимаемые решения — в виде численных значений (скаляров или векторов), объектов любой природы, текстов, имен процедур, приказов и пр.

Алгоритм нахождения значения f_i , $i = 1, \dots, M$ (M — число нейронов), универсальной функции активации i -го нейрона (его значения возбуждения) имеет вид:

$$1. f := \frac{\sum_j p_j \omega_{ij}}{\sum_j \omega_{ij}}; \quad (1)$$

$$2. f_i := \begin{cases} f, & \text{если } f > h, \\ 0, & \text{в противном случае.} \end{cases}$$

Здесь p_j ($j \in \{1, \dots, N\}$, N — число рецепторов) — значение возбуждения j -го рецептора, связанного с i -м нейроном; ω_{ij} — не превышающее единицы значение веса связи j -го рецептора с i -м нейроном;

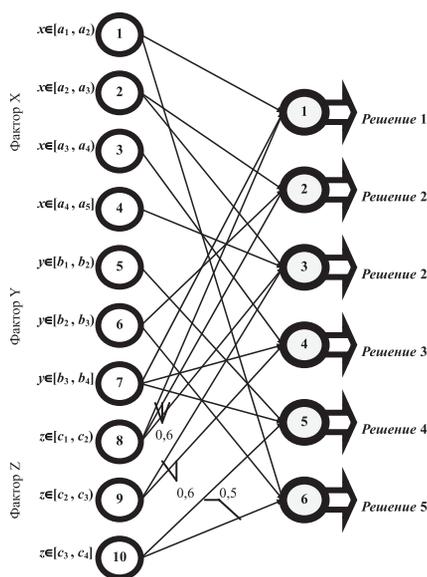


Рис. 1. Логическая нейронная сеть с закреплением рецепторов за интервалами значений факторов

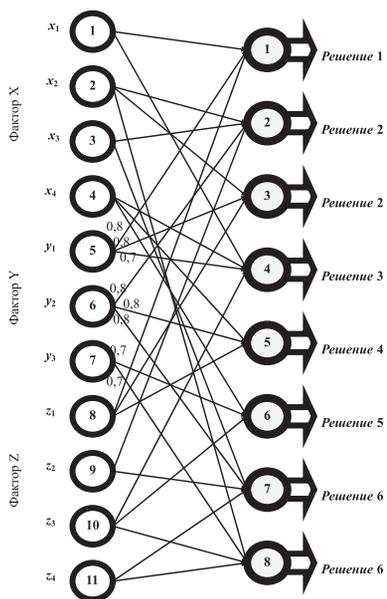


Рис. 2. Логическая нейронная сеть на основе экспериментов, моделирования или экспертных оценок

h — экспериментально (или по другим соображениям) подбираемый порог.

В применении к многочисленным задачам, решаемым в рамках ОС, целесообразно выделить два основных типа ЛНС, отличающихся организацией рецепторного слоя.

На рис. 1 дан пример ЛНС, где за рецепторами закреплены интервалы (закрытые или полуоткрытые) значений факторов. Присутствуют некоторые веса связей. Если вес на стрелке не указан, то он равен единице.

Практически, веса, отличные от единичных, становятся известны в результате длительной эксплуатации СПР. Ведь это — вопрос о том, какие факторы и насколько влияют на достоверность результата применения ЛНС, и целесообразно ли это учитывать. На начальном этапе разработки СПР, как правило, все веса принимаются единичными, если нет очевидных ограничений.

Второй тип ЛНС, отражающий результат экспериментальных исследований, моделирования или экспертных оценок, представлен на рис. 2.

Для обработки компьютером однослойная ЛНС представляется матрицей связи — таблицей. Для примера, приведенного на рис. 2, сопровождающего последующее изложение, такая таблица представлена ниже.

Табличное представление делает весьма удобными модификацию и развитие ЛНС в процессе обучения: добавление новых факторов или их значений, изменение связей, весов, уточнение решений, текст или процедуры которых хранятся отдельно. По таблице же находится указатель на принимаемое решение по максимуму возбужденному нейрону.

2. Возбуждение рецепторов

1. Если значения факторов заданы интервалами вида $[d, e]$, значение p (индекс опущен) возбуждения соответствующего рецептора, при анализе измеренного или предполагаемого значения g , находится как результат счета логического арифметического оператора:

$$p: = \text{if}((g \geq d) \wedge (g < e)) \text{ then } 1. \quad (2)$$

Если интервал имеет вид $[d, e]$, знак " $<$ " в (2) меняется на \leq .

2. Если интервал принадлежности испытываемого значения фактора определен нечетко, "единица" делится на число предполагаемых интервалов принадлежности, что определяет значение p для каждого такого интервала. Однако такие действия характерны при требовании полной автоматизации. Реально же значения p возбуждения рецепторов, закрепленных за предполагаемыми интервалами, следует задавать в диалоговом режиме на основе экспертных оценок или предположений оператора-пользователя.

3. Если измеренное или предполагаемое численное значение g является промежуточным, не

Матрица связей

Рецепторы	1 x_1	2 x_2	3 x_3	4 x_4	5 y_1	6 y_2	7 y_3	8 z_1	9 z_2	10 z_3	11 z_4	1/Сумма весов	Указатель на решение
P — вектор значений возбуждения рецепторов	0,5	0,5				0,66	0,33		0,8	0,2			
Нейрон 1, вектор весов W_1	1				0,8			1				0,36	Процедура Решение 1
Нейрон 2, вектор весов W_2		1	1			0,8			1			0,26	Процедура Решение 2
Нейрон 3, вектор весов W_3			1		0,8			1				0,36	Процедура Решение 2
Нейрон 4, вектор весов W_4	1			1	0,7					1		0,27	Процедура Решение 3
Нейрон 5, вектор весов W_5				1		0,8		1				0,36	Процедура Решение 4
Нейрон 6, вектор весов W_6				1			0,7			1		0,37	Процедура Решение 5
Нейрон 7, вектор весов W_7			1			0,8			1		1	0,26	Процедура Решение 6
Нейрон 8, вектор весов W_8		1					0,7			1	1	0,27	Процедура Решение 6

совпадающим ни с каким значением, которому соответствует рецептор, находятся два ближайших (меньшее d и большее e) значения фактора, за которыми закреплены рецепторы ("ближайшие рецепторы"), и "единица" распределяется обратно пропорционально "расстоянию" до этих рецепторов:

$$p_d = (e - g)/(e - d); p_e = (g - d)/(e - d). \quad (3)$$

Предлагается параллельный счет значений p_d и p_e (ведь $p_e = 1 - p_d$). Рекомендуется значения вида $1/(e - d)$ рассчитывать заранее и хранить до модификации ЛНС. Счет обратной величины на этапе формирования (обучения) ЛНС предпочтителен, так как конвейер деления значительно "длиннее" конвейера умножения, оперативно используемого в рабочем режиме.

4. Если значение g находится вне диапазона значений, представленных рецепторами, т. е. для него можно указать лишь одно значение d или e , за которым закреплен рецептор, то вместо разности $(e - d)$, используемой в (3), можно взять среднее "расстояние" δ между рецепторами сети и выполнить один из операторов:

$$\begin{aligned} p_d &:= \text{if } (g < d) \text{ then } (1 - (d - g)/\delta); \\ p_e &:= \text{if } (g > e) \text{ then } (1 - (g - e)/\delta). \end{aligned} \quad (4)$$

5. Если фактор предполагает нечисловые значения, возбуждение рецепторов задает пользователь в диалоговом режиме.

Например, пусть в ЛНС СПР, обслуживающей охотников (см. рис. 1), рецептор x_1 фактора X соответствует высказыванию "рысь", а рецептор x_2 — высказыванию "дикая кошка". Никто из охотников не смог с полной уверенностью определить, следы принадлежат рыси или дикой кошке? Они полагают значение возбуждения рецепторов 1 и 2, равными 0,5 ($p_1 = p_2 = 0,5$). Пусть фактор Y отражает число охотников. Их оказалось шесть человек. Рецептора, закрепленного за таким количеством, нет. Однако "ближайшими" рецепторами являются y_2 , отвечающий за пять человек, и y_3 , отвечающий за восемь человек. Возбуждаем эти рецепторы распределением "единицы" обратно пропорционально "расстоянию" исследуемого значения b до них: $p_6 = 0,66, p_7 = 0,33$. Пусть на основе подобных рассуждений заполнены клетки фактора Z в таблице: $p_9 = 0,8, p_{10} = 0,2$.

На больших ЛНС с помощью рецепторов можно имитировать любой закон распределения вероятностей, в том числе нормальный.

Таким образом, ЛНС работает с нечеткими данными, что составляет основной смысл ассоциативного мышления.

3. Схема поиска решения

Общая схема нахождения оптимального решения с помощью ЛНС показана на рис. 3. В процедуру НЕЙРОКОМПЬЮТЕР выделены блоки, которые реализуют собственно нейροкомпьютер. Процеду-

ра может быть использована в различных приложениях, не обязательно только в составе ОС.

Легко заметить, что самой трудоемкой операцией здесь является нахождение скалярного произведения вектора значений возбуждения рецепторов и вектора, образуемого строкой весов связей нейрона. Затем значение найденного произведения умножается на соответствующий элемент последнего столбца таблицы — матрицы связей и сравнивается с порогом h для получения неотрицательного значения функции активации (1). Попутно фиксируется нейрон с максимальным значением возбуждения, указывающий на рекомендуемое или автоматически выполняемое решение — по завершении анализа всех нейронов.

Порог h выбирают экспериментально так, чтобы сократить анализ лишних, заведомо отвергнутых вариантов решения.

Если значение возбуждения ни одного нейрона не превысило порог, следует, изучив ситуацию, либо повысить порог, либо "научить" ЛНС новой для нее ситуации.

Таким образом, с помощью ЛНС проводится "голосование" в пользу "наиболее возбуждшегося" решения. Однако, если находятся численные значения, например значение рейтинга объекта, то на основе значений возбуждения нескольких нейронов, преодолевших порог, может быть найдено среднее значение искомой величины по той же

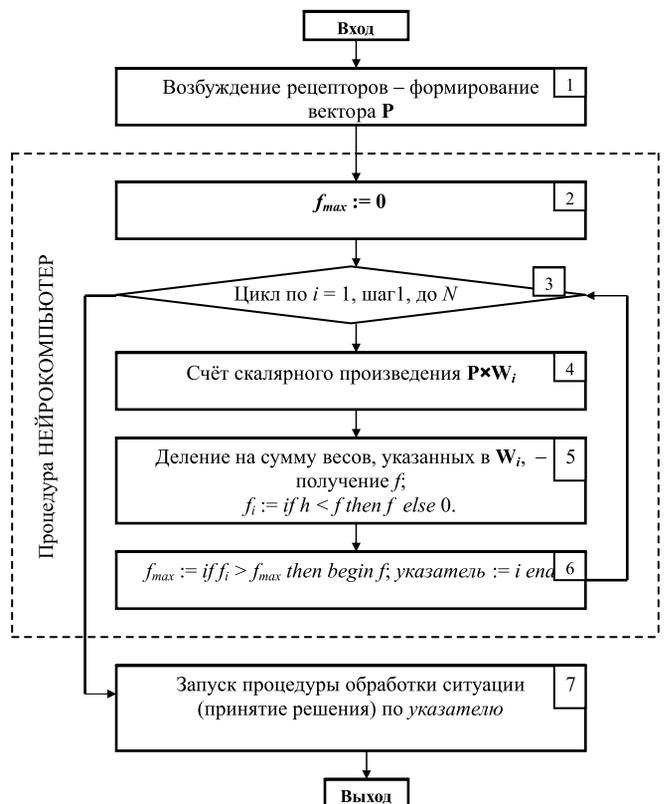


Рис. 3. Схема принятия решения и процедуры НЕЙРОКОМПЬЮТЕР

формуле, что лежит в основе функции активации. Тогда значения возбуждения нейронов используют в качестве весов решений, на которые эти нейроны указывают.

Например, пусть в некоторой, здесь не приведенной ЛНС значение возбуждения нейрона 1, указывающего на ответ "необходимо 40 человек", равно 0,5. Значение возбуждения нейрона 2, указывающего на ответ "необходимо 50 человек", равно 0,8, а значение возбуждения нейрона 3, указывающее на 60 человек, равно 0,6. Возбуждение других нейронов нулевое. Тогда среднее число людей, рекомендуемых ЛНС, составляет

$$\frac{40 \cdot 0,5 + 50 \cdot 0,8 + 60 \cdot 0,6}{0,5 + 0,8 + 0,6} = 50,5 \text{ человек.}$$

Примеры говорят об универсальности ЛНС, их применимости при совокупном, одновременном рассмотрении объектов разной природы, разных типов данных, измерений и др. Это адекватно воспроизводит аппарат логического мышления человека.

Закончим рассмотрение примера. Пусть обосновано значение порога $h = 0,4$. По рис. 2 и таблице рассчитаем значения возбуждения всех нейронов. Все равны нулю, кроме $f_2 = 0,5$. Как максимальное

возбуждение нейронов, это значение указывает на Решение 2.

4. Оптимизация скалярного умножения векторов

Рассмотрим возможность оптимизации (по времени выполнения) скалярного умножения векторов \mathbf{P} и \mathbf{W} (индекс опускаем), отраженного в блоке 4 на рис. 3. Оптимизация достигается применением двух методов:

1. "Зацеплением" двух векторных конвейеров умножения с векторным конвейером сложения.

2. Сложением всех результатов умножения с помощью операции "свертки" массива [4] по схеме "двоичного дерева", что обеспечивает непрерываемый счет на векторном конвейере сложения с окончательным получением суммы в корневой ячейке дерева.

Схема счета по таблице показана на рис. 4.

Итак, даны два вектора $\mathbf{P} = \{p_1, \dots, p_{11}\}$ и $\mathbf{W} = \{\omega_1, \dots, \omega_{11}\}$ ($N = 11$).

Пусть один конвейер умножения обрабатывает нечетные пары элементов, а второй — четные пары элементов. (Обычно используется индексация с нуля; индексация с единицы более наглядно отождествляется с порядком действий.) Рабочий массив \mathbf{A} содержит 21 регистр, $\mathbf{A} = \{a_1, \dots, a_{21}\}$ ($(2N - 1)$ регистров). Конвейеры умножения заполняют первые N регистров. Конвейер сложения запускается тотчас по получении первой пары произведений и синхронно с умножением выполняет сложение по двоично-древесной схеме, как показано на рис. 4.

Схема регулярных действий конвейера сложения при обработке рабочего массива регистров показана на рис. 5. Для краткости используются только индексы (номера) регистров.

При отсутствии в АЛУ векторных конвейеров со специальной аппаратной поддержкой схеме этой обработки соответствует цикл типа "арифметическая прогрессия":

$i = 1$; для $j = 1$ до $N - 1$ выполнять:
begin $a_{N+j} = a_i + a_{i+1}$; $i = i + 2$; *end*.

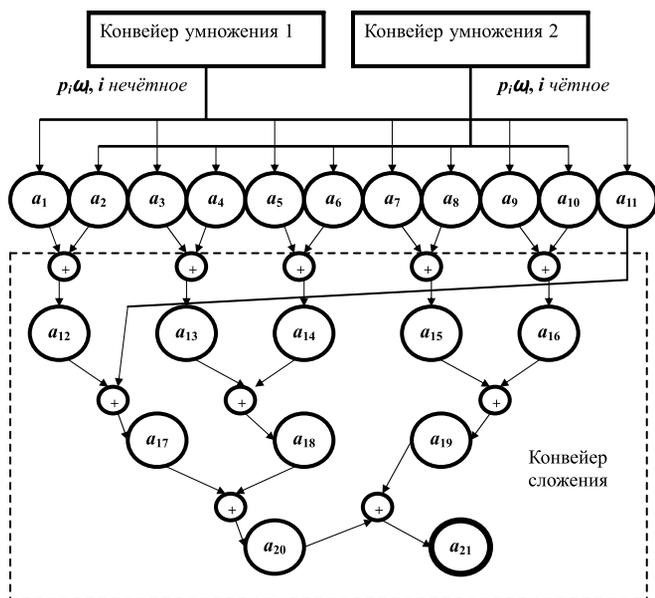


Рис. 4. Схема двоично-древесной обработки буфера

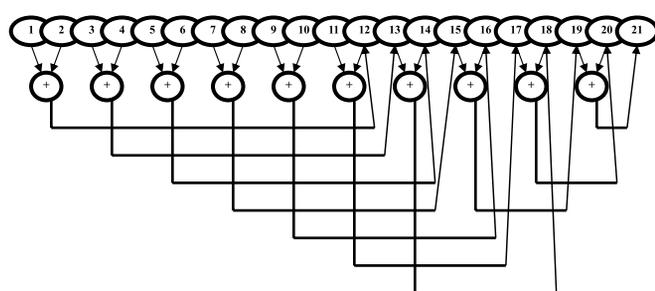


Рис. 5. Схема обработки буфера векторным конвейером сложения

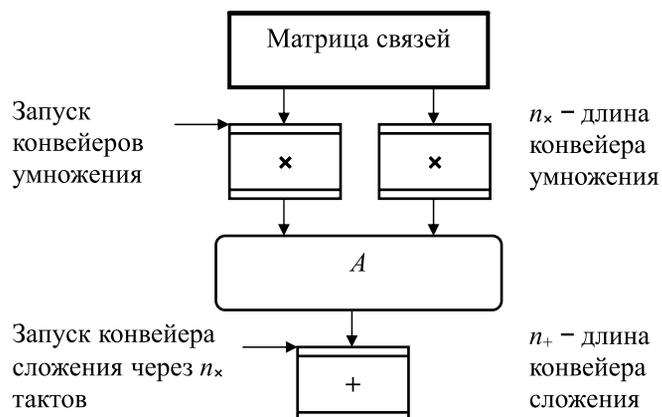


Рис. 6. Последовательность запуска векторных конвейеров

(Выполняя "набросок" программы, мы не следуем канонам какого-либо языка.)

На рис. 6 показана синхронная работа векторных конвейеров при управлении каждым тактом машины.

5. Распараллеливание скалярного умножения векторов

Если АЛУ содержит достаточное число конвейерных ИУ, возможности распараллеливания могут быть расширены. Это относится к скалярному умножению векторов, как основной операции нейрокомпьютера.

На рис. 7 показано использование четырех векторных конвейеров умножения и двух векторных конвейеров сложения для ранее рассмотренной нейросети.

В представленных случаях управления при достаточно большом числе рецепторов в каждом такте синхронизации использования данных не требуется, если в качестве буфера не используется оперативная память (ОП). ОП, вследствие конфликтов при обращении, вносит элемент нестабильности в режим управления каждым тактом компьютера. Большой объем ЛНС может привести к активному

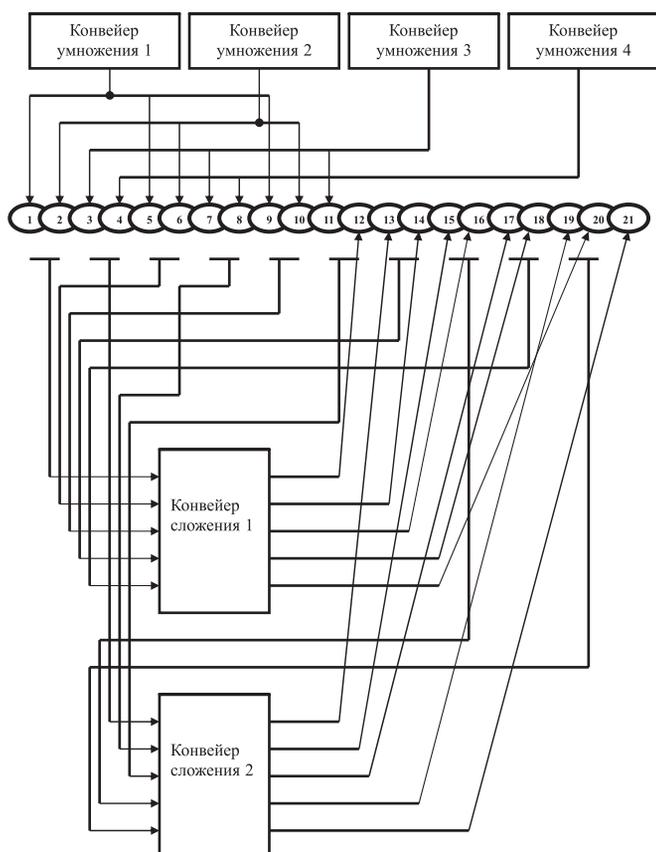


Рис. 7. "Защепление" четырех конвейеров умножения с двумя конвейерами сложения

использованию ОП, если объема сверхоперативного запоминающего устройства (СОЗУ) станет недостаточно. Тогда синхронизация может быть выполнена с помощью *закрытия адресов по считыванию*, которые открываются по окончании записи по ним. В процессорах "Эльбрус" такой синхронизации служат *биты значимости* [3].

Заключение

Теория логических нейронных сетей, построенная на основе математической логики событий, сделала возможным решение самого широкого класса трудноформализуемых задач, а также задач высокой сложности — оперативного управления, обработки изображений, объектов временного ряда, контроля и диагностики сложных систем. ОС компьютера реализует много таких задач. Необходимо преодолеть психологический барьер для превращения "быстрого" метода нейросетевой обработки информации в простой и привлекательный прием программирования многих приложений как в составе ОС, так и в качестве сервисных средств или в составе пакета прикладных программ.

Вместе с тем специфика решаемых задач диктует целесообразное развитие архитектуры суперкомпьютера, в частности, системы команд и аппаратной поддержки операций [7—9]. Это обнаруживается при попытке программно реализовать нетрудоемкий нейрокомпьютер на суперкомпьютере EPIC-архитектуры.

Затраты на создание эффективного нейрокомпьютера значительно ниже тех затрат производительности, которые во многих случаях необходимы для последовательного анализа возникшей ситуации и принятия по ней решения.

Список литературы

1. **Галушкин А. И.** Нейронные сети: основы теории. М.: Горячая линия — Телеком, 2010. 496 с.
2. **Поспелов Д. А.** Моделирование рассуждений. Опыт анализа мыслительных актов. М.: Радио и связь, 1989. 184 с.
3. **Бабаян Б. А., Бочаров А. В., Волин В. С.** и др. Многопроцессорные ЭВМ и методы их проектирования. М.: Высшая школа, 1990. 142 с.
4. **Барский А. Б.** Параллельные информационные технологии. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007. 503 с.
5. **Барский А. Б.** Нейронные сети: распознавание, управление, принятие решений. М.: Финансы и статистика, 2004. 175 с.
6. **Барский А. Б.** Нейросетевые методы оптимизации решений. СПб.: Интермедия, 2016. 293 с.
7. **Барский А. Б.** Оптимальное потактовое расписание выполнения работ в многофункциональном арифметическо-логическом устройстве суперскаляра // Информационные технологии. 2003. № 5. С. 2—12.
8. **Барский А. Б., Шилов В. В.** Оптимизация ветвления при решении задач сортировки на процессоре EPIC-архитектуры // Информационные технологии. 2005. № 1. С. 26—34.
9. **Барский А. Б., Шилов В. В.** Оптимизация предикатных вычислений при решении задач поиска на процессоре EPIC-архитектуры // Информационные технологии. 2005. № 5. С. 2—5.

Neural Network Method of Accelerated Adoption of the Operating System, Making Supercomputer EPIC-Architecture

The operating system (OS) supercomputer, solving problems in real-time, embodies many of the features that monitor the operating parameters: system configuration and redundancy characteristics of technical means and their reliability, control and diagnostic results, performance multi-channel service job flow, etc. According to the settings received, often fuzzy, data representing the point of the factor space and the events defining the situation, quickly make different decisions. Factors or values may be carried out with weights.

Proposed the implementation of decision support systems (DSS) in the OS supercomputer EPIC-architecture that is managed in each step, with the help of single-layer logical neural networks (LNS), allowing parallelization and pipelining of action in the absence of branching.

When building a program of neurocomputers procedure uses the principle of "engagement" conveyors to implement the basic operations — scalar multiplication of vectors. The two conveyors (for even and odd indices) multiplying the excitation vector to vector receptor neuron weights, upon receipt of the next intermediate results immediately transmit them to the addition of the pipeline through the buffer registers fast. This conveyor is made of pair wise addition on the scheme of a binary tree. The final result of the scalar multiplication of vectors produced in the last buffer register — in the root node of the tree. Such a scheme "fold" of the array does not interrupt the pipeline, as devoid of having to wait for intermediate results of summation.

The possibility of "engagement" of four conveyors multiplication and of the two conveyors addition to increase productivity neurocomputer twice.

Neurocomputer can be included in the application package for universal use in many recognition systems, control and decision-making.

Keywords: supercomputer, the EPIC-architecture, logic neural network, matrix connections, neurocomputer, "engagement" conveyors

References

1. Galushkin A. I. *Nejronnye seti: osnovy teorii*, Moscow, Goryachaya linlya — Telekom, 2010, 496 p. (in Russian).
2. Pospelov D. A. *Modelirovanie racyzhdenij. Opyt analiza myslitelnyh aktov*, Moscow, Radio i svyaz', 1989, 184 p. (in Russian).
3. Babayan B. A., Bocharov A. V., Volin V. C. et al. *Mnogoprocessornye EVM i metody ih proektirovaniya*, Moscow, Vysshaya shkola, 1990, 142 p. (in Russian).
4. Barsky A. B. *Parallel'nye informacionnye tehnologii*, Moscow, INTUIT; BINOM. Laboratoriya znaniy, 2007, 503 p. (in Russian).
5. Barskij A. B. *Nejronnye seti: raspoznavanie, upravlenie, prinyatie reshenij*, Moscow, Finansy i statistika, 2004, 175 p. (in Russian).
6. Barsky A. B. *Nejrosetevye metody optimizacii reshenij*, Sankt-Peterburg, Intermediya, 2016, 293 p. (in Russian).
7. Barsky A. B. Optimal'noe potaktovoe raspisanie vypolneniya rabot v mnogofunkcional'nom arifmeticheskoy-logicheskom ustrojstve superskalyara, *Informacionnye tehnologii*, 2003, no. 5, pp. 2—12 (in Russian).
8. Barsky A. B., Shilov V. V. Optimizaciya vetvleniya pri reshenii zadach cortirovki na processore EPIC-arhitektury, *Informacionnye tehnologii*, 2005, no. 1, pp. 26—34 (in Russian).
9. Barsky A. B., Shilov V. V. Optimizaciya predikatnyh vychislenij pri reshenii zadach poiska na processore EPIC-arhitektury, *Informacionnye tehnologii*, 2005, no. 5, pp. 2—5 (in Russian).

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 10.03.2017. Подписано в печать 25.04.2017. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT517. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

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