

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 26

2020

№ 2

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ
И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И
НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ
СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС

Рисунки к статье Г. В. Данилова, А. А. Потапова, И. Н. Пронина, М.А. Шифрина, В. А. Васенина, А. Е. Подоприторы, В. В. Назарова, Н. В. Ласунина, Т. Е. Пронкиной, Ю. В. Струниной, Ю. А. Дорофеек

«МЕТОДОЛОГИЯ ОЦЕНКИ АКТИВНОСТИ И РЕЗУЛЬТАТИВНОСТИ НАУЧНОЙ ДЕЯТЕЛЬНОСТИ В МЕДИЦИНСКОЙ ОРГАНИЗАЦИИ: ОПЫТ НАЦИОНАЛЬНОГО МЕДИЦИНСКОГО ИССЛЕДОВАТЕЛЬСКОГО ЦЕНТРА НЕЙРОХИРУРГИИ ИМЕНИ АКАДЕМИКА Н.Н. БУРДЕНКО»

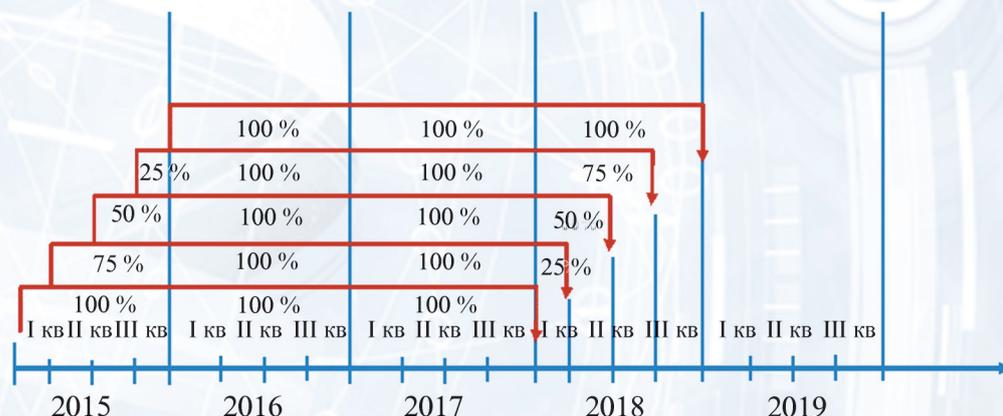


Рис. 1. Схема ежеквартального расчета рейтинговой оценки научной активности и результативности

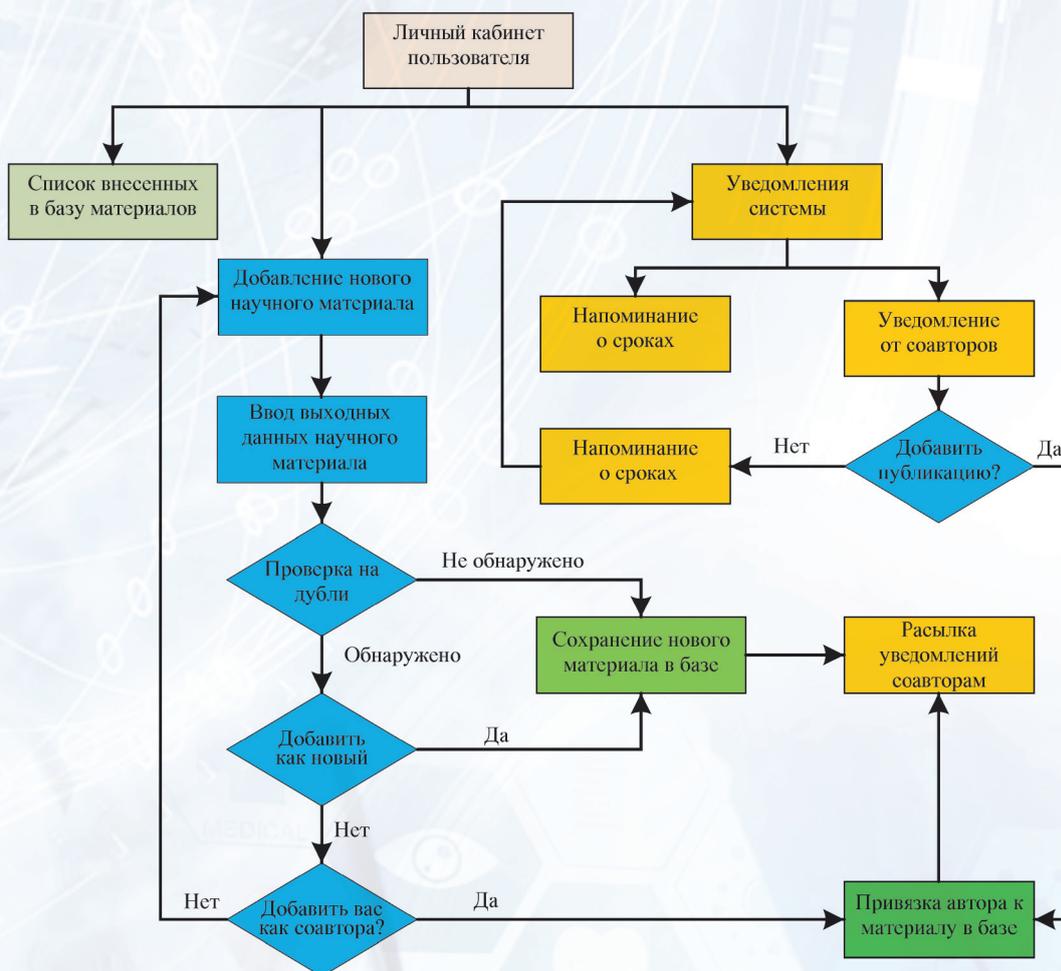


Рис. 2. Блок-схема процесса работы специализированной системы

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 26
2020
№ 2

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

Попов А. Ю. Принципы организации гетерогенной вычислительной системы с набором команд дискретной математики 67

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Пинт Э. М., Петровнина И. Н., Романенко И. И., Еличев К. А. Разработка метода распознавания печатных символов (букв, цифр) для программы персонального компьютера 80

Корнева Н. Н., Назаров В. Н., Могилевский М. М. Архитектура инструмента для визуального анализа спутниковых измерений в поисковых задачах . . 87

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Данилов Г. В., Потапов А. А., Пронин И. Н., Шифрин М. А., Васенин В. А., Подопригора А. Е., Назаров В. В., Ласунин Н. В., Пронкина Т. Е., Струнина Ю. В., Дорофеюк Ю. А. Методология оценки активности и результативности научной деятельности в медицинской организации: опыт Национального медицинского исследовательского центра нейрохирургии имени академика Н. Н. Бурденко 97

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

Соломатин А. Н. Интегрированные системы регионального программирования и автоматизация их разработки 107

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

Кузьмин Ю. Б. Шкалирование уровня автоматизации на производстве 116

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

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

Редакционная коллегия:

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

Редакция:

БЕЗМЕНОВА М. Ю.

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

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 26
2020
No. 2

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

COMPUTING SYSTEMS AND NETWORKS

Popov A. Yu. Principles of the Organization of a Heterogeneous Computing System with a Set of Commands of Discrete Mathematics 67

DIGITAL PROCESSING OF SIGNALS AND IMAGES

Pint E. M., Petrovnina I. N., Romanenko I. I., Yelichev K. A. Development of the Principal Symbol Recognition Method (Letters, Digital) for the Personal Computer Program 80

Korneva N. N., Nazarov V. N., Mogilevsky M. M. The Instrument Architecture for Visual Analysis of Satellite Measurements in Exploratory Problems 87

APPLICATION INFORMATION SYSTEMS

Danilov G. V., Potapov A. A., Pronin I. N., Shifrin M. A., Vasenin V. A., Podopri- gora A. E., Nazarov V. V., Lasunin N. V., Pronkina T. E., Strunina Y. V., Doro- feyuk Y. A. Methodology of Activity and Scientific Research Efficiency Evaluation in Medical Organization: N. N. Burdenko National Medical Research Center of Neurosurgery Experience 97

INFORMATION TECHNOLOGIES IN ECONOMICS AND MANAGEMENT

Solomatin A. N. The Integrated Systems of Regional Programming and Automation of their Development 107

AUTOMATED CONTROL SYSTEMS FOR TECHNOLOGICAL PROCESSES

Kuz'min Yu. B. Taxonomy the Level of Automation in Production 116

Editor-in-Chief:

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

Deputy Editor-in-Chief:

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

Chairman:

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

Editorial Board Members:

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

Editors:

Bezmenova M. Yu.

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

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

А. Ю. Попов, канд. техн. наук, доц., e-mail: alexpopov@bmstu.ru,
Московский государственный технический университет им. Н. Э. Баумана, г. Москва

Принципы организации гетерогенной вычислительной системы с набором команд дискретной математики

В МГТУ им Н. Э. Баумана разработана и продолжает совершенствоваться вычислительная система с набором команд дискретной математики. Система состоит из вычислительных устройств с различной архитектурой и позволяет пользовательским программам обращаться к аппаратным функциям хранения и обработки больших множеств, структур данных, графов и других математических абстракций, используемых в задачах дискретной оптимизации. Основу системы составляет принципиально новое вычислительное устройство — процессор обработки структур Leonhard x64, обладающий высоким быстродействием и функциональностью.

В предыдущих работах продемонстрированы результаты проектирования вычислительной системы, раскрыты особенности выполнения программ и обработки данных, представлена структура нового процессора, приведены результаты экспериментов по измерению производительности.

В данной статье мы обобщаем и систематизируем ряд ключевых принципов построения системы с набором команд дискретной математики, которые в первую очередь обеспечивают высокий уровень производительности. Мы последовательно излагаем причину введения каждого из принципов и следствия его внедрения, демонстрируем примеры их реализации. Представленная статья имеет важный теоретический смысл для всего проекта, так как многие из представленных результатов сформулированы впервые.

Ключевые слова: структура данных, вычислительная система с набором команд дискретной математики, система со многими потоками команд и одним потоком данных, процессор обработки структур, $V + \text{дерево}$

Введение

Наборы команд современных микропроцессоров достигают нескольких сотен машинных инструкций, однако число вычислительных команд в них ограничено: это команды выполнения основных арифметических действий (сложение, вычитание, умножение и деление), команды математической логики (конъюнкция, дизъюнкция, отрицание, эквивалентность), сдвиги операндов, команды выполнения прямых и обратных тригонометрических функций, логарифмирование, извлечение корня и некоторое число команд обработки векторов. С точки зрения поддержки математических операций эти команды покрывают лишь часть востребованных в вычислениях действий. Непосредственно поддерживаются лишь такие разделы математики, как арифметика, математическая логика и тригонометрия. Остальные же разделы математики, в том числе и дискретная математика и принятые в них операции, обеспечи-

ваются лишь посредством трансляции в существующий набор арифметических и логических команд. При этом большинство исполняемых в вычислительном потоке команд не связаны с вычислениями напрямую, а носят сопутствующий характер. Это такие типы команд, как пересылка данных, ветвления и команды управления (переходы, передача управления, возвраты и т. д.).

Анализ кода программ показывает [1–3], что команды пересылки составляют около 50 % потока инструкций, обрабатываемых процессором. Этот факт свидетельствует о проявлении проблемы семантического разрыва, когда большинство команд не связаны напрямую с вычислениями, либо обеспечивают аппаратную поддержку лишь части математических действий. Другие команды, такие как поиск минимума/максимума в блоке векторной обработки (SSE Unit, Streaming SIMD Extensions), закрывают лишь незначительную часть потребностей в аппаратной поддержке вычисли-

тельных алгоритмов. Вместе с тем при решении практических задач на ЭВМ используется существенно большее число математических операций дискретной математики, математической логики, дифференциального и интегрального исчисления, линейной алгебры, аналитической геометрии, математической статистики и других. Однако очевидно, что их реализация требует существенного изменения в структуре электронных вычислительных машин и является крайне трудоемкой. В данной работе мы более подробно остановимся на вопросах аппаратной поддержки операций дискретной математики, которая проводится в рамках экспериментальных научно-исследовательских работ в МГТУ им. Н. Э. Баумана.

Обзор существующих подходов к ускорению алгоритмов дискретной оптимизации

Наиболее перспективным способом повышения производительности компьютерных систем является применение специализированных аппаратных ускорителей (акселераторов). В многочисленных работах по данной тематике описаны примеры устройств, которые позволяют существенно снизить время выполнения как отдельных вычислительных операций, так и целых алгоритмов. Наиболее выдающиеся примеры ускорения для задач дискретно-оптимизационного характера достигаются благодаря применению графических ускорителей GPGPU (General-purpose computing on graphics processing units), а также специализированных ускорителей на основе СБИС и программируемых логических интегральных схем (ПЛИС типа FPGA). Так, благодаря применению платформы GPGPU NVidia Tesla удается ускорить алгоритм поиска кратчайшего пути на графе (Single-source shortest path, SSSP) до 400 раз [4]. Благодаря применению специализированного устройства на основе FPGA достигается ускорение преобразования Хафа в 370 раз [5]. Также в работе [5] приведены примеры ускорения вычислений статистических функций, алгоритмов биоинформатики и алгоритмов финансового анализа от 50 до 100 раз в сравнении с универсальными компьютерами. Неплохих результатов удается достичь и в области криптографии [6].

При этом стоит отметить ряд ограничений и недостатков как технологии ускорения на основе GPGPU, так и специализированных ускорителей на основе СБИС и FPGA. Было выявлено, что эффективность решения задач

на GPGPU существенно зависит как от формата представления данных в локальной памяти графического процессора, так и от возможности представления алгоритма в виде параллельных потоков вычислений. Например, для алгоритма SSSP традиционный вариант представления графа в виде матрица смежности [7, 8] оказывается малоприменимым для реализации на GPGPU в связи с сильной разреженностью исходных данных. Производительность графических ускорителей при таком представлении графа оказывается низкой (в сравнении с универсальными микропроцессорами). Другими проблемами применения GPGPU являются физические ограничения на объемы исходных данных алгоритма, размещаемых во внутренней кэш-памяти, а также необходимость их копирования из оперативной памяти центрального процессора в пространство памяти ускорителя GPGPU.

Вместе с тем существенным недостатком специализированных ускорителей на основе СБИС и FPGA является их низкая универсальность и высокая трудоемкость их разработки: такое устройство изначально создается для решения конкретной алгоритмической задачи и оказывается непригодным для решения других подобных проблем. Временные затраты на организацию системной логики (высокоскоростных шин передачи данных, системного программного обеспечения, средств компиляции кода, программных интерфейсов и пр.) оказываются настолько существенными, что не позволяют в разумные сроки представить такой ускоритель заказчику. Как следствие, разработка высокопроизводительных алгоритмов была и во многом остается трудоемкой научно-исследовательской, а не чисто инженерной задачей.

Понимание указанных недостатков позволило сформулировать ряд первичных требований к разрабатываемой вычислительной системе:

- вычислительная система с аппаратной поддержкой операций дискретной математики должна обладать достаточной универсальностью для ускорения широкого круга задач дискретной оптимизации;
- требуется пересмотреть существующие и разработать ряд новых базовых принципов функционирования вычислительной системы в целях повышения ее эффективности при решении задач дискретной оптимизации;
- архитектура вычислительной системы должна объединять различные вычислительные средства (универсальные микропроцессоры, ускорители вычислений, специализи-

рованные микропроцессоры, контроллеры памяти и системных шин, специализированную память) и поддерживать простое и эффективное программное управление ими;

- необходимо использовать доступную элементную базу ПЛИС FPGA и СБИС для скорейшей реализации системы.

Исходя из сказанного создание действующего образца такой системы в обозримом будущем может показаться маловероятным в связи с противоречивостью требований и масштабностью постановки задачи. Тем не менее, на данный момент уже разработана и прошла цикл испытаний третья версия системы Leonhard x64, она успешно функционирует в составе Центра обработки данных МГТУ им. Н. Э. Баумана, а для заинтересованных разработчиков к ней открыт удаленный доступ. По результатам исследований опубликован ряд работ, в которых отражены как базовые аппаратные принципы [9–12], так и результаты решенных с ее помощью прикладных задач [15, 16].

В предыдущих работах были проанализированы принципы организации существующих ЭВМ и систем, влияющие на эффективность обработки структур данных, разработана общая концепция функционирования вычислительных систем с аппаратной поддержкой операций над структурами данных [9, 10], исследованы способы ускорения вычислительных операций в такой системе [10, 11], разработан и реализован специализированный процессор обработки структур данных. На основе этого устройства создан прототип вычислительной системы [12], разработан набор команд и средства компиляции программ, выполнено проектирование ряда алгоритмов оптимизации [10, 11], проведено тестирование и сравнение полученного решения с существующими универсальными вычислительными системами [12].

По мере дальнейших исследований многие подходы потребовали уточнения и более последовательного изложения. Далее будут представлены шесть основных принципов функционирования системы.

Принципы построения ЭВМ с набором команд дискретной математики

Структуры данных являются ключевым и фундаментальным понятием в программировании и широко применяются в таких классах задач дискретной математики, как задачи опти-

мизации на сетях и графах, общие задачи комбинаторной оптимизации, задачи управления. Исследованиям новых структур, алгоритмам выполнения операций и принципам их применения посвящен ряд фундаментальных трудов [7, 8]. Однако в применяемых в настоящий момент вычислительных системах вся обработка алгоритмов и структур данных проводится на универсальных микропроцессорах, в которых аппаратно реализованы лишь арифметическая и логическая обработки числовых значений.

Следует отметить, что несмотря на большой интерес научного сообщества к созданию и применению различных ускорителей примеров построения эффективной платформы ускорения задач дискретно-оптимизационного характера и структур данных крайне мало. Наиболее близкими по технической сущности можно считать ассоциативные процессоры CAPP (*Content Addressable Parallel Processors*) [13], проекты STARAN и ASPRO, построенные на основе процессорных элементов (ПЭ), объединенных сложной сетью передачи информации [14]. Каждый 8-разрядный ПЭ в таких системах обрабатывает только свою локальную память, выполняя над ее содержимым простейшие арифметические и логические операции. Объединение большого числа простых процессорных элементов позволяет достичь высокого уровня параллельности. В каждый момент времени все процессорные элементы CAPP обрабатывают одну и ту же команду, в связи с чем такая архитектура относится к классу SIMD. В работе [14] исследуется возможность реализации ассоциативных операций над таблицей реляционной базы данных, хранимой в таком процессоре (рассмотрены такие операции, как Поиск, Следующий, Первый, Максимум, Минимум, И-ИЛИ-НЕ операции, Сумма, Количество). Помимо достоинств, ассоциативные процессоры CAPP имеют ряд недостатков:

- требуется реализация сложной сети передачи информации, необходимой для выполнения ряда ассоциативных операций;
- процессор обладает высокой удельной стоимостью единицы обрабатываемой информации (т. е. число логических элементов процессора в расчете на одно обработанное информационное слово достаточно велико);
- реализация операций над несколькими множествами (объединение, пересечение множеств и т. д.) сильно затруднено;
- в связи с ассоциативным характером памяти затрудняется возможность хранения и обработки больших массивов информации.

В связи с вышеизложенными недостатками известных универсальных и специализированных систем был сформулирован первый принцип, определяющий гетерогенный характер дискретно-оптимизационного вычислительного процесса.

Принцип 1 гетерогенности вычислений. Для разработки вычислительной системы с аппаратной поддержкой операций дискретной математики необходимо реализовать специальные аппаратные средства обработки множеств и структур данных. Это позволит построить архитектуру вычислительной системы таким образом, чтобы множества отношений и множества данных обрабатывались параллельно и независимо.

Архитектурные принципы современных вычислительных машин и систем, как показано в ряде работ [9, 12], не способствуют ускорению обработки ссылочных структур данных. Расслоение оперативной памяти ведет к замедлению обращений по произвольным адресам. Пакетный режим передачи информации между процессором и памятью делает неэффективной загрузку небольших информационных слов, а сегментно-страничная организация виртуальной памяти приводит к двойным обращениям к памяти. Отказ же от этих принципов может ускорить обработку ссылочных структур (деревьев, списков), но негативно отразится на обработке векторных структур данных (массивов, хэш-таблиц) и функциональности ЭВМ. Таким образом, дальнейшее совершенствование архитектуры вычислительных машин в рамках существующих принципов (например, увеличение размеров пакетов и страниц, увеличение объемов памяти) будет по-прежнему приводить к противоречивым результатам: при попытке ускорить работу одних алгоритмов будет происходить замедление при обработке других. Программная модель вычислительной машины представляет архитектуру ЭВМ в упрощенном виде, что приводит к разрыву между принципами проектирования программного обеспечения и реальной практикой программирования. Например, алгоритм с лучшей оценкой вычислительной сложности может работать дольше, чем алгоритм, который в большей степени учитывает особенности аппаратного обеспечения, но обладает худшей теоретической оценкой вычислительной сложности. Это противоречие может быть частично устранено благодаря применению аппаратных средств, реализующих более совершенные механизмы для поддержки операций над структурами данных, которые работают

параллельно и одновременно с существующей подсистемой памяти. Также важно, чтобы наличие таких средств не приводило к сокращению универсальности системы, но позволяло ускорить решение ряда задач, основанных на использовании ссылочных структур данных.

Исследования [12] показали необходимость и перспективы глубокого пересмотра основополагающих принципов представления информации в ЭВМ, в том числе форм представления структурированных и неструктурированных данных, способов обработки множеств, механизмов доступа к данным. В связи с этим исследования, направленные на определение новых парадигм обработки данных, в настоящее время чрезвычайно актуальны.

Безусловным преимуществом аппаратной реализации действий над множествами является возможность существенно более быстрой и параллельной обработки тех операций, которые требуется выполнять одновременно над несколькими элементами. В таком алгоритме сами множества или порождающие их процедуры должны быть записаны в память ЭВМ: на внешний носитель, в оперативную память, в постоянное запоминающее устройство или регистры процессора. Доступ к этой памяти может быть выполнен через независимые шины, обработка полученной из памяти информации может вестись независимо от других видов обработки, аппаратные устройства могут учитывать специфические форматы хранимых данных и выполнять основные алгоритмы с большей степенью параллельности. Таким образом, первый принцип определяет цели исследования: разработку новых архитектурных принципов и устройств аппаратной обработки множеств и структур данных.

Принцип 2 двойственности структур данных. Для возможности параллельной обработки структур данных в состав вычислительной системы вносится аппаратное устройство (процессор обработки структур), который обрабатывает лишь ту часть информации, которая определяет взаимные отношения хранимых данных, т. е. структурную часть структур данных. Универсальный центральный процессор обрабатывает информационную составляющую структур данных.

Отметим, что в вычислительных системах, не обладающих свойством гетерогенности, алгоритмы и скалярные данные, а также структуры данных и алгоритмы их обработки хранятся и обрабатываются одними и теми же универсальными микропроцессорами (рис. 1).



Рис. 1. Вычислительная система с аппаратной поддержкой операций над множествами и структурами данных

На основе гетерогенности вычислений выделяется часть аппаратного обеспечения для реализации операций дискретной математики. Принцип двойственности структур данных обосновывает состав и функциональное назначение частей такой системы.

Принцип опирается на понятие структуры данных, обладающей двойственностью:

$$S = (D, R), \quad (1)$$

где D — множество элементов данных; R — множество отношений между элементами данных. Таким образом, структура определяется как совокупность двух множеств: D и R . Их обработка во многом параллельна и происходит по различным алгоритмам: алгоритмы обработки отношений представляют собой действия по модификации или обходу структуры в соответствии с имеющейся структурной информацией. Например, при обработке древовидной структуры свойства самого дерева определяют алгоритм обхода, алгоритм добавления нового элемента и т. д. Алгоритмы же обработки информационной составляющей решают уже конкретную задачу оптимизации, используя структуры данных. Примером может служить реализация алгоритма Дейкстры для поиска кратчайшего пути на графе, т. е. конкретная задача оптимизации, для которой граф может быть представлен в виде списка смежных вершин. Классический вариант алгоритма оперирует такими понятиями, как множество вершин, множество ребер, инцидентность, и не оговариваются конкретные варианты структур данных, которые эти множества и операции реализуют. Поэтому алгоритм Дейкстры является алгоритмом обработки множества данных D (информационная составляющая), в то время как действия над списками определяют алгоритмы обработки отношений R структур данных.

Важной особенностью новой вычислительной системы является наличие специализированного процессора, выполняющего функции

обработки отношений данных R , но не обрабатывающего сами данные. Это устройство носит название процессора обработки структур данных (СП). Действиями над информационной составляющей занимается центральный процессор (ЦП), являющийся универсальным процессорным устройством, реализующим арифметическую и логическую обработку данных.

Таким образом, второй принцип использует понятия структуры данных как совокупности двух множеств и обосновывает независимую их обработку двумя различными устройствами: центральным процессором и процессором обработки структур.

Набор команд дискретной математики DISC (Discrete Instruction Set Computing)

Ключевым вопросом при проектировании любого программно-управляемого устройства является выбор набора команд. Так как целями создания системы является аппаратная поддержка аппарата дискретной математики, были рассмотрены теоретические аспекты реализации алгоритмов дискретной оптимизации и совокупность элементарных преобразований, которые необходимо проводить над множествами. На основе таких понятий, как кванторы, отношения и операции, был определен базовый набор операций системы, представленный в табл. 1.

Исходя из вышесказанного можно отметить, что СП выполняет инструкции высокого уровня, реализующие операции, кванторы и функции дискретной математики. Набор операций СП представляет собой такие действия, как: добавление элемента, поиск элемента, удаление, поиск минимального или максимального элемента, поиск следующего, мощность множества ключей, объединение, дополнение, пересечение [12]. Дополнительно реализованы операции срезов и команды синхронизации потоков команд ЦП и СП. Последняя версия набора команд Leonhard x64 состоит из 20 высокоуровневых кодов операций, перечисленных ниже:

- **Search (SRCH)** выполняет поиск значения, связанного с ключом;
- **Insert (INS)** вставляет пару ключ-значение в структуру. SPU обновляет значение, если указанный ключ уже находится в структуре;

Таблица 1

**Соответствие инструкций DISC функциям,
кванторам и операциям дискретной математики**

Функции, кванторы и операции дискретной математики	Инструкции набора команд DISC
Функция хранения кортежа	INS
Функция отношения элементов множества	NEXT, PREV, NSM, NGR, MIN, MAX
Мощность множества	CNT
Функция принадлежности элемента множеству	SRCH
Добавление элемента в множество	INS
Исключение элемента из множества	DEL, DELS
Исключение подмножества из кортежа	DELS
Включение подмножества в кортеж	LS, GR, LSEQ, GREQ
Отношение эквивалентности множеств	LS, GR, LSEQ, GREQ
Объединение множеств	OR
Пересечение множеств	AND
Разность множеств	NOT
Симметрическая разность множеств	Не реализовано
Декартово произведение множеств	Не реализовано
Булеан множества	Не реализовано

- операция **Delete (DEL)** выполняет поиск указанного ключа и удаляет его из структуры данных;
- последняя версия набора команд Leonhard x64 была расширена двумя новыми инструкциями (NSM и NGR) для обеспечения требований некоторых алгоритмов. Каждая инструкция набора включает до трех операндов: ключа, значения и номера структуры данных. Команды **NSM/NGR** выполняют поиск соседнего ключа, который меньше (или больше) заданного и возвращает его значение. Операции могут быть использованы для эвристических вычислений, где интерполяция данных используется вместо точных вычислений (например, кластеризация или агрегация);
- **Maximum/minimum (MAX, MIN)** ищут первый или последний ключи в структуре данных;
- операция **Cardinality (CNT)** определяет число ключей, хранящихся в структуре;

- команды **AND, OR, NOT** выполняют объединения, пересечения и дополнения в двух структурах данных;
- **срезы (LS, GR, LSEQ, GREQ)** извлекают подмножество одной структуры данных в другую;
- **переход к следующему или переход к предыдущему (NEXT, PREV)** находят соседний (следующий или предыдущий) ключ в структуре данных относительно переданного ключа. В связи с тем что исходный ключ должен обязательно присутствовать в структуре данных, операции NEXT/PREV отличаются от NSM/NGR;
- **удаление структуры (DELS)** очищает все ресурсы, используемые заданной структурой;
- команда **Squeeze (SQ)** дефрагментирует блоки памяти DSM, используемые структурой;
- команда **Jump (JT)** указывает SPU код ветвления, который должен быть синхронизирован с CPU (команда доступна только в режиме MISD).

Приведем пример того, как алгоритм дискретной оптимизации может быть реализован в гетерогенной вычислительной системе с различными типами микропроцессоров. Указанный выше алгоритм Дейкстры [7, 8] позволяет найти кратчайший путь от начальной вершины графа до всех остальных вершин и использует две структуры данных: структуру для представления графа, а также очередь для хранения упорядоченной последовательности вершин. Чтобы понять, какой микропроцессор должен выполнять действия алгоритма в гетерогенной системе, примем следующие обозначения в псевдокоде:

- (D) — это действие основано на операциях дискретной математики и должно выполняться на микропроцессоре Leonhard x64;
- (M) — это операция чтения/записи в память (или регистр), которая может быть инициирована любым процессором в системе;
- (A) — это арифметическая операция, которая может выполняться арифметическим микропроцессором.

Далее приведем псевдокод алгоритма Дейкстры. Пусть граф задан с помощью структуры **Graph**, в котором для каждой вершины **v** указывается поле длины пути **dist** и номер предшествующей вершины графа в кратчайшем пути **previous**. Тогда поиск кратчайшего пути от вершины **source** до каждой вершины графа **Graph** может быть выполнен следующим образом:

Архитектура вычислительной системы с набором команд DISC

```

FOR ALL вершина  $v$  из  $Graph^{(D)}$  DO
   $dist[v] \leftarrow \infty^{(M)}$ 
   $previous[v] \leftarrow undefined^{(M)}$ 
END FOR
 $dist[source] \leftarrow 0^{(M)}$ 
 $Q \leftarrow$  множество вершин из  $Graph^{(D)}$ 
WHILE  $Q^{(D)}$  DO
   $u \leftarrow$  вершина из  $Q$  с минимальным  $dist[]^{(D)}$ 
  Удалить  $u$  из  $Q^{(D)}$ 
  FOR ALL вершина  $v$ , смежная  $u$  DO  $(D)$ 
     $alt \leftarrow dist[u] + ребро(u, v)^{(A)}$ 
    IF  $alt < dist[v]^{(A)}$  THEN
       $dist[v] \leftarrow alt^{(M)}$ 
       $previous[v] \leftarrow u^{(M)}$ 
    END IF
  END FOR
END WHILE
  
```

Как мы видим, наиболее трудоемкие и длительные по времени операции алгоритма связаны с выполнением операций дискретной математики (доступу и изменению множеств), а не с арифметической обработкой чисел. Следовательно, совершенствование аппаратного обеспечения для выполнения этих частей рабочей нагрузки способно значительно повысить общую производительность системы. Таким образом, мы используем некоторые новые аппаратные принципы, чтобы частично устранить ограничения, установленные выше:

- реализуем специально разработанный и высокопараллельный микропроцессор обработки структур данных (СП, версия Leonhard x64) для выполнения инструкций набора команд DISC. Этот микропроцессор не требует программного обеспечения для управления памятью, а доступ ко всем структурам данных был реализован исключительно аппаратно;
- используем чрезвычайно короткий конвейер в СП, состоящий из трех стадий, что значительно уменьшает задержку для операций с зависимыми данными. Такой конвейер неэффективен для общей арифметики, но хорошо подходит для обработки множеств;
- разделяем поток инструкций на два потока: арифметическую обработку и инструкции дискретной математики.

Далее рассмотрим более подробно архитектурные принципы построения системы.



Рис. 2. Поток команд и данных в системе с аппаратной обработкой множеств и структур данных

СП Leonhard x64 осуществляет хранение структур данных в оперативной памяти, обеспечивает их обработку и выдачу скалярных данных в центральный процессор для ее последующей обработки основным вычислительным алгоритмом. Оперативная память, доступная процессору обработки структур, делится на локальную память команд и локальную память структур данных (рис. 2). Вся необходимая для ЦП информация структур данных выдается из СП посредством выполнения команд обработки структур.

На основе представленной на рис. 2 системы был сформулирован принцип структурной декомпозиции системы.

Принцип 3 структурной декомпозиции вычислительной системы для решения задач дискретной оптимизации. Процессор обработки структур имеет доступ к оперативной памяти, в которой хранятся структуры данных и команды. Результаты выполнения команд направляются в центральный процессор для дальнейшего использования в ходе основного вычислительного алгоритма. Способ представления структур данных в локальной памяти процессора обработки структур должен обеспечивать высокую производительность процессора для всего набора операций.

Рассматривая возможные варианты организации структур данных в оперативной памяти, необходимо учитывать аппаратную и временную сложность выполнения операций. В связи с тем, что заранее не известны те из них, которые будут наиболее существенными для конкретных алгоритмов, следует выбирать такой вариант внутреннего представления структур, который обеспечивает хорошие результаты для всего спектра операций. Наиболее пригодными как с точки зрения аппаратной сложности, так и по временным характеристикам были признаны сбалансированные деревья. Среди них следует выделить так называемые сильно ветвящиеся В-деревья и В⁺-деревья. Использование хэш-таблиц не обеспечивает приемлемой вычислительной сложности при определении порядка на множестве, т. е. не позволяет осуществить обход значений. Многочисленные практические примеры, такие как поиск по индексам в базах данных и доступ к информации на flash-накопителях и внешних носителях, показывают, что этот вид деревьев является перспективным для реализации как в программном, так и в аппаратном обеспечении.

Таким образом, третий принцип определяет основы взаимодействия центрального процессора и процессора обработки структур, а также поясняет состав и назначение оперативной памяти СП.

Принцип 4 множественности потоков команд. В вычислительной системе с аппаратной поддержкой операций над структурами данных один поток данных обрабатывается несколькими потоками команд.

Для подтверждения четвертого принципа следует более подробно рассмотреть управление вычислительным процессом и форматов представления данных в вычислительной системе с набором команд дискретной математики.

Из принципов 1–3 следует, что процессор обработки структур — это устройство, которое изменяет и анализирует лишь ту часть информации, которая определяет отношения хранимых данных [9, 12]. Основным форматом данных СП, показанным на рис. 3, является тройка чисел (структура, ключ, значение), где: поле СТРУКТУРА имеет три разряда; каждое из полей КЛЮЧ и ЗНАЧЕНИЕ имеют 64 разряда (справедливо для версий СП с микроархитектурой Leonhard x64). Используются также дополнительные форматы данных для обмена между ЦП и СП, однако они применяются для синхронизации вычислительных процессов и будут подробно рассмотрены в следующих статьях.

В качестве аппаратной формы представления множеств ключей и значений было выбрано двоичное дерево. В таком случае структурная часть содержит информацию о конфигурации дерева и взаимных отношениях всех ее элементов, а сами значения являются информационной частью структуры данных. В случае, когда необходимо выбрать элемент с заданным ключом, процессор обработки структур выполняет поиск и выборку элемента по структурной составляющей, после чего передает информационную составляющую (значение, ассоциированное с ключом) ЦП. Таким образом, действия, исполняемые двумя процессорами, могут выполняться параллельно и независимо, а общий поток команд обработки данных для такой ЭВМ разделяется на два потока: поток команд обработки информационной составляющей и поток обработки структурной составляющей структу-

ры данных. Это позволяет классифицировать предложенную вычислительную систему как систему со многими потоками команд и одним потоком данных.

Примером решения вычислительной задачи в такой системе может служить некоторая программа обработки данных, на некотором шаге которой требуется найти минимальное значение ключа и увеличить его в n раз. Тогда последовательность команд, выполняемых процессорами, будет следующей:

1. Процессор обработки структур выполняет команду поиска минимального элемента в структуре: (ключ, значение) = MIN (структура).

2. Пара значений (ключ, значение) передаются в ЦП и удаляется из структуры: DEL (структура, ключ, значение).

3. ЦП меняет поле ключа: $ключ = ключ \cdot n$.

4. ЦП передает новую пару значений в процессор обработки структур: ADD (структура, ключ, значение).

Первая команда приводит к выполнению алгоритма обхода структуры и сравнению ключей для поиска наименьшего. Ключ найденной вершины должен быть увеличен в n раз, что будет выполняться арифметико-логическим устройством (т. е. ЦП) и для внесения изменений может потребовать перестроение структуры процессором обработки структур. Очевидно, что действия по поиску вершины с минимальным ключом и последующее перестроение структуры должен выполнить СП, а за изменение значения ключа отвечает арифметически-логическое устройство ЦП.

В предложенной вычислительной системе в памяти процессора обработки структур содержится информация, характеризующая взаимное расположение элементов данных. Иными словами, это информация, указывающая на последовательность в соответствии с их числовыми значениями (структурная составляющая). Это позволяет процессору обработки структур начать выполнение очередной команды по поиску максимального ключа структуры сразу после завершения предыдущей операции удаления, не дожидаясь обработки ключа в АЛУ ЦП. Таким образом, процессор обработки структур выполняет повторяющиеся операции поиска и удаления максимального элемента в соответствии со своим потоком команд, а также выдает поток данных ЦП, который обрабатывает их вторым потоком команд, т. е. ЦП последовательно накапливает значение счетчика стоимости. Потоки команд независимы и определяются лишь характером



Рис. 3. Основной формат данных процессора обработки структур с микроархитектурой Leonhard x64

решаемой задачи. В результате оба процессора обрабатывают структуру данных одновременно и параллельно. Эта особенность подтверждает, что вычислительная система, соответствующая предложенным принципам, относится к классу "много потоков команд и один поток данных".

Для пояснения следующего принципа отметим, что классификация ЭВМ и вычислительных систем по числу потоков команд и данных была предложена М. Флинном. Согласно работе [17] наличие двух потоков команд и/или данных в вычислительной машине обязательно предполагает организацию архитектурного параллелизма вычислений, что существенно влияет на всю концепцию построения вычислительной машины. В работе [17] предлагаются четыре класса вычислительных машин: с одиночным потоком команд и одиночным потоком данных (ОКОД); с одиночным потоком команд и множественным потоком данных (ОКМД); со множественным потоком команд и одиночным потоком данных (МКОД); со множественным потоком команд и множественным потоком данных (МКМД).

Принцип 5 иерархической вложенности архитектур вычислительных систем. Принцип иерархической вложенности вычислительных систем различных классов предполагает, что в одной МКМД-системе может быть реализовано несколько МКОД-систем, в МКОД-системе может быть реализовано несколько ОКМД-систем, в одной ОКМД-системе может быть реализовано несколько ОКОД-вычислительных устройств.

Практические реализации вычислительных машин классов ОКОД, ОКМД и МКМД широко распространены: к классу ОКОД относятся все однопроцессорные ЭВМ; к классу ОКМД относятся устройства векторной и матричной обработки (SIMD SSE, 3DNow, Altivec и другие); к классу МКМД относятся многоядерные вычислительные системы. И только класс МКОД (рис. 4) остается невостребованным в настоящее время.

В вычислительной МКОД-системе взаимодействуют несколько устройств, которые ведут обработку единственного потока данных, причем возможна как последовательная обработка данных, передаваемых от устройства к устройству, так и обработка первоначального потока из локальной памяти всеми устройствами. Этот тип архитектуры соответствует описанной ранее архитектуре системы с аппаратной поддержкой операций дискретной математики, когда одно устройство выполняет обработку информационной составляющей, а второе об-

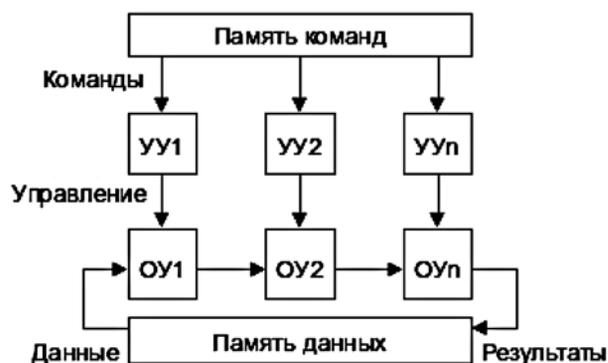


Рис. 4. Вычислительная система с многими потоками команд и одним потоком данных

рабатывает структурную составляющую. Вся структура данных хранится в локальной памяти и доступна одному или обоим процессорам. В настоящий момент не предложены и не реализованы в промышленных масштабах системы, обладающие такой архитектурой. К классу МКОД иногда относят конвейерные ЭВМ, однако большинство исследователей отрицают это, так как ступени конвейера не являются независимыми и обрабатывают все же один поток команд. К МКОД также относят так называемые систолические матрицы, которые реализовывались в проектах в 60–70-е годы прошлого века. Результаты этих проектов повлияли на развитие векторно-конвейерных процессоров, однако имели ряд функциональных ограничений. В связи с этим считается, что систем данного класса в настоящий момент не существует. Причиной этого явилось отсутствие четко сформулированных принципов построения таких систем, которые были бы востребованы в вычислительных оптимизационных задачах и при обработке данных. Так, в качестве возможных примеров в работе [17] и других источниках называется цифровая фильтрация несколькими устройствами одного сигнала, а также использование нескольких устройств криптографического анализа информации в целях ее декодирования. Указанные примеры являются узкоспециализированными задачами в ограниченных областях, в то время как обработка множеств и структур данных имеет широкое распространение.

Таким образом, до представления результатов проекта МГТУ им. Н. Э. Баумана не было предложено разумной и обоснованной концепции применения МКОД-парадигмы вычислений, востребованной при решении практических задач, не было получено ни одного примера реализации всех указанных четырех



Рис. 5. Принцип иерархической вложенности классов вычислительных систем при обработке структур данных и множеств

классов в виде единой вычислительной системы. Поэтому приведенный принцип иерархической вложенности, сформулированный на основе 13 лет кропотливой исследовательской и конструкторской работ по созданию вычислительной системы с набором команд дискретной математики, является важным достижением и сформулирован впервые.

Реализация четвертого класса вычислительных систем (много потоков команд и один поток данных), не реализованного до данного исследования, позволила определить иерархические отношения всех четырех классов (рис. 5):

- в МКОД-системе может быть реализовано несколько ОКМД (один поток команд и много потоков данных) вычислительных устройств, обрабатывающих параллельно элементы данных. Например, на одном уровне В+-дерева могут храниться несколько пар (ключ, значение). Все пары могут параллельно обрабатываться одним потоком команд в узлах ОКМД вычислителя;
- устройство с одним потоком команд и многими потоками данных состоит из нескольких примитивных устройств, обрабатывающих по одному потоку данных (ОКОД). Например, такой блок может обрабатывать один ключ в вершине. Таким образом, устройства с одним потоком команд и одним потоком данных при обработке структур данных могут быть иерархически вложены в устройства класса ОКМД;
- для параллельной обработки многих структур данных одновременно могут использоваться несколько взаимосвязанных МКОД-систем, что позволяет получить вычислительную систему с многими потоками команд и многими потоками данных (МКМД).

Таким образом, принцип иерархической вложенности определяет возможность включения структур ЭВМ одного класса в другие, что и было продемонстрировано в ходе проекта реализации вычислительной системы с набором команд дискретной математики.

Принцип 6 работы процессора обработки структур. Процессор обработки структур может работать в режиме МКОД, режиме сопроцессора и в комбинированном режиме.

ЦП выполняет функции основного управляющего устройства: загружает задачу из внешней памяти или сети, выделяет место в оперативной памяти ЦП для размещения исходных данных и результатов, выполняет инициализацию содержимого оперативной памяти, запускает задачу на исполнение, передает данные в СП для формирования структур данных или использует уже сформированные структуры, получает результаты обработки структур и использует их в арифметически-логической обработке, передает результаты во внешние устройства и освобождает память. Основной же вычислительный алгоритм разделяется на два потока команд: команды обработки данных, обрабатываемые ЦП, и команды обработки структур, обрабатываемые СП. Для функционирования системы процессоры должны получать следующую информацию (рис. 6):

- ключи, значения и номера структур (в случае хранения нескольких структур одновременно) передаются ЦП в качестве аргументов для выполнения команд в СП;
- результаты выполнения команд (ключи, значения, флаги) передаются из СП в ЦП;
- события синхронизации вычислительных потоков поступают из ЦП в СП и в обратном направлении для координации двух частей алгоритмов;
- команды управления поступают из ЦП в СП для настройки режимов работы и выполнения сервисных процедур;

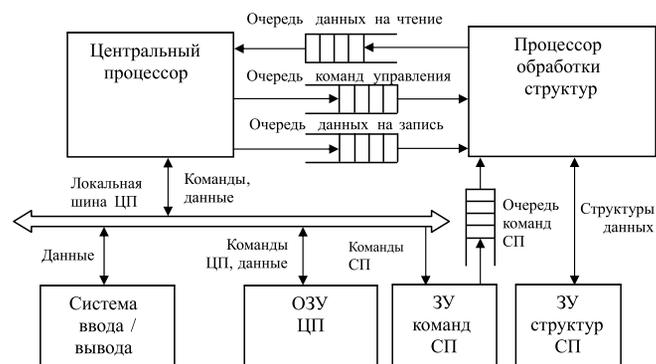


Рис. 6. Пример взаимодействия устройств в системе МКОД

- команды обработки структур данных поступают в СП из локальной памяти команд;
- команды обработки скалярных данных (значений) и сами значения поступают в ЦП из локальной памяти.

Для гетерогенной вычислительной системы, работающей под управлением ЦП, возможны несколько режимов работы других процессоров. Могут быть использованы следующие режимы:

- **режим сопроцессора** — управляющие команды для обработки множеств и структур данных формируются в ЦП системы и поступают через очередь управляющих команд;
- **режим МКОД** — команды выбираются процессором обработки структур из локального ОЗУ команд. СП использует независимую память для хранения структур, что обеспечивает параллельность и независимость арифметически-логической обработки ЦП и обработки множеств и структур данных в СП.
- **комбинированный режим** — команды поступают как от ЦП, так и из локального ОЗУ. Выбор очередной команды осуществляется в порядке их готовности в соответствии с приоритетами. Таким образом, память для хранения структур данных является единым ресурсом системы, предоставляемым условно в виде сервиса с помощью процессора обработки структур другим обрабатываемым устройствам.

Обеспечение работы системы в перечисленных режимах и передаче командной информации и данных можно организовать многими способами, используя шины передачи данных. Известны несколько вариантов топологии вычислительных машин [1], которые могут быть реализованы и для МКОД-систем. В зависимости от назначения, конструктивных особенностей и используемой элементной базы могут быть использованы такие топологии, как общая шина, независимые шины, иерархия шин.

Конкретный вариант топологии может быть выбран по различным критериям, таким как производительность; сложность аппаратной реализации; степень готовности технология разработки программного обеспечения; возможность масштабирования системы. Под масштабируемостью будем понимать технические возможности по

увеличению рабочей нагрузки системы посредством увеличения количественных характеристик (дублирования таких узлов, как ЦП и СП, памяти, числа шин и пр.).

Таким образом, шестой принцип определяющий способы командного управления вычислительной системой и взаимодействия ЦП и процессора обработки структур.

Производительность микропроцессора Leonhard x64

В проведенных экспериментах был использован микропроцессор Leonhard x64, реализованный на основе ПЛИС FPGA Virtex 6 со встроенным локальным микропроцессором Microblaze. Мы использовали третью версию Leonhard x64 со следующими параметрами: 64-битные ключи и значения; максимальное число ключей в памяти структур 100 млн (100 663 296 записей); максимальное число структур данных: 7; 4 Гбайт объем локальной памяти структур; рабочая частота Leonhard x64 100 МГц. В качестве основы альтернативной системы был выбран ЦП Intel Core i5-4300U CPU, 2 ядра, 4 потока, тактовая частота 1.90GHz, 8 Гбайт ОЗУ. Результаты экспериментов показаны в табл. 2.

Таблица 2

Экспериментальное исследование производительности Leonhard x64

Эксперимент	Число тактов на операцию		Архитектурное ускорение
	Intel Core i5-4300U CPU, 2 ядра, 1.90GHz, 8 Гбайт ОЗУ	Leonhard x64, 100 MHz, 4 Гбайт память структур	
Последовательная вставка 1М элементов во множество, <i>Average sequential insert rate, ASIR</i>	3201	741	4,3
Случайная вставка 1М элементов во множество, <i>Average random insert rate, ARIR</i>	3298	1425	2,3
Последовательный поиск элементов в 1М множестве, <i>Average sequential search rate, ASSR</i>	1903	255	7,5
Случайный поиск элементов в 1М множестве, <i>Average random search rate, ARSR</i>	3870	985	3,9
Последовательное удаление элементов из 1М множества, <i>Average sequential delete rate, ASDR</i>	2921	360	8,1
Случайное удаление элементов из 1М множества, <i>Average random delete rate, ASDR</i>	6485	943	6,9
Обход множества 1М элементов, <i>LSM 1M keys traversing time, LSMTT</i>	2017	262	7,7
Поиск ближайшего в 1М множестве, <i>Average neighbours search rate, ANSR</i>	3729	1338	2,8

Приведенные эксперименты позволяют комплексно оценить производительность систем при выполнении базовых операций над множествами: вставка, поиск, удаление, обход множеств. Эксперименты построены на основе выполнении действий по формированию множеств или его изменению. В качестве основы выбрано множество, состоящее из 1 млн элементов. Так, эксперимент по последовательной вставке элементов во множество (Average sequential insert rate, ASIR) предполагает создание множества из последовательно возрастающих элементов. Эксперимент по вставке в множество случайных значений отличается тем, что последовательность значений формируется случайным образом.

Как видно из табл. 2, во всех проведенных экспериментах было достигнуто архитектурное ускорение при выполнении базовых операций дискретной математики. Это доказывает целесообразность реализации архитектурных принципов, описанных в данной статье. При этом аппаратная сложность микропроцессора Leonhard x64, выраженная в количестве ресурсов кристалла, составляет около 1 млн вентилей. Это более чем в 1300 раз меньше, чем у микропроцессора Intel Core i5, участвующего в сравнении.

Однако в связи с 19-кратной разницей в тактовых частотах универсальная система на основе микропроцессора Intel Core i5 имеет от 2- до 5-кратного преимущества в сравнении с микропроцессором Leonhard x64 по времени выполнения тестов. Для устранения этого отставания в настоящее время разрабатывается новая версия микропроцессора Leonhard x64 с существенно большим быстродействием.

Выводы

Вычислительная система с набором команд дискретной математики, соответствующая архитектурным принципам класса МКОД, разработана и успешно реализована в МГТУ им. Н. Э. Баумана. Вычислительная система использует параллельную обработку структур данных на основе специализированного аппаратного блока — процессора обработки структур. Принципы функционирования МКОД-системы основаны на параллельной обработке двух взаимосвязанных частей структур данных: информационной и структурной.

В работе приведены первые шесть базовых принципов функционирования вычислитель-

ной системы с аппаратной поддержкой операций над множествами и структурами данных:

- принцип гетерогенности вычислений;
- принцип двойственности структур данных;
- принцип структурной декомпозиции вычислительной системы для решения задач дискретной оптимизации;
- принцип множественности потоков команд;
- принцип иерархической вложенности архитектур вычислительных систем;
- принцип работы процессора обработки структур.

В следующих работах будут продемонстрированы принципы хранения и аппаратной обработки информации оперативной памяти *процессора обработки структур*, принципы построения такого процессора, принципы функционирования программного обеспечения в МКОД-системе.

Список литературы

1. **Цилькер Б. Я., Орлов С. А.** Организация ЭВМ и систем. СПб.: Питер, 2011. С. 688.
2. **Kankowski P.** x86 Machine Code Statistics. 2009. URL: http://www.strchr.com/x86_machine_code_statistics (дата обр. 27.03.2018).
3. **Huang I. J., Peng T. C.** Analysis of x86 instruction set usage for DOS/Windows applications and its implication on super-scalar design // IEICE Transactions on Information and Systems. 2002. Vol. E85-D, N. 6. P. 929–939.
4. **Davidson A., Baxter S., Garland M., Owens J. D.** Work-Efficient Parallel GPU Methods for Single-Source Shortest Paths // 2014 IEEE 28th International Parallel and Distributed Processing Symposium. Phoenix, AZ. 2014. P. 349–359.
5. **Accelerating High-Performance Computing With FPGAs.** White Paper. Altera Corporation. WP-01029-1.1: tech. rep. October 2007. P. 8. URL: <https://www.intel.ru/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01029.pdf> (дата обр. 19.09.2019).
6. **Integrated Cryptographic and Compression Accelerators on Intel Architecture Platforms.** Intel Corporation: tech. rep. 2013. P. 5. URL: <https://www.intel.ru/content/dam/www/public/us/en/documents/solution-briefs/integrated-cryptographic-compression-accelerators-brief.pdf> (дата обр. 19.09.2019).
7. **Кнут Д.** Искусство программирования. Т. 3. Сортировка и поиск. М.: Вильямс, 2000. 832 с.
8. **Кормен Т., Лейзерсон Ч., Ривест Р.** Алгоритмы: построение и анализ. М.: МЦНМО, 2000. 960 с.
9. **Попов А. Ю.** Электронная вычислительная машина с многими потоками команд и одним потоком данных: патент 71016 Рос. Федерация. № 2006115810; заявл. 10.05.2006; опубл. 20.02.2008. Бюл. № 5. 1 с.
10. **Попов А. Ю.** Исследование вариантов реализации алгоритмов Крускала и Прима в вычислительной системе с многими потоками команд и одним потоком данных // Наука и образование. МГТУ им. Н. Э. Баумана. Электрон. журн. 2015. № 11. С. 505–527.
11. **Попов А. Ю., Подольский В. Э.** Методика декомпозиции информационного графа программы для организации параллельной обработки данных на ЭВМ МКОД // Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение. 2016. Т. 1, № 106. С. 112–128.

12. **Popov A.** An Introduction to the MISD Technology // HICSS50. Proceedings of the 50th Hawaii International Conference on System Sciences, 2017. P. 1003–1012.

13. **Foster C. C.** *Content Addressable Parallel Processors*. New York: Van Nostrand Reinhold Company, 1976. P. 169.

14. **Batcher K. E.** *STARAN parallel processor system hardware* // *National Computer Conference*. 1974. P. 405–410.

15. **Popov A., Rasheed B.** Network Graph Datastore Using DISC Processor // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).

Saint Petersburg and Moscow, Russia. 2019. P. 1582–1587. doi: 10.1109/EIConRus.2019.8656749.

16. **Popov A., Abdymanapov Ch.**, Motion Planning Algorithms Using DISC // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). Saint Petersburg and Moscow, Russia. 2019. P. 1844–1847. doi: 10.1109/EIConRus.2019.8657250.

17. **Flynn M. J.** Very High-Speed Computing Systems // Proceedings of the IEEE. 1966. Dec. Vol. 54, N. 12. P. 1901–1909.

A. Yu. Popov, Ph.D., Associate Professor of the Department "Computer Systems and Networks", Bauman Moscow State Technical University, Moscow, e-mail: alexpopov@bmstu.ru

Principles of the Organization of a Heterogeneous Computing System with a Set of Commands of Discrete Mathematics

The computing system with a discrete mathematics instruction set (DISC) is improving in the Bauman Moscow State Technical University. This system consists of microprocessors with different architectures which allow programs to call the hardware functions of large sets and data structures operations, graphs processing and other discrete mathematics functions useful for optimization process. The core of this system is a principally new computing device which called Structure processor Leonhard x64. In the previous works, the results of computer system design were shown, the features of programs execution and data processing were disclosed, the structure of the new processor was presented, and the results of experiments on measuring performance were presented. In this article, we summarize and systematize a number of key principles for constructing a system with a DISC command set, which primarily provide a high level of performance. We consistently state the reason for the introduction of each of the principles and the consequences of its implementation, and demonstrate some examples. The presented article has an important theoretical meaning for the entire project, since many of the results are formulated for the first time.

Keywords: data structure, computer system with a set of discrete mathematics commands; system with many streams commands and one stream of data; structure processing processor; $B + tree$

DOI: 10.17587/it.26.67-79

References

1. **Tsilker B. Ya., Orlov S. A.** Computer Organization and Systems: A Textbook for High Schools, SPb.: Peter, 2011, 688 p. (in Russian).

2. **Kankowski P.** x86 Machine Code Statistics, 2009, available at: http://www.strchr.com/x86_machine_code_statistics (date of access 03.27.2018).

3. **Huang J. J., Peng T. C.** *IE J. Trans.*, 2002, vol. E85-D, no. 6, pp. 929–939.

4. **Davidson A., Baxter S., Garland M., Owens J. D.** Work-Efficient Parallel GPU Methods for Single-Source Shortest Paths, *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, Phoenix, AZ, 2014, pp. 349–359.

5. **Accelerating High-Performance Computing With FPGAs**. White Paper. Altera Corporation. WP-01029-1.1: tech. rep. October 2007, pp. 8, available at: <https://www.intel.ru/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01029.pdf> (access: 19.09.2019).

6. **Integrated Cryptographic and Compression Accelerators on Intel Architecture Platforms**. Intel Corporation: tech. rep. 2013, pp. 5, available at: <https://www.intel.ru/content/dam/www/public/us/en/documents/solution-briefs/integrated-cryptographic-compression-accelerators-brief.pdf> (access: 19.09.2019).

7. **Knut D.** The Art of Programming, volume 3. Sorting and searching, Moscow, Williams, 2000, 832 p. (in Russian).

8. **Kormen T., Leyzerson Ch., Rivest R.** Algorithms: construction and analysis, Moscow, MTSNMO, 2000, 960 p. (in Russian).

9. **Popov A. Yu.** Electronic computer with many streams of commands and one data stream: patent 71016 Ros. Federation.

No. 2006115810; declare 05/10/2006; publ. February 20, 2008 Bul № 5. 1 p. (in Russian).

10. **Popov A. Yu.** Investigation of options for the implementation of Kruskal and Pym algorithms in a computer system with many streams of commands and one data stream, *Science and Education. MGTU them. N. E. Bauman. Electron. Journals*, 2015, no. 11, pp. 505–527 (in Russian).

11. **Popov A. Yu.** Method of decomposition of the information graph of the program for organizing parallel data processing on an MKOD computer / A. Yu. Popov, V. E. Podolsky // *Vestnik MGTU im. N. E. Bauman. Ser. Instrument making*. 2016. T. 1, No. 106. pp. 112–128 (in Russian).

12. **Popov A. Yu.** An Introduction to the MISD Technology / A. Popov // HICSS50. Proceedings of the 50th Hawaii International Conference on System Sciences, 2017. pp. 1003–1012 (in Russian).

13. **Foster C. C.** *Content Addressable Parallel Processors, New York, Van Nostrand Reinhold Company*, 1976, 169 p.

14. **Batcher K. E.** *STARAN parallel processor system hardware, National Computer Conference*, 1974, pp. 405–410.

15. **Popov A. Yu., Rasheed B.** Network Graph Datastore Using DISC Processor, St. Petersburg and Moscow, Russia, 2019, pp. 1582–1587. doi: 10.1109 / EIConRus.2019.8656749.

16. **Popov A. Yu., Abdymanapov Ch.** Motion Planning Algorithms Using DISC, St. Petersburg and Moscow, Russia, 2019, pp. 1844–1847. doi: 10.1109 / EIConRus.2019.8657250.

17. **Flynn M. J.** Very High-Speed Computing Systems, *Proceedings of the IEEE*, 1966, vol. 54, no. 12, pp. 1901–1909.

Э. М. Пинт, канд. техн. наук, проф.,

И. Н. Петровнина, канд. техн. наук, доц., e-mail: irisha-vas@yandex.ru,

И. И. Романенко, канд. техн. наук, доц., e-mail: rom1959@yandex.ru,

К. А. Еличев, канд. техн. наук, доц., e-mail: yelichevk@mail.ru

Пензенский государственный университет архитектуры и строительства

Разработка метода распознавания печатных символов (букв, цифр) для программы персонального компьютера

Авторы работы разработали новый способ определения типов символов печатных текстов различных шрифтов. Предлагаемое устройство состоит из системы восприятия и персонального компьютера (ПК). Было установлено, что важные и необходимые признаки для различения типов символов печатных текстов находятся в найденной комбинации направлений, образуемых частями контура символа. По разработанному алгоритму выделяются указанные направления путем окантовки, начиная с концевой точки, изображения формы символа, находящегося в запоминающем устройстве ПК, по которым однозначно определяются типы символов печатных текстов разных шрифтов.

Ключевые слова: печатный символ, окантовка, программа, алгоритм, эталонный код, персональный компьютер, основное направление, код символа, упрощение кода символа, определение кода символа

Введение

До настоящего времени еще не решена задача определения типов символов печатных текстов, шрифты которых значительно отличаются по форме. Эту задачу должно выполнять устройство, во-первых, воспринимающее печатные тексты с источника, во-вторых, определяющее типы символов печатных текстов разных шрифтов.

Рассматриваемое устройство может использоваться для автоматической обработки опытных данных, для упрощения нахождения нужной информации в библиотеках, для анализа сведений, имеющих на промышленных и других предприятиях.

Для ввода конфигурации печатных символов в матрицу персонального компьютера (ПК) применялось специально созданное устройство, но возможно также использование широко применяемых сканеров, применение которых не влияет на расшифровку печатных символов с различным начертанием.

При реализации второй функции рассматриваемого устройства — определения типов символов печатных текстов — встречаются определен-

ные трудности, главной из которых является то, что устройство должно определять типы символов различных шрифтов, т. е. уметь определять одну и ту же букву или цифру, но напечатанную с другой высотой, шириной, с другим наклоном, более жирно, более узко и т. д.

Существующие способы определения типов символов печатных текстов можно разделить на группы:

- 1) способы наложения или шаблонов;
- 2) способы с использованием специальных отметок или стилизованной формы символов;
- 3) способы определения символов по элементам формы;
- 4) способы с использованием фрагментов зондов;
- 5) интегральные способы;
- 6) способы с использованием обучаемых систем.

Способы наложения базируются на сопоставлении печатного символа с определенными трафаретами, шаблонами. Авторы исследований [1] предлагают изображение печатной цифры с помощью оптической системы проецировать на диск, в котором вырезаны трафареты цифр. За диском установлен фотоэле-

мент. Когда фотоэлемент полностью затемняется, срабатывает реле, и цифра распознается.

По второму способу символы отличаются друг относительно друга по какому-либо специальному кодированию изображений символов. Была спроектирована читающая машина для распознавания цифр с использованием написания цифр в виде вертикальных полосок [2].

Третий способ в качестве существенных признаков использует особенности геометрической формы символов печатного текста: число прямых и кривых отрезков, сегментов, расположение отрезков, наличие выпуклостей и впадин и т. д. (рис. 1). Для распознавания печатных знаков используется число пересечений знака горизонтальными линиями, расположенными на разных уровнях по высоте, и учитывается наличие кривых отрезков и концевых точек [3]. По предлагаемому алгоритму проводились работы, направленные на совершенствования метода считывания различных печатных образов [7–12].

Способы определения символов с помощью фрагментов и зондов в качестве существенных признаков используют наличие или отсутствие черного поля изображения символа в данном фиксированном месте поля. Английская компания разработала читающую машину, в которой для распознавания цифр применялся метод деления печатного символа на пять фрагментов, наличие или отсутствие в которых черного поля кодировалось соответственно "1" или "0" [4].

Интегральные способы в качестве существенных признаков принимают интегральные характеристики геометрической формы символов, например, распределение плотности изображения по исследуемому полю (рис. 2). Разработанное устройство для распознавания печатных знаков использовало проекции печатных знаков на вертикальную и горизонтальную оси [5].

Для способов определения типов символов с использованием обучаемых систем изображения определяемых символов с помощью поля,

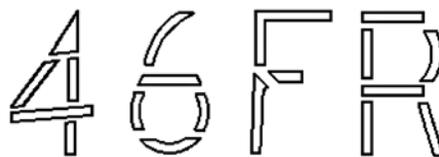


Рис. 1. Вид элементов фигуры, используемых для различения символов по элементам формы



Рис. 2. Закон изменения проекции площади фигуры символа при проектировании ее на горизонтальную ось

состоящего из фотоэлементов, вписывались в запоминающее устройство ЭВМ. Алгоритм обучения ЭВМ состоял из четырех частей:

- 1) проведение плоскостей, разделяющих один тип символа относительно другого типа для предъявленных машине символов (рис. 3, а–в);
- 2) выбрасывание лишних плоскостей (рис. 3, г);
- 3) выбрасывание лишних кусков плоскостей (рис. 3, з);
- 4) определение типов новых символов.

Исследователи как за рубежом, так и в России, пытались создать машинный интеллект с возможностью выполнения операций по постоянному самообучению [6].

Анализ проведенных исследований показывает, что все перечисленные выше способы определения типов печатных символов в настоящее время не используются (кроме третьего способа), так как они позволяют определять ограниченное число типов символов, в лучшем случае в пределах одного шрифта.

Третий способ позволяет определять типы печатных символов при определенных вариациях их формы, но числу определяемых типов

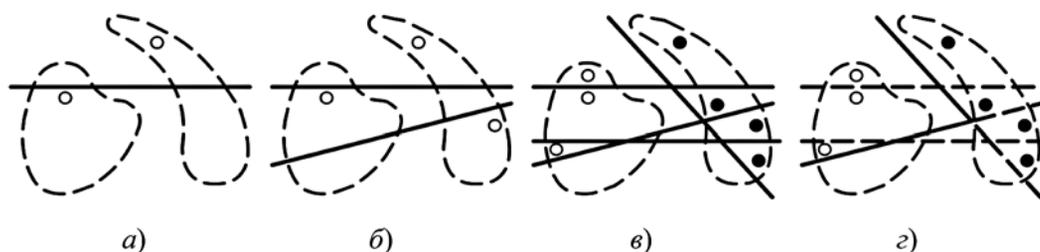


Рис. 3. Пояснения алгоритма обучения ЭВМ

символов оказывается ограниченным и замыкается в основном в пределах одного шрифта.

Предлагаемое в данной работе устройство позволяет определять типы печатных символов (буквы, цифры) разных шрифтов независимо от изменений в определенных пределах размеров символов, положения элементов формы символов, наличия помех и т. д.

1. Постановка задачи

Первоочередной задачей являлось создание для ЭВМ методики распознавания печатных символов независимо от вида шрифта, разной толщины частей изображения, разного масштаба и т. д.

2. Метод решения задачи

Проведенный авторами работы анализ формы символов печатных текстов дал возможность в качестве важных и необходимых признаков символов использовать определенную комбинацию направлений, в которую входят направления, образуемые частями контура символа в порядке его окантовки. В используемой комбинации направления отмечаются определенными номерами и идут по кругу под углом $2\pi/8$ друг относительно друга (рис. 4).

Воспринятое с помощью системы восприятия разработанного устройства изображение символа с источника информации передается в запоминающее устройство персонального компьютера, где заполненные ячейки соответствуют элементарным участкам переданного изображения.

По разработанной программе осуществляется окантовка формы изображения символа, начиная с конечной точки. Каждый символ печатного текста представляется в виде кода, состоящего из последовательности номеров основных направлений, входящих в контур символа и в найденную комбинацию направлений. Основное направление определяется числом

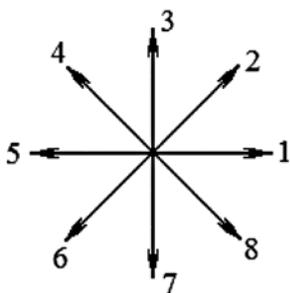


Рис. 4. Найденная комбинация направлений

заполненных ячеек запоминающего устройства ПК, которое должно быть больше или равно числу заполненных ячеек, составляющих максимальную толщину линий и других частей изображения символа (так называемый вес). Неосновные направления, образуемые толщиной элементов символа, не фиксировались в коде. В коде фиксировались те неосновные направления, которые возникали при переходе от одного элемента символа к другому.

Представление символа в виде кода, состоящего в основном из номеров основных направлений, позволило однозначно определять тип каждого символа относительно других типов, независимо от масштаба элементов символа, от разных шрифтов, т. е. выбранная комбинация направлений являлась оптимальной.

Окантовка формы изображения символа осуществлялась следующим образом. Начиная с конечной заполненной ячейки, имеющей наименьший номер адреса, опрашивались соседние заполненные ячейки в целях нахождения основного направления. После его определения следовало перемещение на соседнюю заполненную ячейку в этом направлении и т. д., пока не была осуществлена окантовка изображения. Номер основного направления при перемещении от ячейки к ячейке фиксировался в запоминающем устройстве.

Встречались заполненные ячейки, когда выявлялись сразу два основных направления, т. е. наблюдалась развилка. Тогда адрес этой ячейки запоминался, и следовал переход на соседнюю ячейку по основному направлению с наименьшим номером. Если по этому направлению заканчивалась окантовка, то она продолжалась, начиная с ячейки, где была развилка по другому основному направлению, и т. д. до тех пор, пока не заканчивалась окантовка от всех зафиксированных развилочек.

Были составлены эталонные коды типов печатных символов. Эталонный код был образован путем окантовки контура каждого идеально напечатанного типа символа с учетом особенностей различных шрифтов [13].

Каждому типу печатного символа соответствовал свой эталонный код. Эталонный код состоял из последовательности номеров основных направлений (неосновные рассмотрены выше), входящих в контур символа.

По разработанной программе составлялся упрощенный код каждого символа, в котором из изображения символа убирались заполненные ячейки, составляющие лишнюю толщину элементов символа, а также убирались следу-

ющие направления, возникающие от возможных помех изображения, от ненужных украшений формы символа для некоторых шрифтов, от возможных изменений наклонов элементов изображения символа [14, 15].

Упрощенный код каждого символа по разработанной методике сравнивался поочередно со всеми эталонными кодами печатных символов для определения типа символа [14].

3. Результаты исследования

Для экспериментов на разработанном устройстве использовали два шрифта с отличающимися по форме символами: символы одного шрифта имели дополнительные украшения — "засечки", символы другого шрифта их не имели (рис. 5).

При этом, как выяснилось, формы контуров символов выбранных шрифтов повторялись в основном во всех остальных шрифтах русского алфавита для одинаковых по смыслу символов.

В результате проводимых экспериментов выяснялось, какие K направлений контуров формы символов должны быть выбраны для определения наибольшего числа типов символов печатных текстов. При этом при нахождении комбинации направлений важно было также учитывать, какие символы определяются, так как буквы имеют различную повторяемость в текстах P_i .

Для оценки определения устройством печатных символов двух анализируемых шрифтов был введен следующий параметр:

$$L = C_k \sum_{i=1}^{C_k} \frac{P_i}{2},$$

где C_k — число определяемых букв для k направлений; P_i — вероятность появления букв в текстах.

Максимальное значение величины L было получено для комбинации направлений, изображенной на рис. 4. При использовании этой комбинации все типы печатных символов рассматриваемых шрифтов, представленные в виде кода направлений, образуемого после окантовки формы символа, различались друг относительно друга независимо от размеров элементов символа, от используемых шрифтов.

Поясним, как осуществлялось во время окантовки формы символа перемещение от одной заполненной ячейки на соседнюю заполненную ячейку в любом основном направлении

УФХЦЩЪЫЬЭЮЯ
уфхцщъыьэюя
890
а)

ЙКЛМНОПРСТУФХЦ
йкклмнопрстуфхц
1234567890
б)

Рис. 5. Компьютерные шрифты:

а — буквы и цифры без засечек; б — буквы и цифры с засечками

из комбинации направлений, изображенных на рис. 4. Пусть заполненная ячейка изображения символа имеет адрес 001001 (рис. 6).

Для перемещения персональным компьютером в любом основном направлении из выбранной комбинации направлений нужно к адресу заполненной ячейки памяти ПК (рис. 6) добавлять следующие числа:

- для 1 направления (+) 001000,
- для 2 направления (+) 001001,
- для 3 направления (+) 000001,
- для 4 направления (−) 000111,
- для 5 направления (−) 000001,
- для 6 направления (−) 001001,
- для 7 направления (−) 000001,
- для 8 направления (+) 000111.

000100	001100	010100	100100
000010	001010	010010	100010
000001 ← 5	001001	010001 → 1	100001
6 ↙	001000	010000	100000

Diagram showing movement directions from the central cell 001001:

- 1: Right (→)
- 2: Up-right (↗)
- 3: Up (↑)
- 4: Up-left (↖)
- 5: Left (←)
- 6: Down-left (↘)
- 7: Down (↓)
- 8: Down-right (↙)

Рис. 6. Передвижение по любому из выбранной комбинации направлений из заполненной ячейки на соседние

Например, надо переместиться на соседнюю ячейку в направлении 2. Тогда персональный компьютер прибавляет к адресу 001001 число 001001 и получается число 010010, т. е. адрес соседней ячейки по направлению 2 (рис. 6) и т. д.

Образование кода символа в виде последовательности направлений, образуемых при окантовке контура символа, осуществляется таким образом, что для одного направления в определенную ячейку заносится номер направления и соответствующее этому направлению число номеров и т. д. для других направлений. Это позволяет оперировать с единичными номерами направлений, т. е. с масштабно-преобразованным символом, и учитывать число номеров одного направления. При сравнении упрощенного кода символа с любым эталонным кодом подсчитывается возможное число несовпадений, и по окончании сравнения со всеми эталонными кодами по меньшему числу несовпадений определяется символ. Сравнение упрощенного кода символа с эталонным кодом может либо не дать несовпадения, либо при определенных вариациях формы символа (разные шрифты, дефекты и т. д.) дать малое число несовпадений по сравнению с числом несовпадений, получаемым от сравнения упрощенного кода символа с другими эталонными кодами.

Сравнение с эталонным кодом и определение символа идет согласно выбранной методике следующим образом. Первый номер направления из эталонного кода сравнивается по порядку с первым номером упрощенного кода символа. Если номера совпадают, то они стираются из эталонного кода и из упрощенного кода символа. Если совпадения нет, то номер направления из эталонного кода сравнивается со следующим по порядку номером направления упрощенного кода символа и т. д. до совпадения номеров.

После совпадения номеров следует очередной номер направления из эталонного кода, который сравнивается с оставшимися номерами упрощенного кода символа, следующими за только что стертым номером. Если после n -го сравнения совпадения нигде не произошло, стирания нет, то при $(n + 1)$ -м сравнении очередной номер из эталонного кода сравнивается с номерами направлений упрощенного кода символа, следующими за стертым номером после $(n - 1)$ -го сравнения, или, если и ранее не было со-

впадений, сравнение начинается с первого номера направления упрощенного кода символа. Сравнение номеров заканчивается либо когда все номера эталонного кода прошли цикл сравнения, либо когда за стертым номером упрощенного кода символа не следует очередного номера направления. Число несовпадений подсчитывается путем сложения оставшихся номеров в эталонном коде и в упрощенном коде символа, причем номер направления из упрощенного кода символа суммируется в общее число несовпадений только в том случае, если этому направлению соответствует сумма номеров, равная или больше веса, т. е. направления является основным.

Например (рис. 7, а), запись буквы "П" по направлениям будет следующая:

1 333 333 332 1111 777 777 77 8 11

Таблица 1

Номера направлений	1	3	2	1	7	8	1
Суммы номеров	1	8	1	4	8	1	2

Запись буквы "П" (рис. 7, б) другого шрифта и значительно меньших размеров (например, одна буква заглавная, другая строчная) будет:

3333 2 11 77777

Таблица 2

Номера направлений	3	2	1	7
Суммы номеров	4	1	2	5

В табл. 1 и 2 входят номера направлений и сумма номеров по каждому направлению.

Вес равен трем ячейкам матрицы или двум номерам направления (для простоты выбрали буквы с толщиной, равной одной ячейке

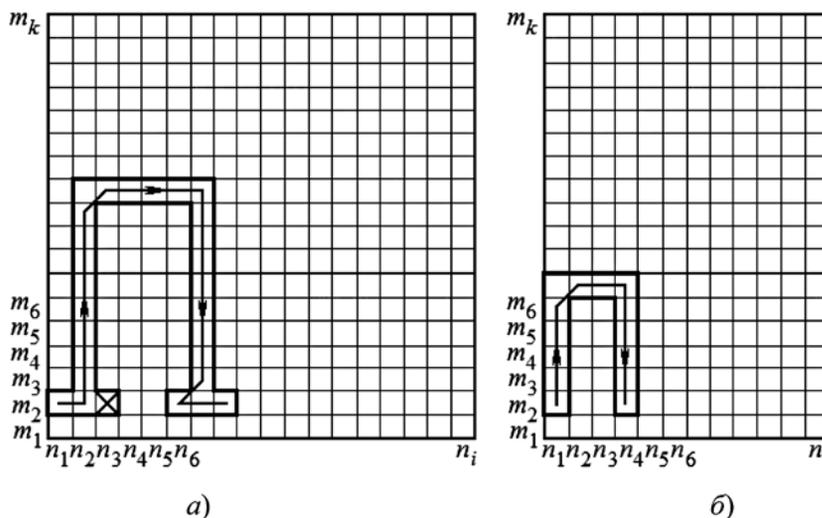


Рис. 7. Обход по контуру буквы "П" разных шрифтов

матрицы). Эталонный код может выглядеть как "317" или как "3217". Допустим, что выбрали эталонный код "3217". После сравнения этого эталонного кода с кодом первой буквы (рис. 7, а), согласно изложенному выше правилу, в эталонном коде не остаются номера, а код буквы после стирания совпавших номеров будет выглядеть: "181". В этом коде первые два направления "1" и "8" имеют суммы меньше веса, поэтому эти направления не дадут совпадений. Для последнего направления "1" сумма равна весу (т. е. направление основное), получаем всего одно несовпадение, и буква "П" распознается. Вторая буква "П" не дает несовпадений. Буквы "П" разной формы и масштаба распознаются одинаково (можно проследить, что сравнение этих букв с эталонными кодами других символов даст гораздо большее число несовпадений).

Отметим, что, с одной стороны, сумма по направлению "2" в эталонном коде "3217" ("3, 1, 7" — отражают основные направления) является дополнительным полезным характерным признаком буквы "П", с другой стороны, суммы по направлениям "1" и "8" отражают декоративные украшения, поэтому не входят в эталонный код буквы "П", не дают несовпадений и на распознавании не сказываются.

Рассмотрим определение сходных по форме букв "В" и "З" рассматриваемых шрифтов (рис. 8, а, б и рис. 9, а, б). На указанных рисунках даются изображения в памяти ПК букв "В" и "З" рассматриваемых шрифтов. На этих рисунках приведены изображения в памяти персонального компьютера букв "В" и "З" рассматриваемых шрифтов, где сплошной линией показана первая окантовка формы букв, а штриховой линией — вторая окантовка, начинающаяся с ячейки, где на-

блюдается развилка. Зачеркнутые ячейки иллюстрируют то, что заполнение в них исключается при ликвидации лишней толщины частей букв.

Таблица 3 показывает первоначальные коды букв "В" и "З" и коды, полученные после упро-

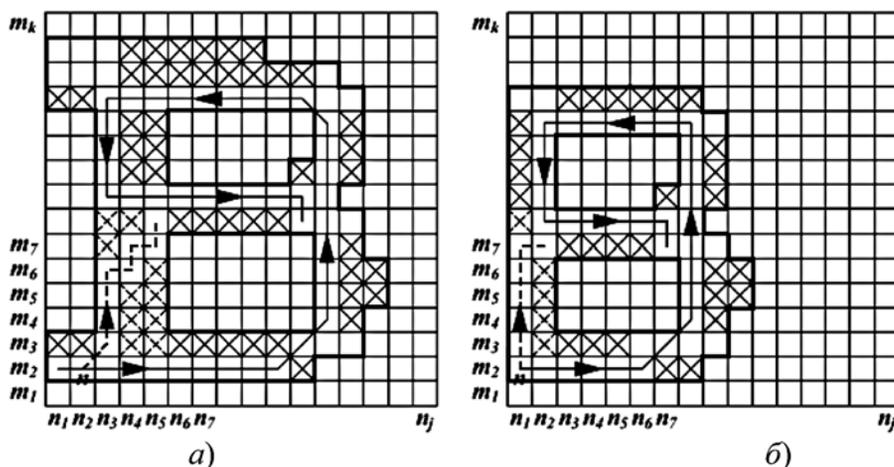


Рис. 8. Изображения букв "В":

а — для шрифта с засечками; б — для шрифта без засечек

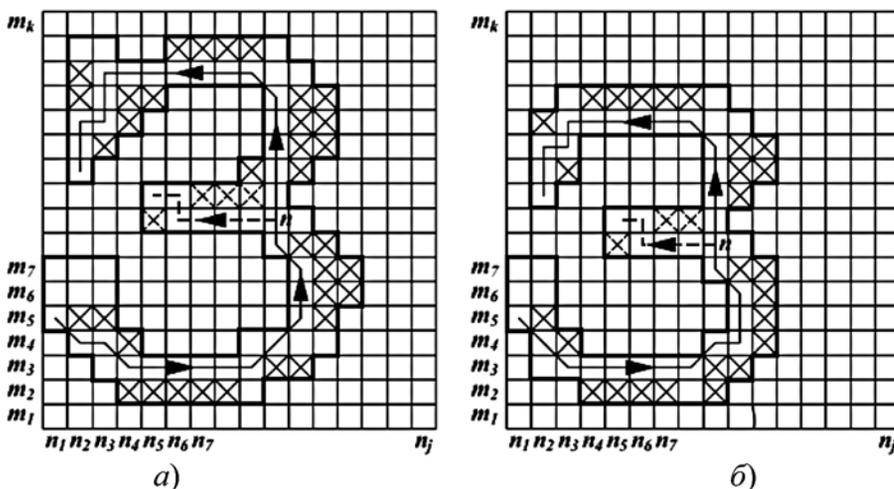


Рис. 9. Изображения букв "З":

а — для шрифта с засечками; б — для шрифта без засечек

Таблица 3

В рис. 9, а	Первоначальная запись по направлению	111111112233333334555555577771111111723331313
	Упрощенная запись по направлению	1111111122333333345555555777711111117233333
В рис. 9, б	Первоначальная запись по направлению	1111122333333335555557777111117333331
	Упрощенная запись по направлению	1111122333333335555557777111117333331
З рис. 10, а	Первоначальная запись по направлению	818111112233433333455555777555535
	Упрощенная запись по направлению	8811111223333333455555775555
З рис. 10, б	Первоначальная запись по направлению	881111121334333334555577755535
	Упрощенная запись по направлению	88111111333333345555577555

шения первоначальных кодов. Черта указывает на запись кода после развилки.

Для определения какой-либо буквы, например "В", относительно других по разработанной программе номера упрощенного кода буквы "В" сравниваются с номерами эталонных кодов всех букв. При этом при сравнении упрощенного кода буквы "В" с каким-либо эталонным кодом подсчитывается число несовпадений номеров в этом эталонном коде и в упрощенном коде буквы "В".

При сравнении, например, упрощенного кода буквы "В" со своим эталонным кодом число несовпадений номеров будет равно нулю, либо минимальным по сравнению с числами несовпадений, полученными после сравнения упрощенного кода буквы "В" с другими эталонными кодами. Таким образом, тип буквы относительно других типов букв определяется по нулевому или минимальному числу несовпадений, получаемому после сравнения упрощенного кода определенной буквы со своим эталонным кодом.

В таблице 4 представлены эталонные коды букв "В" и "З" и число несовпадений от сравнения упрощенных записей букв по направлениям с приведенными стандартными видами.

Как нетрудно заметить из табл. 4, сходные по форме буквы "В" и "З" в достаточной степени распознаются друг относительно друга; в то же время одноименные буквы разных шрифтов распознаются одинаково.

Таблица 4

Эталонные коды букв	Число несовпадений				
	В		З		
	рис. 9, а	рис. 9, б	рис. 10, а	рис. 10, б	
В	1 2 3 5 7 1 3	0	0	3	4
З	2 1 3 4 5 7 5	4	5	0	0

Заключение

Было создано устройство для определения типов символов печатных текстов разных шрифтов.

В результате проведенных исследований были выбраны важные и достаточные признаки символов рассматриваемых шрифтов, позволившие устройству надежно [13] (в отличие от других устройств) по разработанной программе определять типы символов печатных текстов разных шрифтов, несмотря на то, что в изо-

бражениях символов возможны определенные помехи, дополнительные украшения, нехарактерные отклонения, разные толщины и размеры частей формы символов. Разработанное устройство позволит автоматизировать процесс обработки печатной информации и образов на различных объектах промышленности, на транспорте, в банках, компаниях и т. д.

Список литературы

1. **Сочивка В. П.** Электронные опознающие устройства. М.: Энергия, 1964. 56 с.
2. **David H.** Direct Reading for Data Processing // *Process Control and Automation*. 1960. Vol. 7, N. 3. P. 3—30.
3. **Seema A., Rajeshwar D.** Pattern Recognition in an Electronic Reader // *International Journal of Computer Science and Telecommunications*. August 2012. Vol. 3, Iss. 8. P. 25—29.
4. **Nielsen W. F.** System Analysis Using Digital Computers // *Electronic industries*. August, 1960. Vol. 19, N. 8. P. 212—214.
5. **Романов В. П.** Интегральные методы опознавания. Конференция по обработке информации, машинному переводу и автоматическому чтению текста. Акад. наук СССР. Ин-т науч. информации. М., 1961. 23 с.
6. **Ковалевский В. А.** Корреляционный метод распознавания изображений // *Журнал вычислительной математики и математической физики*. 1962. № 4 (2). С. 684—694.
7. **Pradeep J., Srinivasan E., Himavathi S.** Diagonal Feature Extraction Based Handwritten Character // *International Journal of Computer Applications* (0975—8887). Vol. 8, N. 9. P. 17—21.
8. **Pradeep J., Srinivasan E., Himavathi S.** Diagonal based feature Extraction for handwritten alphabets recognition system using neural network // *International Journal of Computer Science & Information Technology (IJCSIT)*. 2011. Vol. 3, N. 1. P. 27—37.
9. **Sharma N., Patnaik T., Kumar B.** Recognition for Handwritten English Letters // *International Journal of Engineering and Innovative Technology: A Review*. 2013. Vol. 2, Iss. 7. P. 318—321.
10. **Singh V., Kumar B., Patnaik T.** Feature Extraction Technique for handwritten Text in Various Script // *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, March 2013, Vol. 3. P. 238—241.
11. **Woodford C.** Last updated: 30.12.2018, URL: <http://www.explainthatstuff.com/how-ocr-works.html>.
12. **Woodford C.** Last updated: 11.11.2018. URL: <http://www.explainthatstuff.com/how-ocr-works.html>.
13. **Пинт Э. М., Яшин А. В., Еличев К. А.** Оценка надежности распознавания компьютером печатных знаков // *Материалы V международной научной конференции "Актуальные проблемы современной науки-2009"*. Премиус: Наука 1 студия, 2009. С. 103.
14. **Пинт Э. М., Петровнина И. Н., Романенко И. И., Еличев К. А.** Полный алгоритм рационального метода распознавания компьютером печатных знаков разных шрифтов и других символов // *Вестник. БГТУ им. В. Г. Шухова*. 2013. № 1. С. 210.
15. **Пинт Э. М., Петровнина И. Н., Романенко И. И., Еличев К. А.** Меры для распознавания компьютером дорожных и печатных знаков разных шрифтов с определенными дефектами изображений знаков // *Модели, системы, сети в экономике, технике, природе и обществе*. 2016. № 1 (17). С. 308—318.

E. M. Pint, Cand. tech. sciences, Professor,
I. N. Petrovnina, Cand. tech. Sc., Associate Professor, e-mail: irisha-vas@yandex.ru,
I. I. Romanenko, Cand. tech. Sc., Associate Professor, e-mail: rom1959@yandex.ru,
K. A. Yelichev, Cand. tech. Sc., Associate Professor, e-mail: yelichevk@mail.ru,
Penza State University of Architecture and Construction, Penza, 440028, Russian Federation

Development of the Principal Symbol Recognition Method (Letters, Digital) for the Personal Computer Program

The authors of the work developed a new method for determining the types of characters of printed texts of various fonts. The proposed device consists of a perception system and a personal computer (PC). It was found that important and necessary signs for distinguishing the types of characters in printed texts are in the found combination of directions formed by the parts of the outline of the symbol. According to the developed algorithm, the indicated directions are distinguished by edging, starting from the end point, the image of the shape of the character in the PC storage device, by which the types of characters of printed texts of different fonts are clearly identified.

Keywords: printed symbol, edging, program, algorithm, reference code, personal computer, mainstream, character code, simplification of character code, definition of character code

DOI: 10.17587/it.26.80-87

References

1. **Sochivka V. P.** Electronic Identification Devices, Moscow, Energy, 1964, 56 p. (in Russian).
2. **David H.** Direct Reading for Data Processing, *Process Control and Automation*, 1960, vol. 7, no. 3, pp. 3–30.
3. **Seema A., Rajeshwar D.** Pattern Recognition in an Electronic Reader, *International Journal of Computer Science and Telecommunications*, August 2012, vol. 3, iss. 8, pp. 25–29.
4. **Nielsen W. F.** System Analysis Using Digital Computers, *Electronic Industries*, August 1960, vol. 19, no. 8, pp. 212–214.
5. **Romanov V. P.** Integral recognition methods, *Conference on information processing, machine translation and automatic reading of text*, Acad. sciences of the USSR. Institute of Science information, Moscow, 1961, pp. 23. (in Russian).
6. **Kovalevsky V. A.** Correlation method of image recognition, *Journal of Computational Mathematics and Mathematical Physics*, 1962, no. 4 (2), pp. 684–694 (in Russian).
7. **Pradeep J., Srinivasan E., Himavathi S.** Diagonal Feature Extraction Based Handwritten Character, *International Journal of Computer Applications* (0975–8887), vol. 8, no. 9, pp. 17–21.
8. **Pradeep J., Srinivasan E., Himavathi S.** Diagonal based feature Extraction for handwritten alphabets recognition system using neural network, *International Journal of Computer Science & Information Technology (IJCSIT)*, 2011, vol. 3, no. 1, pp. 27–37.
9. **Sharma N., Patnaik T., Kumar B.** Recognition for Handwritten English Letters, *International Journal of Engineering and Innovative Technology: A Review*, 2013, vol. 2, iss. 7, pp. 318–321.
10. **Singh V., Kumar B., Patnaik T.** Feature Extraction Technique for handwritten Text in Various Script, *International Journal of Soft Computing and Engineering (IJSCCE)*, March 2013, vol. 3, pp. 238–241.
11. **Woodford C.** Last updated: 30.12.2018, available at: <http://www.explainthatstuff.com/how-ocr-works.html>.
12. **Woodford C.** Last updated: 11.11.2018, available at: <http://www.explainthatstuff.com/how-ocr-works.html>.
13. **Pint E. M., Yashin A. V., Elichev K. A.** Assessment of the reliability of computer recognition of printed characters // Materials of the V international scientific conference "Actual problems of modern science-2009". Premium /: Science 1 studio, 2009, P. 103 (in Russian).
14. **Pint E. M., Petrovnina I. N., Romanenko I. I., Elichev K. A.** The full algorithm of the rational method of computer recognition of printed characters of different fonts and other characters, *Nauchno-teoreticheskij zhurnal Vestnik. BGTU im. V. G. Shuhova*, 2013, no. 1, pp. 210 (in Russian).
15. **Pint E. M., Petrovnina I. N., Romanenko I. I., Elichev K. A.** Measures for the recognition by the computer of road and printed signs of different fonts with certain defects of character images, *Nauchno-informacionnyj zhurnal Modeli, sistemy, seti v ekonomike, tekhnike, prirode i obshchestve*, 2016, no. 1 (17), pp. 308–318 (in Russian).

Н. Н. Корнева, вед. программист, nkorneva@romance.iki.rssi.ru,
В. Н. Назаров, нач. отдела, vnazarov@romance.iki.rssi.ru,
М. М. Могилевский, канд. физ.-мат. наук, зав. лабораторией, mogilevsky2012@gmail.com,
Институт космических исследований РАН (ИКИ РАН), г. Москва

Архитектура инструмента для визуального анализа спутниковых измерений в поисковых задачах

Рассматривается архитектура инструмента, автоматизирующего процесс визуального анализа временных рядов спутниковых измерений. Основное внимание уделено проектированию и описанию архитектуры в условиях недостаточной формализации физической постановки задачи, а также постоянно эволюционирующих нужд конечных пользователей (исследователей). Для описания архитектуры используются понятия и подходы из международного стандарта ISO/IEC/IEEE 42010:2011 Systems and software engineering — Architecture description (в Российской Федерации введен в действие в 2017 г. как ГОСТ Р 57100—2016 Системная и программная инженерия. Описание архитектуры). Рассматривается также архитектура инструмента в разрезе возможности одновременного функционирования инструмента, созданного на ее основе, в двух различных окружающих средах: в среде без доступа к наземному научному комплексу ИКИ РАН и в среде с доступом к нему.

Ключевые слова: архитектура программного обеспечения, ISO/IEC/IEEE 42010:2011, ГОСТ Р 57100—2016, точка зрения, представление, окружающая среда, визуальный анализ спутниковых измерений, поисковая задача

1. Постановка задачи

В 2004 г. французский Национальный центр космических исследований (CNES) осуществил запуск космического аппарата (КА) DEMETER. В качестве одного из научных приборов на КА был установлен прибор ICE (Instrument Champ Electrique), обеспечивающий, в частности, измерения трех электрических компонент в диапазоне крайне низких частот (КНЧ) 10 Гц ... 1,25 кГц [1]. Исследователю (ученому) необходимо выявить возмущения КНЧ-сигналов, предположительно обусловленные воздействием на ионосферу коротковолнового передатчика HAARP (эксперименты такого рода рассматривались в работах [2, 3] и др.).

Выявление (поиск) и изучение возмущений нестационарных сигналов, предположительно связанных с теми или иными событиями, — распространенная и часто возникающая задача при интерпретации результатов научного космического эксперимента. Характерной особенностью данной задачи является отсутствие строгой формализации критериев поиска, а следовательно, требуемый анализ не может быть выполнен полностью в автоматическом режиме. Роль исследователя в этом случае заключается не только в интерпретации измерений, но и в решении целого ряда задач, напрямую не связанных с работой ученого. К наиболее трудоемким из них относятся: выбор и изуче-

ние инструмента анализа данных, а также подготовка исходных данных. При этом нередко для анализа одного и того же набора данных требуется использовать различные инструменты, так как не один из них не предоставляет в полной мере требуемой функциональности. Как следствие, одни и те же наборы данных и промежуточные результаты приходится хранить в различных форматах.

В связи со всем вышперечисленным сформулируем задачу в более общем виде: даны измерения трех ортогональных компонент векторного поля на некотором временном интервале, необходимо разработать инструмент (программный продукт), автоматизирующий процесс анализа измерений на предмет выявления возмущений сигнала, предположительно связанных с теми или иными событиями. Инструмент должен позволять интеграцию в состав наземного научного комплекса (ННК) ИКИ РАН.

2. Анализ требований и способов решения

В работе [4] для исследования существенно нестационарного, многомерного (трехмерного), ограниченного во времени сигнала на предмет выявления возмущений, предположительно связанных с теми или иным событиями, было предложено использовать визуальный анализ. В данной работе понятие визуального анали-



Рис. 1. Метод визуального анализа

за расширено до циклического процесса, который включает в себя не только графическое представление данных и интерпретацию полученных представлений исследователем, но и этап автоматического анализа данных с дальнейшим построением на основе его результатов визуализаций (рис. 1). Аналитические методы при этом могут быть выработаны на базе графических представлений.

Обратим внимание на тот факт, что в силу недостаточной формализации физической постановки задачи, способа выявления возмущений сигналов, а также постоянно эволюционирующих нужд конечных пользователей (исследователей) основополагающим фактором, влияющим на реализацию инструмента, является деятельность ученого, направленная на исследование временных рядов спутниковых измерений. В данной работе авторы хотят акцентировать внимание на проектировании архитектуры инструмента в разрезе данного фактора.

Проанализируем требования к каждой из составляющих процесса визуального анализа с точки зрения вышесказанного.

Данные. Формат входных данных специфицирован частично. Известно, что входные данные содержат измерения трех ортогональных компонент векторного поля в пространстве и параметры, которые позволяют ассоциировать каждое из измерений с меткой времени (время начало регистрации, частоту дискретизации). Следовательно, дальнейшее использование данных требует их предварительной обработки в соответствии с динамической спецификацией. Поскольку специфика работы исследователя такова, что одни и те же результаты предобработки неоднократно используются на протяжении длительного времени, то предлагается их сохранять в некотором внутреннем формате, оптимизированном для дальнейшего использования модулями программного комплекса.

Визуализация. Согласно постановке задачи основная цель анализа измерений состоит

в выявлении возмущений сигнала, обусловленных каким-либо событием. Таким образом, графическое представление измерений должно позволять наблюдать изменение формы сигнала во времени и пространстве. Наиболее полно, с точки зрения ученых, этой цели исследования отвечает интерактивное графическое представление результатов эксперимента в виде набора трехмерных векторов в прямоугольной системе координат (прежде всего, в силу изначальной трехмерной структуры сигнала). Поскольку изменение формы сигнала необходимо наблюдать в динамике, то предложенная модель должна быть не только интерактивна, но и должна позволять произвольную навигацию по графическим моделям.

Кроме того, исходя из анализа временных рядов измерений, регистрируемых прибором ICE на КА DEMETER, можно предположить, что векторы на изображении будут довольно тесно располагаться по отношению друг к другу, что затруднит их визуальное распознавание. Качество распознавания предлагается повысить за счет разнесения отображаемых векторов по глубине или, другими словами, за счет создания у пользователя¹ стереоэффекта. Для стереовизуализации измерений предлагается использовать эклипсный метод, реализованный на базе решения NVIDIA 3D Vision для видеокарт пользовательского сегмента (обоснование выбора приведено в работе [4]).

Автоматический анализ. На данный момент времени не представляется возможным четко зафиксировать набор алгоритмов обработки и анализа измерений. Следовательно, программный комплекс должен предоставить:

- инструментарий пользователю для разработки новых алгоритмов;
- возможность добавления разработанных пользователем алгоритмов к уже существующему множеству;
- возможность комбинации пользователем уже существующих алгоритмов при проведении анализа измерений.

В рамках данного программного комплекса исходя из целей анализа (выявление и изучение временных особенностей сигналов) предлагается разбить все множество алгоритмов на две группы:

- поисковые алгоритмы;
- фильтры обработки.

¹ Здесь и далее по тексту пользователь и конечный пользователь употребляются как синонимы.

Поисковые алгоритмы должны обеспечивать возможность поиска участка на массиве измерений согласно заданному критерию поиска. Разработка нового алгоритма данной группы заключается в формировании нового типа критерия поиска. Предназначение фильтров обработки состоит в выделении из данных только тех, которые нужны исследователю, причем в отличие от поисковых алгоритмов фильтры модифицируют сигнал. В качестве примера фильтра обработки можно привести алгоритм расчета оценки математического ожидания амплитуды сигнала для исследуемых временных интервалов.

Этап автоматического анализа тесно связан с этапом визуализации в том смысле, что пользователю должна быть предоставлена возможность визуализации результатов автоматического анализа. Формы визуализации определяются типом полученных результатов и прорабатываются на этапе добавления нового алгоритма в систему. Для тех алгоритмов, результатом работы которых являются трехмерные векторы, должна быть предусмотрена возможность трехмерной визуализации результатов с предоставлением всего набора средств для работы с этой визуализацией (стереовизуализация, интерактивность, навигация). Для визуализации результатов поисковых алгоритмов также предлагается табличная форма.

Общий анализ исследования сигналов методом визуального анализа показывает, что результаты работы этапа автоматического анализа, а также результаты интерпретации графических моделей исследователем по отношению к исходным данным могут представлять как самостоятельную ценность, так и ценность в разрезе текущей задачи, решаемой исследователем. Следовательно, пользователю должна быть предоставлена возможность сохранения результатов, что требует разработки форматов хранения.

Кроме требований, обусловленных самой сущностью процесса визуального анализа, существенным является наличие возможности интеграции инструмента в ННК ИКИ РАН. Реализация этого требования должна подразумевать простоту переключения из режима автономной работы в режим, когда инструмент "подключен" к ННК для конечного пользователя. Такое условие обусловлено тем, что часть результатов научных космических экспериментов на данный момент размещена исключительно в архивах зарубежных космических агентств.

3. Архитектура программного комплекса

Согласно международному стандарту [5] (в Российской Федерации введен в действие в 2017 г. как [6]) для выражения архитектуры с позиции определенных интересов используются представления. Представление соответствует определенной точке зрения, устанавливающей соглашения для построения, интерпретации и использования данного представления. Для выражения архитектуры инструмента с позиции деятельности ученого по исследованию сигнала методом визуального анализа зададим точку зрения "Рабочий процесс". Формально данная точка зрения характеризуется следующими свойствами:

а) заинтересованные стороны: конечные пользователи, разработчики;

б) интересы: рабочие процессы системы, где рабочий процесс — это серия задач для получения результата, имеющего ценность для конечного пользователя;

в) тип модели: диаграмма деятельности UML (язык моделирования UML).

Представление, соответствующее точке зрения "Рабочий процесс", приведено на рис. 2. Для его создания на основе анализа требований, рассмотренных в п. 2, была проведена декомпозиция деятельности исследователя на рабочие процессы. Выделение набора задач в рабочий процесс осуществлялось таким образом, чтобы, с одной стороны, этот процесс реализовывал логически законченный с точки зрения ученого набор задач по исследованию сигнала, а с другой стороны, мог бы выполняться самостоятельно или же параллельно с другими типами процессов. Такая декомпозиция позволит упростить адаптацию системы к возможным изменениям в процессе анализа.

Как видно из диаграммы, в результате декомпозиции было выделено четыре типа рабочих процессов (на диаграмме каждому из них присвоен номер). Процессы первого типа (номер 1 на диаграмме) являются вспомогательными, тогда как все остальные типы реализуют набор действий, необходимый для исследования сигнала методом визуального анализа. Рассмотрим каждый из типов более подробно.

3.1. Рабочий процесс № 1

Схема работы процесса состоит из нескольких последовательных действий. Начинается работа с подключения к источнику данных. В контексте данной архитектуры источник данных — это

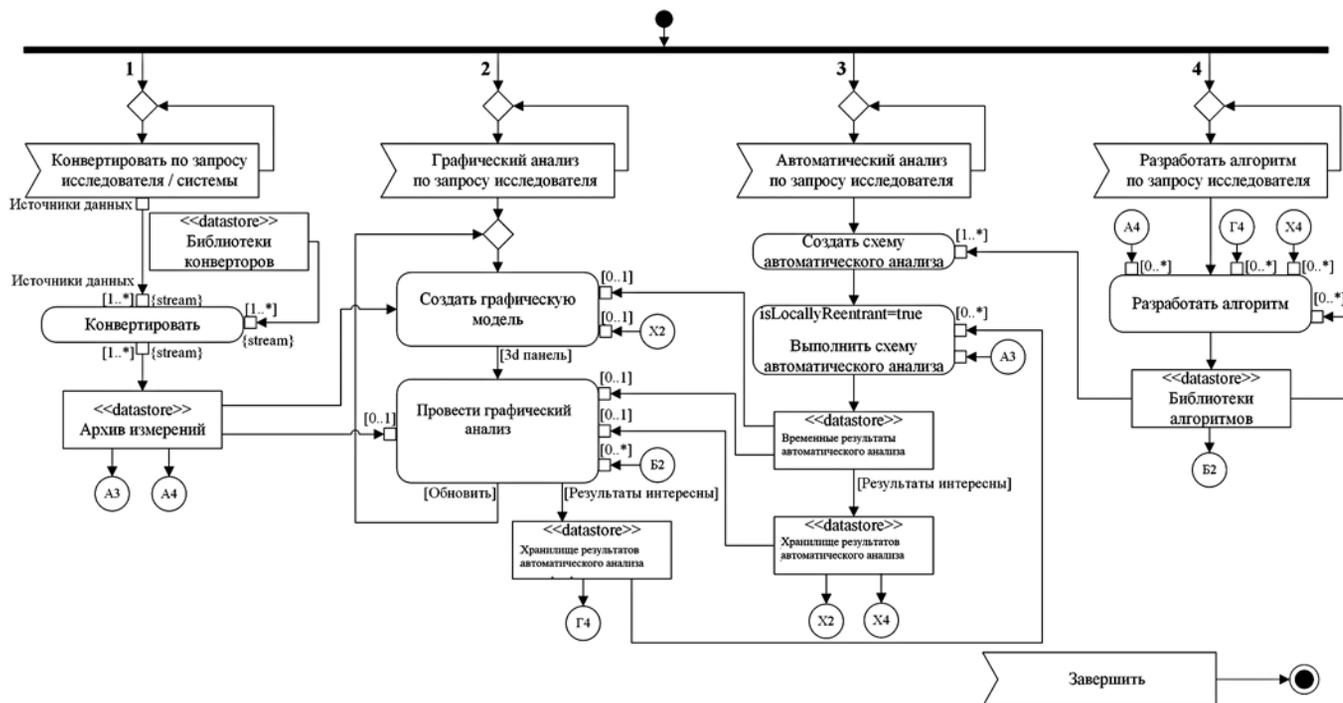


Рис. 2. Представление (точка зрения "Рабочий процесс")

набор данных, их формат (опционально) и сведения, необходимые для доступа к ним. Затем на основании формата данных (задается вручную, если не специфицирован в источнике) выбирается конвертор. Результат работы конвертора — это набор данных во внутреннем формате. Внутренний формат представляет собой контейнер, состоящий из компонентов. Например, бинарный компонент для экспериментальных данных содержит массив векторов вида $[x; y; z]$, где x, y, z — перекодированные (в случае необходимости) измерения из источника данных. Структура контейнера определена стандартом ECMA-376 OpenXML [7], при этом для конечного пользователя такой контейнер выглядит как один файл. Сохранение полученного контейнера в архив измерений — заключительная операция рассматриваемого рабочего процесса.

Вызов процессов данного типа может быть осуществлен как пользователем, так и системой при подключении к источнику данных. А расширение поведения процесса в смысле добавления нового формата входных данных реализуется за счет подключения к системе новых конверторов.

3.2. Рабочий процесс № 2

Рабочий процесс второго типа автоматизирует одну из центральных функций исследования сигнала методом визуального анализа:

создание интерактивных визуализаций данных и интерпретацию полученных графических моделей исследователем по отношению к исходным данным. Общая схема работы процесса заключается в циклическом выполнении двух этапов: создания графической модели некоего набора данных и их анализа пользователем. Рассмотрим каждый из них.

Этап 1. Создание графической модели. Данный этап инициируется пользователем и заключается в преобразовании данных к графической модели. Выбор способа преобразования реализуется за счет выбора соответствующей панели визуализации. Под панелью визуализации в данной архитектуре понимается не только само графическое представление данных, но и набор функций для управления визуализацией. Организация работы этапа посредством панелей предоставляет ряд преимуществ: во-первых, в случае необходимости позволяет расширить функциональность программного комплекса за счет добавления новых панелей; во-вторых, дает возможность визуализировать одни и те же данные различными способами.

Исходя из анализа требований и типов данных предусматривается несколько типов панелей визуализации:

- 3D-панель;
- панель состояний;
- информационная панель;
- панель результатов поиска.

3D-панель используется для визуализации временного ряда измерений из архива измерений и результатов работы поиска (см. подробнее п. 3.3) в виде набора трехмерных векторов. Изображение трехмерного объекта на экране выполняется методом центрального проецирования. Для работы с визуализацией в панели предусмотрена возможность ее масштабирования в нескольких режимах (масштаб; подогнать под экран; подогнать под модель) и вращения. Также 3D-панель позволяет организовать просмотр визуализации в полноэкранном режиме и стереорежиме. Панель состояний предназначена для вывода кратких сведений о временном ряде, визуализируемом в 3D-панели. К таким сведениям относятся временной интервал в формате с/по и его продолжительность. Панель позволяет пользователю самостоятельно определять формат вывода сведений (всемирное время или номер отсчета). Информационная панель отображает сведения о массиве анализируемых данных (продолжительность, частота дискретизации, сведения о примененных методах автоматического анализа и др.). Панель результатов поиска используется для представления результатов поиска в табличной форме.

Этап 2. Графический анализ. Данный этап выполняется полностью в ручном режиме пользователем и заключается в интерпретации визуализаций из 3D-панели, полученных на этапе 1. Для снижения трудоемкости графического анализа системой предоставляется два типа инструментов: инструменты для выборки измерений и фильтры отображения. К инструментам первой группы относятся инструменты навигации (позиционирование по времени / номеру отсчета; сдвиг с шагом по временному ряду измерений; движение по результатам поиска) и инструмент изменения размера выборки. Инструменты данного типа инициируют новый цикл рабочего процесса, или, иными словами, можно сказать, что результатом работы инструмента является визуализация в 3D-панели, построенная на новом наборе данных. Инструменты второй группы (фильтры отображения) влияют только на вид визуализации в 3D-панели, они также инициируют новый цикл рабочего процесса, однако результатом работы фильтра отображения является новая визуализация, построенная на том же наборе данных. Архитектура программного комплекса предполагает возможность комбинации пользователем инструментов друг с другом как из одной, так и из разных групп.

В случае необходимости исследователем может быть принято решение о сохранении результатов графического анализа. Операция сохранения не прерывает ни одного из типов рабочих процессов, т. е. после инициирования сохранения пользователь сможет продолжить работу в системе в обычном режиме. Форматы хранения результатов графического анализа зависят от предполагаемого способа использования данных результатов в дальнейшем. Так, если результаты предполагается использовать в другом программном обеспечении, то их можно сохранить в исходном формате (формате, специфицированном в источнике данных). Для тех случаев, когда результаты графического анализа будут использоваться в рамках данного программного комплекса, они сохраняются во внутреннем формате с добавлением новых компонентов (например, так сохраняется шаблон, используемый в дальнейшем для поиска по шаблону). Одни и те же результаты графического анализа могут быть сохранены пользователем в разных форматах. Добавление новых форматов хранения позволит расширить смысловую нагрузку результатов графического анализа (на данный момент под результатами понимается какой-либо участок сигнала, представляющий ценность для исследователя).

3.3. Рабочий процесс № 3

Рабочие процессы данного типа инициируются пользователем, но выполняются полностью в автоматическом режиме. Общая схема работы процесса заключается в автоматической обработке временного ряда измерения из архива измерений способом, определенным исследователем в схеме автоматического анализа. Под схемой автоматического анализа понимается набор алгоритмов из библиотеки алгоритмов и порядок применения этих алгоритмов к некоторому набору данных. Результатом выполнения схемы автоматического анализа является новый набор данных, который в случае необходимости может быть визуализирован. В архитектуре программного комплекса концептуально заложена эта возможность: добавление новой формы представления реализуется за счет добавления новой панели визуализации. Кроме того, полученный набор данных по запросу пользователя может быть сохранен для дальнейшего использования. Операция сохранения, как и в случае с сохранением результатов графического анализа, не прерывает ни одного из типов рабочих процессов.

Расширение данного процесса реализуется за счет добавления новых алгоритмов в библиотеки алгоритмов.

Важным (в разрезе поставленной задачи) частным случаем рабочего процесса третьего типа является рабочий процесс, в котором схема автоматического анализа состоит только из поисковых алгоритмов. Назначение такого процесса заключается в поиске участка на массиве измерений согласно заданным критериям поиска. Результатом работы процесса является временной ряд измерений, получаемый из исходного ряда путем исключения участков, не отвечающих критериям поиска. Поскольку структура выходного ряда не отличается от входного, то в системе предусмотрена возможность визуализации результатов поиска в виде набора трехмерных векторов посредством 3D-панели. Как следствие, у исследователя появляется возможность графического анализа результатов поиска. Кроме того, выход процесса можно интерпретировать как набор участков, тогда результаты можно представить в табличной форме, где каждая строка — это описатель найденного участка (время начала и конца). Формат сохранения результатов поиска подразумевает сохранение только параметров найденных участков (без самих отсчетов) и ссылок на расположение временного ряда измерений. Выбранный формат позволяет избавиться от избыточности данных (отсчеты хранятся только в одном месте).

3.4. Рабочий процесс № 4

Основу работы рабочего процесса третьего типа, а также графического анализа (см. п. 3.2, этап 2) составляют алгоритмы. При этом, как следует из анализа требований, на данный момент времени не представляется возможным четко зафиксировать требуемый набор алгоритмов. Рабочий процесс четвертого типа призван обеспечить пользователя функционалом для разработки новых алгоритмов. Однако создание подобного функционала с нуля видится довольно проблематичным, в первую очередь, из-за того, что в силу неопределенности задачи невозможно предсказать, какой математический аппарат может понадобиться разработчику алгоритмов. На данном этапе для этой цели прорабатывается вариант использования специализированных математических пакетов (например, MATLAB) или же связок типа интерпретируемый язык программирования плюс редактор плюс библиотека для научных

расчетов (например, Python + IPython + SciPy). Тогда рабочий процесс четвертого типа должен будет обеспечить пользователя всеми необходимыми данными, а также предоставить возможность использования уже имеющихся в системе алгоритмов. На выходе рабочего процесса должен быть получен готовый алгоритм, который затем будет интегрирован в систему (желательно с минимальным привлечением ручного труда).

4. Программный комплекс как часть ННК ИКИ РАН

Основу работы современного ННК составляет центр хранения и обработки данных. В ИКИ РАН под таким центром (далее Центр данных) понимают информационную систему, состоящую из средств хранения, обработки и анализа тематической информации, а также средств тактического и стратегического планирования космических исследований. Концепция архитектуры Центра данных, предложенная руководителем отдела Наземных научных комплексов ИКИ РАН Назаровым В. Н., подразумевает иерархическую структуру, состоящую из трех основных уровней:

- технического — к данному уровню относятся технические средства хранения и обработки данных, а также средства связи;
- функционального — средства информационной поддержки научных космических операций, прежде всего хранилище данных и средства обработки данных;
- экспертного — специалисты в области космических операций, фундаментальных и прикладных исследований, поддерживающие качество информации (полноту, достоверность, ценность информации и др.).

Тогда согласно данной модели программный продукт относится к средствам обработки данных функционального уровня, а в соответствии с терминологией [5] Центр данных является окружающей средой инструмента.

Рассмотренная в п. 3 архитектура программного комплекса позволяет инструменту, разработанному на ее основе, работать в двух различных окружающих средах: в среде без доступа к Центру данных и в среде с доступом к нему. При этом в архитектуре уже реализовано требование наличия возможности переключения между автономным режимом и режимом работы в составе Центра данных для конечного пользователя. Реализуется это за счет того, что

некоторые хранилища данных и библиотеки (см. рис. 2) рассматриваются как внутренние средства инструмента в автономном режиме и как внешние — в противном случае. В последнем варианте все внешние средства относятся к средствам функционального уровня. На данном этапе проработки архитектуры Центра данных к внешним средствам предлагается отнести: Хранилище результатов графического анализа, Хранилище результатов автоматического анализа и Библиотеки алгоритмов. Источники данных являются внешними для инструмента в обоих режимах.

Для моделирования потоков данных между инструментом и средствами функционального уровня Центра данных воспользуемся диаграммой потоков данных. Данная модель позволяет идентифицировать элементы окружающей среды (внешние сущности), используемые данные и потоки этих данных между системой и внешними сущностями. Все эти аспекты важны для исследователя в том смысле, что позволяют понять, к каким ресурсам Центра данных система, а соответственно, и сам пользователь может получить доступ. Диаграмма потоков данных приведена на рис. 3. Рассмотрим ее внешние сущности и способы взаимодействия инструмента с ними.

Хранилища данных. Занимают центральное место среди средств функционального уровня. Можно выделить два вида взаимодействий инструмента с ними:

- 1) по типу "потребитель", когда система запрашивает данные из хранилищ;
- 2) по типу "поставщик", когда программный комплекс записывает данные в хранилища.

Во взаимодействиях первого типа, как видно из диаграммы, могут участвовать: источники данных, хранилище результатов графического анализа и хранилище результатов автоматического анализа. Организация взаимодействия зависит от форматов запрашиваемых данных и в общем случае требует согласование форматов обмена, однако в частном случае рассогласование может быть устранено за счет добавления

новых форматов в инструмент. Взаимодействие с источниками данных осуществляется через архив измерений, который наполняется рабочим процессом первого типа. Способы организации архива измерений зависят от технических характеристик доступа к источнику данных, а в случае совпадения внутреннего формата с форматом источника данных архив измерений можно вообще исключить из рассмотрения (данные будут напрямую запрашиваться из источника данных).

Взаимодействие по типу "поставщик" осуществляется системой по запросу пользователя и представляет собой запись значимых результатов в хранилище результатов графического анализа и хранилище результатов автоматического анализа. Значимость результатов определяется исследователем и в общем случае зависит от времени. Так, допустим, некоторые результаты поиска могут быть значимыми только в разрезе получения других результатов, а следовательно, представлять интерес лишь до тех пор, пока этот результат не будет достигнут. Таким образом, способ организации хранилищ данного типа оказывает влияние на архитектуру инструмента в том смысле, что возможно потребует от инструмента наличие возможности сохранения результатов анализа независимым от Центра данных способом, что уже предусмотрено архитектурой инструмента.

Средства обработки данных. Для обработки измерений и создания новых алгоритмов рабочими процессами второго, третьего и четвертого типа используются библиотеки алгоритмов. Выделение данного средства обработки во внешнюю сущность, как представляется, позволит разделять алгоритмы между различными системами обработки и анализа. Взаимодействие инструмента с выполняемым алгоритмом из библиотеки алгоритмов осуществляется через наборы данных, которые получают на его выходе. При этом выполняться алгоритм может средствами технического уровня Центра данных, что при правильной организации работ позволит повысить эффективность работы исследователя.

В заключение заметим, что в дальнейшем рассмотренную диаграмму можно включить в состав представления, которое очерчивает границы системы и отражает ее взаимодействия с ее окружающей средой для всех заинтересованных сторон (в основном конечных пользователей и разработчиков). Для конструирования такого представления можно воспользоваться точкой зрения "Контекст" из работы [8].



Рис. 3. Инструмент как часть Центра данных

Заключение

Основным результатом работы является архитектура инструмента, автоматизирующего процесс визуального анализа временных рядов спутниковых измерений. В силу недостаточной формализации физической постановки задачи, а также постоянно эволюционирующих нужд исследователей серьезный упор был сделан на ее проектирование в разрезе деятельности ученого по исследованию сигнала. Данный подход позволил:

- провести декомпозицию процесса анализа сигнала на более простые процессы. Каждый из них, с одной стороны, реализует логически законченный набор задач по исследованию сигнала, а с другой стороны, поддается автоматизации, может выполняться самостоятельно или же параллельно с другими типами процессов, а результаты его работы зависят только от исходных данных и не зависят от работы других процессов;
- выявить и заложить в архитектуру механизмы наращивания функционала. Расширение функциональности осуществляется за счет добавления новых панелей визуализации (исследователю требуется новый тип графической модели анализируемых данных), алгоритмов (имеющиеся алгоритмы не отвечают целям исследования) и форматов данных (добавление новых источников данных, изменение смысловой нагрузки результатов анализа);
- выявить и заложить в архитектуру механизмы настройки процесса анализа. Под настройкой понимается, прежде всего, возможность комбинации имеющихся алгоритмов в схемах автоматического анализа, а также возможность для исследователя определять порядок (с некоторыми ограничениями) вызова рабочих процессов;
- выявить и заложить в архитектуру необходимость одновременного функционирования инструмента в двух различных окружающих средах: без доступа к Центру данных и с доступом к нему.

Для описания архитектуры с позиции деятельности ученого при исследовании сигнала методом визуального анализа были использованы понятия и подходы из работы [5]. Все полученные при этом продукты (диаграммы, решения, обоснования и т. п.), с одной стороны, понятны конечным пользователям, а с другой стороны, единообразным образом могут быть включены в документацию по разработке и со-

проведению инструмента, обеспечивая тем самым устойчивую платформу для его развития и внедрения.

В заключение отметим, что предложенная архитектура на данный момент в полной мере учитывает только специфику спутниковых измерений и самой физической постановки задачи. Это выражается в наличии:

- специализированного типа панели визуализации — 3D-панели, которая отвечает целям исследователя;
- функционала для разработки ученым новых методов анализа в рамках уже функционирующего инструмента, что крайне необходимо в ситуации отсутствия физических моделей исследуемых явлений;
- возможности добавления новых форматов входных данных в уже функционирующую реализацию (разработка новых методов анализа нередко требует от ученого анализа измерений с различных приборов различных КА, которые хранятся в различных форматах);
- возможности переключения из режима автономной работы в режим, когда инструмент "подключен" к ННК ИКИ РАН (часть результатов научных космических экспериментов на данный момент размещена исключительно в архивах зарубежных космических агентств).

Однако все вышеперечисленное благодаря возможности расширения функциональности и поведения уже функционирующей реализации предложенной архитектуры не препятствует интеграции инструмента в любую другую отрасль знаний, где есть необходимость в поиске связей между переменными, когда представления об этих связях отсутствуют или недостаточны.

Список литературы

1. ICE Experiment — Electric Field Instrument [Электронный ресурс]. URL: <http://demeter.cnrs-orleans.fr/> (дата обращения 25.04.2019).
2. Milikh G. M., Vartanyan A., Papadopoulos K., Parrot M. Focusing of HF radio-waves by ionospheric ducts // *J. Atmos. Solar-Terr. Phys.* 2011. V. 73, N. 13. P. 1647-1680. doi:10.1016/j.jastp.2011.02.022.
3. Mogilevsky M., Chuginin D., Moiseenko I., Romantsova T. Magnetospheric modification caused by HF heater // *Proc. 40th COSPAR Scientific Assembly. Moscow, 2—10 August 2014.* C5.1-0016-14.
4. Корнева Н. Н., Могилевский М. М., Назаров В. Н. Визуальный анализ как способ интерпретации результатов спутниковых ионосферных измерений для поисковых задач // *Геомагнетизм и аэронавигация.* 2016. Т. 56, № 3. doi:10.7868/S0016794016030093.

5. **ISO/IEC/IEEE 42010:2011** Systems and software engineering — Architecture description / 1st ed. 2011.

6. **ГОСТ Р 57100—2016** Системная и программная инженерия. Описание архитектуры. М.: Стандартинформ. 2016.

7. **Standard ECMA-376** Office Open XML File Formats. Part 2: Open Packaging Conventions (OPC) / 1st ed. 2006.

8. **Rozanski N., Woods E.** Software systems architecture: working with stakeholders using viewpoints and perspectives. 2nd ed. Addison-Wesley, 2012. 679 c.

N. N. Korneva, Lead Programmer, nkorneva@romance.iki.rssi.ru,
V. N. Nazarov, Chief of Department, vnazarov@romance.iki.rssi.ru,
M. M. Mogilevsky, Chief of Laboratory, mogilevsky2012@gmail.com,
Space Research Institute, Russian Academy of Sciences, Moscow

The Instrument Architecture for Visual Analysis of Satellite Measurements in Exploratory Problems

The paper considers the architecture of the instrument that automates the process of visual analysis of time series of satellite measurements. Most of the focus is on design and description of the architecture under conditions of insufficient formalization of the physical problem definition, as well as constantly evolving of end users (researchers) needs. Concepts and principles of the international standard ISO/IEC/IEEE 42010:2011 Systems and software engineering — Architecture description (in the Russian Federation was brought into force in 2017 as GOST R 57100-2016 Системная и программная инженерия. Описание архитектуры) are used for the architecture describing. The instrument architecture is also considered in terms of the possibility of simultaneous functioning of the instrument developed on its basis in two different environments: in an environment without access to the Space Research Institute science ground segment and in an environment with access to it.

Keywords: software architecture, ISO/IEC/IEEE 42010:2011, GOST R 57100-2016, viewpoint, view, environment, visual analysis of satellite measurements, exploratory problem

DOI: 10.17587/it.26.88-96

References

1. **ICE Experiment** — Electric Field Instrument [Electronic source], available at: <http://demeter.cnrs-orleans.fr/> (access date 25.04.2019).

2. **Milikh G. M., Vartanyan A., Papadopoulos K., Parrot M.** Focusing of HF radio-waves by ionospheric ducts, *J. Atmos. Solar-Terr. Phys.*, 2011, vol. 73, no. 13, pp. 1647–1680. doi:10.1016/j.jastp.2011.02.022.

3. **Mogilevsky M., Chugunin D., Moiseenko I., Romantsova T.** Magnetospheric modification caused by HF heater, *Proc. 40th COSPAR Scientific Assembly*, Moscow, 2–10 August 2014, C5.1-0016-14.

4. **Korneva N. N., Mogilevskii M. M., Nazarov V. N.** Visual analysis as a method of interpretation of the results of satellite ionospheric measurements for exploratory problems, *Geomagnetism and Aeronomy*, 2016, vol. 56, no. 3, doi:10.1134/S0016793216030099 (in Russian).

5. **ISO/IEC/IEEE 42010:2011** Systems and software engineering — Architecture description, 2011.

6. **GOST R 57100-2016** Systems and software engineering. Architecture description, Standartinform, 2016 (in Russian).

7. **Standard ECMA-376** Office Open XML File Formats. Part 2: Open Packaging Conventions (OPC), 2006.

8. **Rozanski N., Woods E.** Software systems architecture: working with stakeholders using viewpoints and perspectives, Addison-Wesley, 2012. 679 p.

Г. В. Данилов¹, канд. мед. наук, уч. секретарь, e-mail: gdanilov@nsi.ru,
А. А. Потапов², акад. РАН, проф., e-mail: apotapov@nsi.ru,
И. Н. Пронин², акад. РАН, проф., e-mail: pronin@nsi.ru,
М. А. Шифрин², канд. физ.-мат. наук, вед. спец., e-mail: shifrin@nsi.ru,
В. А. Васенин², док. физ.-мат. наук, проф., e-mail: vassenin@msu.ru,
А. Е. Подопрigора¹, канд. мед. наук, врач-нейрорентгенолог, e-mail: podop@nsi.ru,
В. В. Назаров³, мл. науч. сотр., e-mail: vvnazarov@nsi.ru,
Н. В. Ласунин³, канд. мед. наук, нач. отдела, e-mail: nlasunin@nsi.ru,
Т. Е. Пронкина³, вед. инженер, e-mail: pronkina@nsi.ru,
Ю. В. Струнина³, вед. инженер, e-mail: ustrunina@nsi.ru,
Ю. А. Дорофeюк³, канд. техн. наук, ст. науч. сотр., e-mail: dorofeyuk_julia@mail.ru,
¹ ФГАУ "Национальный медицинский исследовательский центр нейрохирургии
имени академика Н. Н. Бурденко" Минздрава России,
² Московский государственный университет им. М. В. Ломоносова,
³ Институт проблем управления им. В. А. Трапезникова РАН

Методология оценки активности и результативности научной деятельности в медицинской организации: опыт Национального медицинского исследовательского центра нейрохирургии имени академика Н. Н. Бурденко¹

Учет результативности научной деятельности организации — ключевой инструмент для разработки стратегического плана ее развития и управления. В настоящей статье описан опыт ФГАУ "НМИЦ нейрохирургии им. акад. Н. Н. Бурденко" Минздрава России по разработке методологии учета и анализа результатов научной деятельности с 2014 по 2019 гг. в целях повышения ее эффективности путем стимулирования отдельных сотрудников. В отмеченный период по объективным причинам система анализа научной результативности эволюционировала. Однако предложенные методы ее мониторинга и оценки позволили сохранить тенденцию роста научной результативности и при этом поставить не менее важные задачи управления качеством научных работ.

Ключевые слова: эффективность научной деятельности, система мониторинга научной деятельности, учет, отчетность, ключевые показатели эффективности, нейрохирургия

Введение

Учет активности и результативности научной деятельности организации — ключевой инструмент управления ею и основание для стратегического планирования. Учет научной продукции позволяет объективно оценивать ее количество и качество, определять рейтинг организации в научной среде, а также оцени-

вать результативность научной деятельности отдельных работников, что открывает возможности дальнейшего развития [1–5].

Под результативностью научной деятельности принято понимать, прежде всего, объемы научной продукции разных видов, созданной в течение заданного периода времени. Оценка результативности научной деятельности проводится с использованием измеримых ключевых показателей (от англ. Key Performance Indicators, KPI). К таким показателям можно отнести число (долю) научных продуктов определенной ка-

¹ Работа выполнена при поддержке Российского Фонда фундаментальных исследований (грант 16-29-12880).

тегории (например, число работ в журналах с импакт-фактором более 10), индексы цитируемости ученых и их динамику, число журналов определенной категории, в которых за отчетный период опубликованы работы [6—11]. Очевидно, что выбор ключевого показателя должен определяться той задачей, ради которой он формируется. Полноценный, непрерывный и качественный учет научных продуктов со всеми их атрибутами может быть обеспечен исключительно с помощью информационных технологий. Автоматизированные информационные системы, разрабатываемые различными учреждениями для анализа своей публикационной активности, как правило, являются инструментами внутреннего пользования, адаптированными под потребности каждой организации.

В настоящее время существует ряд коммерческих информационно-аналитических систем, предназначенных для наукометрического анализа в научно-исследовательских организациях и в высших учебных заведениях. Примером такого продукта является информационная система "БИТ.Наука" на базе программных технологий IC [12]. Данный вид программного обеспечения позволяет гибко адаптировать учет и анализ научной продукции под разнообразные запросы учреждения, принимая во внимание особенности его структурной и функциональной организации. Подобные программные продукты предназначены для локального использования внутри организации.

Альтернативным решением является учет публикационной активности в независимой системе, которая одновременно обеспечивает открытое представление публикаций и других результатов научной деятельности в сети Интернет. В настоящее время подобный сервис в числе целого ряда других предоставляет информационно-аналитическая система "ИСТИНА", разработанная в Московском государственном университете им. М. В. Ломоносова [13]. Использование такого решения делает общедоступными результаты научной деятельности организации и ее сотрудников, способствует продвижению этих результатов в научном сообществе и формированию имиджа учреждения.

Еще одним источником данных для учета и анализа являются наукометрические базы — Российский индекс научного цитирования (РИНЦ), Scopus, Web of Science и др. [14]. Информация из этих источников позволяет проводить сравнительную оценку научной результативности разных ученых и исследовательских

организаций. Одним из основных недостатков подобных баз являются задержки в размещении информации об уже опубликованных работах, тогда как динамический учет публикационной активности сотрудников требует оперативного анализа и отслеживания результативности сотрудников в реальном времени.

Несмотря на ряд достоинств независимые инструментальные средства, подобные отмеченным выше, не могут в полной мере обеспечить всесторонний учет научной продукции учреждения, поскольку требуют дополнительной адаптации к его особым потребностям.

В настоящей статье освещена методология учета и анализа результатов научной деятельности, реализованная в Национальном медицинском исследовательском центре нейрохирургии им. Н. Н. Бурденко (ФГАУ "НМИЦ нейрохирургии им. акад. Н. Н. Бурденко" Минздрава России), с учетом задач стимулирования научной активности в 2014—2019 гг. [15]. При разработке предлагаемого инструментария авторы исходили, прежде всего, из потребностей НМИЦ нейрохирургии, однако созданная методология может быть использована и в других медицинских учреждениях, для которых характерна высокая научная активность.

Методология и технологии учета и оценки научной продукции НМИЦ нейрохирургии им. Н. Н. Бурденко, реализованная в период 2014—2016 гг.

С 2014 г. в НМИЦ нейрохирургии внедрена система премирования сотрудников по результатам научной деятельности в течение календарного года. Оценка научной результативности учитывала число и качество научных публикаций, подготовленных за указанный отчетный период. Такой подход с самого начала требовал системной организации следующих процессов:

- 1) сбор данных о результатах научной деятельности сотрудников (прежде всего публикационной);
- 2) определение ценности научных продуктов;
- 3) индивидуальная оценка научной результативности сотрудников;
- 4) определение размера финансового вознаграждения сотрудников.

В 2014—2016 гг. реализация каждого процесса совершенствовалась. В конце 2016 г. система премирования сотрудников за научную результативность имела следующие ключевые свойства:

1) основания для премирования сотрудника — показатель его публикационной активности в отчетный период;

2) период, за который проводился анализ публикационной активности, — 1 год;

3) сбор данных о научных публикациях проводился централизованно путем заполнения электронных таблиц сотрудниками;

4) в соответствии с действующей в 2016 г. классификацией научных продуктов (статьи зарубежные и отечественные, монографии зарубежные и отечественные, тезисы на зарубежных и отечественных конференциях, патенты и т. д.) продукт каждого класса оценивался фиксированным числом баллов;

5) финансовое вознаграждение начислялось сотрудникам в соответствии с разработанным в НМИЦ нейрохирургии критерием премирования, рассчитываемым на основе суммы набранных баллов каждым отдельным сотрудником за подготовленные им научные публикации и другие результаты интеллектуальной деятельности.

В 2017 г. в связи с необходимостью выполнения Указа Президента РФ от 7 мая 2012 г. № 597 "О мероприятиях по реализации государственной социальной политики" и перехода на систему оплаты труда в соответствии с принципами эффективного трудового договора возникла потребность регулярной (ежемесячной) оценки результативности научной деятельности сотрудников НМИЦ нейрохирургии. Для решения этой комплексной проблемы были определены следующие важные задачи:

1) определить период, в рамках которого показатель научной результативности сотрудника является достаточно репрезентативным;

2) упростить способ сбора данных о научных результатах сотрудников;

3) упростить способ проверки собранных данных о научных продуктах сотрудников;

4) сделать систему ценностей научной продукции более гибкой и дифференцированной, а также учитывающей вариации в ценности научных продуктов одного класса (например, статей в журналах с разным импакт-фактором);

5) максимально автоматизировать все бизнес (деловые) процессы в системе учета, анализа и оценки научной продуктивности сотрудников.

Принципы оценки научной деятельности, действовавшие в конце 2016 г., не позволяли оптимально решать описанные выше задачи по ряду следующих причин. Система ценностей научной продукции в 2016 г. была относительно жесткой и не учитывала разное потен-

циальное качество научной продукции одного класса. Например, статья в зарубежной печати могла быть издана в журналах с разным импакт-фактором, что косвенно отражало трудовые затраты и активность автора, но не было учтено в системе ценностей.

Ежемесячная оценка научной результативности по модели расчетов 2016 г. оказалась бы легко "смещаемым", нестабильным показателем, который не отражал бы истинную вовлеченность и продуктивность сотрудника в течение года. Публикации каждого сотрудника выходят в печать, как правило, неравномерно в течение года. Следовательно, для объективной оценки научной результативности сотрудника требуется рассматривать его деятельность на достаточно длинном интервале времени.

Система сбора данных о научных продуктах была реализована в виде стандартных веб-форм (электронных таблиц), которые позволяли эффективно собирать и хранить данные в едином архиве, но не решали задачу контроля дублирования данных и валидации их при вводе. Данные об одной и той же публикации вносил каждый соавтор, число которых могло достигать 10 и более человек. Это приводило к дополнительным сложностям при проверке работ, потенциальной неполноте и невозможности полностью исключить ошибки в данных.

Отсутствие какого-либо специализированного аналитического программного обеспечения также делало невозможным быстрое решение задачи ежеквартальной проверки и анализа собранных данных.

Система оценки активности и результативности научной деятельности (АРНД) в НМИЦ нейрохирургии им. Н. Н. Бурденко в 2017—2019 гг.

Для максимально объективной оценки научной продуктивности сотрудника и адекватного материального вознаграждения в 2017 г. потребовалось разработать новый критерий, формируемый по объективным данным и отражающий совокупную ценность научных продуктов, подготовленных сотрудником на репрезентативном промежутке времени, при этом легко рассчитываемый в любой точке используемой временной шкалы. Данная оценка могла быть использована для расчета рейтинга сотрудника в целях поддержки здоровой научной конкуренции и мотивации на получение более значимых научных результатов.

Периодический расчет такого показателя (ежемесячного или ежеквартального) потребовал непрерывного сбора данных о научных продуктах, выпускаемых сотрудниками НМИЦ нейрохирургии. Для максимальной автоматизации процесса сбора, валидации и учета научной продукции было принято решение использовать специализированную информационную систему, данные в которую могли бы вносить сами сотрудники или выделенный под эту задачу персонал. Процедуру проверки вводимых сотрудниками данных в систему необходимо было максимально упростить и по возможности автоматизировать. Частично задачи валидации данных планировалось решать на уровне самой системы, максимально автоматизировав при этом расчет рейтинговой оценки.

Разработку новой системы учета и оценки научной продукции начали с определения ключевых показателей научной активности и результативности.

Ключевые показатели для непрерывной оценки активности и результативности научной деятельности в 2017–2019 гг.

Ключевые показатели (англ. Key Performance Indicators, KPI) — индикаторы деятельности организации, которые помогают ей в достижении стратегических и тактических (операционных) целей.

Разработка адекватных и эффективных KPI становится принципиально важной в процессе перехода оценки труда сотрудников на систему "эффективного трудового договора (контракта)".

Эффективный контракт — это трудовой договор с работником, в котором конкретизированы его должностные обязанности, условия оплаты труда, показатели и критерии оценки эффективности деятельности для назначения стимулирующих выплат в зависимости от результатов труда и качества оказываемых услуг, а также меры социальной поддержки.

Для реализации принципов эффективного контракта в части научных исследований и разработок в НМИЦ нейрохирургии были предложены следующие KPI:

- 1) оценка результативности научной деятельности сотрудника за фиксированный период;
- 2) оценка активности участия сотрудника в научных проектах НМИЦ нейрохирургии за фиксированный период.

В 2017 г. в систему учета была внедрена оценка *результативности научной деятельности со-*

трудника в качестве первого KPI по научной деятельности. Для расчета рейтинговой оценки рассматривался трехлетний период — временной интервал, в течение которого показатель научной результативности сотрудника являлся достаточно объективным. Данная оценка рассчитывалась с учетом объемов научной продукции разных категорий и различной условной ценности. В связи с тем, что в 2015, 2016 и 2017 гг. системы ценностей научной продукции различались, для расчета обобщающей оценки научной результативности за 3 года использовались относительные величины.

Показатель *активности научной деятельности* — второй KPI для оценки научной деятельности сотрудников НМИЦ нейрохирургии — величина, отражающая участие сотрудника в плановых научных проектах организации. Такой показатель был введен в практику учета в 2018 г.

На основании перечисленных выше показателей формировалась общая рейтинговая оценка научной активности и результативности (далее — *рейтинговая оценка*) для каждого сотрудника.

Рейтинговая оценка — величина, отражающая совокупную ценность научных продуктов, подготовленных отдельным сотрудником за фиксированный отрезок времени, а также долю участия соответствующего сотрудника в плановых научных проектах НМИЦ нейрохирургии. Рейтинговая оценка сотрудника за фиксированный период рассчитывалась как вклад сотрудника в общую ценность научной продукции НМИЦ нейрохирургии и в работу по запланированным научным проектам. Значение рейтинговой оценки определяет долю ежемесячной премии сотрудника за научную активность и результативность. Методика расчета данной оценки подробно изложена в следующем разделе.

Для расчета отмеченных выше показателей используется система ценностей научной продукции, модифицированная в НМИЦ нейрохирургии им. Н. Н. Бурденко в 2017 г.

Система ценностей научной продукции НМИЦ нейрохирургии им. Н. Н. Бурденко в 2017–2019 гг.

"Система ценностей" научных продуктов НМИЦ нейрохирургии — это система балльных оценок каждого вида научного продукта. *Общая ценность* научной продукции НМИЦ нейрохирургии — *сумма оценок* всех научных продук-

тов для всех сотрудников (измеряется в баллах). В такой системе оцениваются научные продукты следующих категорий: монографии; главы в монографиях; научные статьи; диссертации; патенты; свидетельства о регистрации программного обеспечения; научно-популярные статьи; тезисы в изданиях, индексируемых в базах данных SCOPUS или Web of Science.

Полная классификация научных продуктов НМИЦ нейрохирургии и соответствующие им балльные оценки приведены в табл. 1.

Оценка результативности научной деятельности сотрудника за фиксированный период представляет собой сумму оценок всех научных продуктов, подготовленных сотрудником в рамках соответствующего периода. Данная

Таблица 1

Классификация и градация оценок научных продуктов в НМИЦ нейрохирургии

Научный продукт	Балл
Научная монография зарубежная (для титульных авторов)	12
Класс 1 (методические рекомендации, брошюры до 50 страниц включительно, переизданные монографии)	12
Класс 2 (научные монографии более 50 до 100 страниц включительно)	16
Класс 3 (научные монографии более 100 до 300 включительно)	24
Класс 4 (большие научные монографии более 300 страниц)	30
Научная монография отечественная (для титульных авторов)	
Класс 1 (методические рекомендации, брошюры до 50 страниц включительно, переизданные монографии)	10
Класс 2 (научные монографии более 50 до 100 страниц включительно)	12
Класс 3 (научные монографии более 100 до 300 включительно)	18
Класс 4 (большие научные монографии более 300 страниц)	24
Глава в зарубежной научной монографии (для автора/соавтора главы)	
Класс 1 (в монографиях объемом до 100 страниц включительно)	10
Класс 2 (в монографиях объемом более 100 страниц)	12
Глава в отечественной научной монографии (для автора/соавтора главы)	
Класс 1 (в монографиях объемом до 100 страниц включительно)	4
Класс 2 (в монографиях объемом более 100 страниц)	6
Научная статья в зарубежном журнале или в отечественном журнале на английском языке (для первого автора)	
в журнале с неопределенным импакт-фактором	9
в журнале с импакт-фактором ≤ 1	10
в журнале с импакт-фактором более 1 до 2 включительно	12
в журнале с импакт-фактором более 2 до 4 включительно	13
в журнале с импакт-фактором более 4	15
Научная статья в зарубежном журнале или в отечественном журнале на английском языке (для соавтора)	
в журнале с неопределенным импакт-фактором	4
в журнале с импакт-фактором ≤ 1	5
в журнале с импакт-фактором более 1 до 2 включительно	6
в журнале с импакт-фактором более 2 до 4 включительно	7
в журнале с импакт-фактором более 4	8
Научная статья в отечественном журнале (для первого автора)	6
Научная статья в отечественном журнале (для соавтора)	3
Научная статья в зарубежном сборнике (для первого автора)	10
Научная статья в зарубежном сборнике (для соавтора)	5
Научная статья в отечественном сборнике (для первого автора)	6
Научная статья в отечественном сборнике (для соавтора)	3
Диссертация	
Докторская	24
Кандидатская	16
Другие результаты интеллектуальной деятельности	
Патент	12
Свидетельство о регистрации прав на программное обеспечение	12
Научно-популярные и другие публикации	
Научно-популярные статьи, публикации в СМИ	2
Тезисы, индексируемые в базах данных Scopus или Web of Science в международных журналах	1
Тезисы, индексируемые в базах данных Scopus или Web of Science в отечественных журналах	0,5

оценка отражает совокупную ценность научной продукции, подготовленной сотрудником.

Оценка текущей активности сотрудника в научных проектах НМИЦ нейрохирургии за фиксированный период представляет собой произведение доли участия сотрудника (не более 0,3) в научном проекте согласно протоколу исследования, утвержденному Ученым советом НМИЦ нейрохирургии им. Н. Н. Бурденко, и условной ценности одного научного проекта (20 баллов).

Сумма оценки результативности научной деятельности сотрудника и оценки активности сотрудника в научных проектах за фиксированный период формируют ценность научной активности и результативности конкретного сотрудника за фиксированный период.

Вклад сотрудника в общую ценность научной продукции и научную активность НМИЦ нейрохирургии за фиксированный период рассчитывается как доля ценности научной активности и результативности сотрудника в общей сумме ценностей научной активности и результативности всех сотрудников.

Методика ежеквартального расчета рейтинговой оценки

Для ежеквартального расчета рейтинговой оценки используется "квартальный коэффициент", равный временной доле соответствующего и предшествующих кварталов в календарном году (т. е. 0,25 — для первого квартала, 0,5 — для второго квартала, 0,75 — для третьего квартала, 1 — для четвертого квартала). Такой подход с ежеквартальным перерасчетом рейтинговой оценки был предложен в 2017 г., поскольку в тот период информационная система учета не сохраняла точную дату добавления продукта в базу данных и не могла предоставить "фильтр" для отбора научных продуктов за произвольный временной интервал (например, за три года от определенной даты).

Рейтинговая оценка (РО) в текущем году $YYYY$ рассчитывается ежеквартально по следующей формуле:

$$PO_{YYYY} = (ВКЛАД_{YYYY-3} \cdot (1 - КК) + ВКЛАД_{YYYY-2} + ВКЛАД_{YYYY-1} + ВКЛАД_{YYYY} \cdot КК) / 3,$$

где используются следующие обозначения:

КК — кварталный коэффициент;

$ВКЛАД_{год} = [СУММА\ БАЛЛОВ\ СОТРУДНИКА_{год}] / [СУММА\ БАЛЛОВ\ ВСЕХ\ СОТРУДНИКОВ\ ОРГАНИЗАЦИИ_{год}]$;

$СУММА\ БАЛЛОВ\ СОТРУДНИКА_{год}$ — сумма оценки результативности научной деятельности сотрудника и оценки активности сотрудника в научных проектах за год;

$СУММА\ БАЛЛОВ\ ВСЕХ\ СОТРУДНИКОВ\ ОРГАНИЗАЦИИ_{год}$ — сумма ценностей всех совокупных научных продуктов и научной активности всех сотрудников за год.

Таким образом, рейтинговая оценка сотрудника отражает вклад сотрудника в общую ценность научной активности и результативность НМИЦ нейрохирургии за предшествующий период времени (3 года).

Перерасчет рейтинговой оценки сотрудника происходит ежеквартально, рейтинговая оценка применяется ежемесячно для определения размера стимулирующих выплат в течение квартала, следующего за ее перерасчетом.

Таким образом, в результате применения предложенной формулы оценка научной активности происходит с помощью так называемого скользящего окна шагом в три года, позволяющего объективно учитывать вклад сотрудника за релевантный временной промежуток.

Схема ежеквартального расчета рейтинговой оценки показана на рис. 1 (см. вторую сторону обложки).

На рис. 1 видно, что в 2018 г. при ежеквартальном перерасчете рейтинга постепенно "ослабевает" оценка 2015 г. и приобретает больший вес оценка 2018 г. При этом в конце 2018 г. рейтинг обновляется только для 2016—2018 гг. Размер ежемесячной денежной премии при такой рейтинговой системе может быть рассчитан как произведение рейтинговой оценки и части фонда оплаты труда, выделенной в текущем месяце на премирование сотрудников за научную активность.

Информационно-аналитический программный комплекс для мониторинга и оценки результатов научно-исследовательской деятельности

В качестве основных компонентов информационно-аналитического программного комплекса для мониторинга и оценки результатов научно-исследовательской деятельности в НМИЦ нейрохирургии им. Н. Н. Бурденко были определены следующие:

1) модуль сбора, учета и предварительного анализа данных о научной деятельности;

2) модуль проверки и валидации данных о научной продукции;

3) модуль анализа учетной научной продукции и расчета рейтинговых оценок.

Схема процесса, который должен обеспечить первый этап ввода и валидации данных о научных продуктах самими сотрудниками (первые два модуля), показана на рис. 2 (см. вторую сторону обложки).

С учетом полученного в 2014—2016 гг. опыта сбора и анализа данных о научной продуктивности были сформулированы требования к информационной системе учета результатов научной деятельности:

1) наличие в системе персонального аккаунта сотрудника с указанием идентифицирующих атрибутов;

2) возможность пополнения базы данных сотрудников организации;

3) наличие классификатора научной продукции;

4) возможность добавления классификаторов администратором от учреждения;

5) наличие механизмов валидации вводимой информации, защита от "некорректного" ввода данных;

6) автоматизированная связь с отечественными и международными базами цитирования;

7) уведомления соавторов по e-mail с запросом подтверждения соавторства;

8) анализ потенциального дублирования записей о научных продуктах и научных сотрудниках в базе данных;

9) программный интерфейс для выгрузки исходных данных;

10) автоматизированное вычисление рейтинга научной продуктивности.

Среди доступных в 2017 г. решений описанным выше критериям максимально соответствовала Интеллектуальная Система Тематического Исследования НАукометрических данных ("ИСТИНА"), разработанная в Московском государственном университете им. М. В. Ломоносова [13, 16]. Система позволяет вносить информацию в относительно простом для пользователя графическом интерфейсе и в определенной степени обеспечивает контроль качества вносимых данных. Снимок одного из экранов графического интерфейса системы "ИСТИНА" показан на рис. 3 (см. третью сторону обложки).

Система "ИСТИНА" предоставляет инструментальные средства формирования отчетов по шаблонным формам. Однако классификация результатов научной деятельности в этой системе и в НМИЦ нейрохирургии не полностью соответствуют друг другу. В Лаборатории биомедицинской информатики и искусствен-

ного интеллекта НМИЦ нейрохирургии на языке статистического программирования R (версия 3.4.4) с использованием специализированных пакетов для выполнения http-запросов и технологии быстрого прототипирования веб-приложений "shiny" в IDE RStudio (версия 1.1.442) был разработан комплекс аналитических программ для задач проверки данных и расчета рейтинговых оценок. Данные из системы "ИСТИНА" экспортируются через API после передачи специального токена, разбираются функциями R-приложения и анализируются по заданному сценарию.

Снимок одного из экранов графического интерфейса пользователя аналитического модуля, созданного в Лаборатории биомедицинской информатики и искусственного интеллекта, представлен на рис. 4 (см. третью сторону обложки).

В 2017 г. информационная система "ИСТИНА" и локальный программный аналитический модуль НМИЦ нейрохирургии стали использоваться в едином комплексе в целях учета и анализа научных результатов.

Администрирование аккаунта НМИЦ нейрохирургии в информационной системе "ИСТИНА" обеспечивается, как правило, двумя (реже — тремя) администраторами-суперпользователями. В рабочую группу для поддержки пользователей системы НМИЦ нейрохирургии включены 16 сотрудников — ответственных по структурным подразделениям. Проверка и анализ данных выполняется двумя администраторами-суперпользователями. Число активных пользователей системы в течение года составляет около 200 человек.

Динамика количества научной продукции НМИЦ с 2014 по 2018 гг.

В результате внедрения системы мотивации научной деятельности сотрудников НМИЦ нейрохирургии объемы ежегодной научной продукции с 2014 по 2018 гг. существенно увеличивались (табл. 2). При этом несмотря на изменения в системе учета научной активности и премирования сотрудников этот тренд не менялся. Так, в 2018 г. было опубликовано 230 статей в научных рецензируемых журналах, включая 52 статьи на английском языке (против 111 статей в 2014 г., включая 8 работ на английском языке, табл. 2).

Мотивационная система, основанная на финансовом поощрении научной деятельности, привела к увеличению объемов научной продукции практически в 2 раза за 4 года. Создан-

ные механизмы расчета рейтинговой оценки позволили обеспечить справедливое распределение премиальных средств в части научной деятельности между сотрудниками Центра и выполнить мероприятия дорожной карты (Указ Президента Российской Федерации № 597 от 7 мая 2012 г. № 597) по повышению заработной платы, а также обеспечить основу для перехода на принципы эффективного контракта в части научной деятельности Центра.

Качественный состав опубликованных статей как наиболее значимой и репрезентативной научной продукции Центра нейрохирургии показал, что в 2018 г. из 230 статей 52 работы были

опубликованы на английском языке, причем 34 статьи — в журналах, имеющих международный импакт-фактор (включая 12 статей в профильных нейрохирургических журналах).

В табл. 3 представлены журналы, в которых в 2018 г. были опубликованы работы сотрудников НМИЦ нейрохирургии и для которых определен импакт-фактор в системе Scopus или Web of Science.

При существенном увеличении объема публикаций важным является обеспечение их ценности и качества. Детальный анализ структуры публикаций выявил относительно небольшую долю статей в профильных высокорейтинговых международных нейрохирургических журналах, в которых сотрудники НМИЦ нейрохирургии им. Н. Н. Бурденко были бы первыми авторами. Такой показатель при значительном размере научного продукта НМИЦ нейрохирургии свидетельствует, прежде всего, о высоком потенциале для публикации результатов исследований в высокорейтинговых международных изданиях.

Основаниями для этого вывода также являются:

1) очень высокая интенсивность клинической работы и уникальный по объему поток пациентов (более 9500 оперативных вмешательств и более 2600 радиотерапевтических/радиохирургических процедур ежегодно);

2) уникальный по набору компетенций и квалификации коллектив НМИЦ нейрохирургии им. Н. Н. Бурденко;

3) наличие самых современных диагностических и хирургических инструментов, позволяющих решать научные задачи на высоком технологическом уровне.

Как показывают данные в табл. 2, внедрение в практику системы ежеквартальной рейтинговой оценки на основании публикационной активности в период последних 3 лет с ежемесячным премированием вместо системы премирования по результатам каждого календарного года не изменило возрастающий тренд на увеличение публикационной активности и нарастания доли научных публикаций в международных рейтинговых журналах, что является одним из стратегических показателей для развития науки в НМИЦ нейрохирургии им. Н. Н. Бурденко. Наличие журналов с высоким импакт-фактором, в которых в 2018 г. были опубликованы статьи в соавторстве с сотрудниками НМИЦ нейрохирургии им. Н. Н. Бурденко, также свидетельствует о потенциале дальнейшего развития публикационной активности.

Таблица 2

**Результаты научной деятельности сотрудников
НМИЦ нейрохирургии в динамике с 2014 по 2018 гг.**

Показатели	2014 г.	2015 г.	2016 г.	2017 г.	2018 г.
Статьи в журналах, <i>в том числе международных</i>	111 8	135 12	151 24	210 43	230 52
Монографии	8	10	11	19	20
Патенты	3	8	5	12	12
Совокупный импакт-фактор	198	168	279	231	331
Индекс Хирша организации (РИНЦ)	31	37	43	47	52

Таблица 3

**Публикации сотрудников в международных журналах,
имеющих импакт-фактор, в 2018 г. (n = 34)**

Название журнала	Импакт-фактор
Nature	41,577
Lancet Oncology	36,418
Acta Neuropathologica	15,872
Nucleic Acids Research	11,561
Neuro-Oncology	9,384
Anesthesiology	6,523
British Journal of Anaesthesia	6,499
Neuropathology and Applied Neurobiology	6,059
European Journal of Anaesthesiology	3,958
Antimicrobial resistance and infection control	3,568
Behavioural Brain Research	3,173
Journal of Critical Care	2,872
Minerva Anestesiologica	2,693
Current Opinion in Anaesthesiology	2,581
Journal of Molecular Neuroscience	2,454
Consciousness and Cognition	2,272
Acta Neurochirurgica	1,929
World Neurosurgery	1,924
International Journal of Neuroscience	1,848
Pathogens and global health	1,703
American Journal of Human Biology	1,575
Journal of Craniofacial Surgery	0,772

Выводы и дальнейшие перспективы

Для успешного развития качества и увеличения количества научной продукции НМИЦ нейрохирургии необходимо решение задач управления качеством научных исследований. Результаты фундаментальных и прикладных исследований самого высокого качества необходимы для принятия клинических решений, что ратифицировано концепцией доказательной медицины и национальным стандартом "Надлежащая клиническая практика" (Good Clinical Practice, GCP, ГОСТ Р 52379—2005), которые устанавливают высокие требования к методологии и организации исследований [17]. В настоящее время эти требования формализованы Международным советом по гармонизации исследований с участием человека в виде инструкций, которые стали de-facto стандартом и регламентом клинических испытаний. Обеспечить высокое качество исследований возможно только с помощью организации специализированной службы, занимающейся проведением фундаментальных и прикладных исследований с участием человека или с использованием биологического материала.

Интенсификация проведения фундаментальных исследований в области нейронаук, формирование и интегрирование в медицину инновационной биомедицинской науки для обеспечения опережающего научно-технологического развития и ускоренного внедрения в практику здравоохранения научных разработок соответствуют приоритетным направлениям развития науки, технологий и техники в Российской Федерации, перечню критических технологий Российской Федерации, стратегии научно-технологического развития Российской Федерации (Указ Президента РФ № 642 от 1 декабря 2016 г.) и национальным целям и стратегическим задачам развития Российской Федерации на период до 2024 г. (Указ Президента РФ от 1 декабря 2016 г.). В связи с важностью обеспечения высокого методического качества научных исследований в НМИЦ нейрохирургии создано специализированное научное подразделение — Институт нейронаук и технологий, в задачи которого входят консолидация усилий в области фундаментальных и прикладных исследований, а также обеспечение исследований с участием пациентов самой современной методологией с позиций доказательной медицины. Развитие системы планирования исследований параллельно с учетом результатов научной деятель-

ности видится наиболее значимым шагом для обеспечения дальнейшего роста качества и количества научной продукции.

Список литературы

1. **Gilyarevskii R. S.** Publication activity as an indicator of scientific performance // *Scientific and Technical Information Processing*. 2014. Vol. 41, N. 3. P. 170—177.
2. **Franceschini F., Galetto M., Maisano D.** Management by measurement: Designing key indicators and performance measurement systems. Berlin Heidelberg: Springer, 2007. 242 p.
3. **Braun T., Glänzel W., Schubert A.** Scientometric indicators: a 32 country comparative evaluation of publishing performance and citation impact. World Scientific Publ. Co., Singapore. 1985.
4. **World University Rankings 2013-14.** Times Higher Education. URL: https://www.timeshighereducation.com/world-university-rankings/2014/world-ranking#!/page/0/length/25/sort_by/rank/sort_order/asc/cols/undefined (дата обращения: 29.09.19).
5. **Ranking Methodology of Academic Ranking of World Universities — 2013.** URL: <http://www.shanghairanking.com/ARWU-Methodology-2013.html> (дата обращения: 29.06.19).
6. **Измерение рейтингов университетов: международный и российский опыт /** Под ред. Ф. Э. Шереги и А. Л. Арефьева / Министерство образования и науки Российской Федерации. М.: Центр социологических исследований, 2014. 504 с.
7. **Lehmann S., Jackson A. D., Lautrup B. E.** Measures for measures: Are some ways of measuring scientific quality better than others? Analysis of the reliability of commonly used methods for comparing citation records // *Nature*. 2006. N. 444. P. 1003—1004.
8. **Westney L. H.** Historical rankings of science and technology: A citationist perspective // *The Journal of the Association for History and Computing*. 1998. N. 1.
9. **Batista P. D., Campiteli M. G., Kinouchi O., Martinez A. S.** Is it possible to compare researchers with different scientific interests? // *Scientometrics*. 2006. Vol. 68, Iss. 1. P. 179—189.
10. **Формула** расчета персонального рейтинга научных сотрудников НИИ механики МГУ. URL: <https://www.msu.ru/science/reiting/niimech.pdf> (дата обращения: 21.04.19).
11. **Правила** расчета рейтинга-показателя результативности научной деятельности (ПРНД) ученого. URL: <https://www.inr.ac.ru/a/pr/prnd/r12.pdf> (дата обращения: 15.05.19).
12. **БИТ.Наука** — готовое решение на базе IC. URL: <https://www.lcbt.ru/lcsoft/modul-nauka/> (дата обращения: 15.05.19).
13. **Садовничий В. А., Васенин В. А., Афонин С. А., Козицын А. С., Голомазов Д. Д.** Информационная система "ИСТИНА" как Big Data-инструментарий в области управления на основе анализа наукометрических данных // *Материалы Всероссийской конференции с международным участием "Знания — Онтологии — Теории"* (ЗОНТ-2015). Новосибирск: Институт математики им. С. Л. Соболева СО РАН. Т. 1. С. 115—123.
14. **Lehmann S., Jackson A. D., Lautrup B. E.** Measures and Mismeasures of Scientific Quality. 2005. URL: <http://arxiv.org/abs/physics/0512238>.
15. **Potapov A., Likhтерman L., Danilov G.** Great Hospitals of the Russian Federation: National Medical Research Center for Neurosurgery Named After N. N. Burdenko: History and Contemporaneity // *World neurosurgery*. 2018. Vol. 120. P. 100—111.
16. **Персональный** рейтинг работника — Документация по системе "ИСТИНА" (руководство пользователя). URL: <https://docs.istina.msu.ru/ratings/main.html> (дата обращения: 20.05.19).
17. **ГОСТ Р 52379—2005** Надлежащая клиническая практика. URL: <http://docs.cntd.ru/document/120004114> (дата обращения: 10.05.2019).

G. V. Danilov¹, Scientific Secretary, e-mail: gdanilov@nsi.ru,
 A. A. Potapov², Professor, e-mail: apotapov@nsi.ru,
 I. N. Pronin², Professor, e-mail: pronin@nsi.ru,
 M. A. Shifrin², Head Specialist, e-mail: shifrin@nsi.ru,
 V. A. Vasenin², Professor, e-mail: vasenin@msu.ru,
 A. E. Podoprigora¹, Neuroradiologist, e-mail: podop@nsi.ru,
 V. V. Nazarov³, Junior Research Associate, e-mail: vvnazarov@nsi.ru,
 N. V. Lasunin³, Department Head, e-mail: nlasunin@nsi.ru,
 T. E. Pronkina³, Lead Engineer, e-mail: pronkina@nsi.ru,
 Y. V. Strunina³, Lead Engineer, e-mail: ustrunina@nsi.ru,
 Y. A. Dorofeyuk³, Senior Research Associate, e-mail: dorofeyuk_julia@mail.ru,
¹ Federal State Autonomous Institution "N. N. Burdenko National Medical Research Center
 of Neurosurgery" of the Ministry of Health of the Russian Federation,
² Lomonosov Moscow State University,
³ Institute of Control Sciences Academician V. A. Trapeznikov

Methodology of Activity and Scientific Research Efficiency Evaluation in Medical Organization: N. N. Burdenko National Medical Research Center of Neurosurgery Experience

Accounting for the scientific research efficiency of the organization is a key tool for its strategic plan formulation for organization development and management. This article describes the experience of Federal state autonomous institution "N. N. Burdenko national medical research center of neurosurgery" of the Ministry of health of the Russian Federation on development of activity and scientific research efficiency evaluation methodology from 2014 to 2019 for the purpose of its efficiency increment by employees encouragement. During this period, for objective reasons, the system of analysis of scientific performance has been evolved. However, the proposed methods of its monitoring and evaluation allowed to maintain the trend of scientific performance growth and at the same time to set no less important tasks of scientific works quality management.

Keywords: efficiency of scientific activity, monitoring system of scientific activity, accounting, reporting, key performance indicators, neurosurgery

Acknowledgements: The work was supported by The Russian Foundation for Basic Research (grant 16-29-12880).

DOI: 10.17587/it.26.97-106

References

- Gilyarevskii R. S. Publication activity as an indicator of scientific performance. *Scientific and Technical Information Processing*, 2014, vol. 41, no. 3, pp. 170–177.
- Franceschini F., Galetto M., Maisano D. Management by measurement: Designing key indicators and performance measurement systems, Berlin Heidelberg, Springer, 2007, 242 p.
- Braun T., Glänzel W., Schubert A. Scientometric indicators: a 32 country comparative evaluation of publishing performance and citation impact, Singapore, World Scientific Publ. Co., 1985.
- World University Rankings 2013-14. Times Higher Education, available at: https://www.timeshighereducation.com/world-university-rankings/2014/world-ranking#!/page/0/length/25/sort_by/rank/sort_order/asc/cols/undefined (date of access: 29.09.19).
- Ranking Methodology of Academic Ranking of World Universities — 2013, available at: <http://www.shanghairanking.com/ARWU-Methodology-2013.html> (date of access: 29.06.19).
- Sheregi F. E., Arefiev A. L. ed. Measuring University rankings: international and Russian experience. Ministry of education and science of the Russian Federation Confederations, Moscow, Centr sociologicheskikh issledovaniy, 2014, 504 p. (in Russian).
- Lehmann S., Jackson A. D., Lautrup B. E. Measures for measures: Are some ways of measuring scientific quality better than others? Analysis of the reliability of commonly used methods for comparing citation records, *Nature*, 2006, no. 444, pp. 1003–1004.
- Westney L. H. Historical rankings of science and technology: A citationist perspective, *The Journal of the Association for History and Computing*, 1998, no. 1.
- Batista P. D., Campiteli M. G., Kinouchi O., Martinez A. S. Is it possible to compare researchers with different scientific interests?, *Scientometrics*, 2006, vol. 68, iss. 1, pp. 179–189.
- The formula for calculating the personal rating of researchers of the Institute of mechanics of Moscow state University, available at: <https://www.msu.ru/science/rejting/niimech.pdf> (date of access: 21.04.19) (in Russian).
- Rules for calculating the rating-indicator of scientific performance (PRND) of the scientist, available at: <https://www.inr.ac.ru/a/pr/prnd/r12.pdf> (date of access: 15.05.19) (in Russian).
- BIT.Science-ready solution based on IC, available at: <https://www.lcbit.ru/lcsoft/modul-nauka/> (date of access: 15.05.19) (in Russian).
- Sadovnichiy V. A., Vasenin V. A., Afonin S. A., Kozicyn A. S., Golomazov D. D. Information system "ISTINA" as Big Data-tools in the field of management based on the analysis of scientometric data, *Proceedings of conference "Knowledge-Ontology-Theory"* Novosibirsk: Institute of mathematics named by S. L. Soboleva CO PAH, vol. 1, pp. 115–123 (in Russian).
- Lehmann S., Jackson A. D., Lautrup B. E. Measures and Mismeasures of Scientific Quality, 2005, available at: <http://arxiv.org/abs/physics/0512238>.
- Potapov A., Likhтерman L., Danilov G. Great Hospitals of the Russian Federation: National Medical Research Center for Neurosurgery Named After N. N. Burdenko: History and Contemporaneity, *World Neurosurgery*, 2018, vol. 120, pp. 100–111.
- Personal rating of the employee — documentation on the system ISTINA (user manual), available at: <https://docs.istina.msu.ru/ratings/main.html> (date of access: 20.05.19) (in Russian).
- GOST R 52379—2005 Good clinical practice, available at: <http://docs.cntd.ru/document/1200041147> (date of access: 10.05.2019) (in Russian).

А. Н. Соломатин, канд. физ.-мат. наук, вед. науч. сотр., e-mail: a.n.solomatin@bk.ru,
Федеральный исследовательский центр "Информатика и управление" РАН, Москва

Интегрированные системы регионального программирования и автоматизация их разработки

Рассматриваются особенности интегрированных систем регионального программирования, состоящих из отдельных систем регионального планирования, проектирования, экологической безопасности и т. д. Интегрированная система строится непосредственно под конкретную задачу за счет автоматического синтеза программ-конвертеров, обеспечивающих информационные связи между этими системами, и существует как единое целое только в виде компьютерной модели.

Ключевые слова: интегрированные системы регионального программирования, системы планирования и проектирования, автоматизация разработки, автоматический синтез расчетных программ

Введение

В течение многих лет в отделе методов проектирования развивающихся систем ФИЦ "Информатика и управление" РАН ведется работа по разработке математических моделей, методов, алгоритмов и программных систем для решения задач комплексного освоения территорий. Было сформировано новое научное направление "региональное программирование" — область региональной экономики, где разрабатываются теория и количественные методы решения задач, возникающих при составлении проектов программ комплексного освоения территорий [1].

Задачи комплексного освоения добывающих регионов возникают при формировании и развитии на них территориально-производственных комплексов в целях освоения природных ресурсов этих территорий. Общая задача комплексного освоения разбивается на ряд тесно связанных друг с другом частных задач, таких как планирование развития основных и сопутствующих отраслей, формирование проектов развития этих отраслей, формирование схем развития производственной и социальной инфраструктуры, вариантов расселения, определение воздействия на экосистему региона и т. д.

Решение перечисленных задач комплексного освоения обеспечивается с помощью систем регионального программирования (Р-систем) —

специализированных автоматизированных систем различного назначения, в частности, систем планирования, проектирования и управления для различных отраслей, экологического мониторинга, оценки экологических рисков и т. д. [2].

Эти системы обладают многими специфическими особенностями, поэтому их можно рассматривать как самостоятельный класс сложных автоматизированных систем со следующими особенностями [3]:

- сложность объекта автоматизации, которым является осваиваемая территория со всем разнообразием природных, экономических и социальных характеристик и связей;
- необходимость комплексного решения задач планирования, проектирования, управления для различных отраслей и уровней управления;
- необходимость увязки во времени и пространстве программ развития различных отраслей;
- сложность и разнообразие используемого математического аппарата из различных областей математики;
- невозможность полной формализации описания как объекта освоения, так и процесса выбора проектов;
- неограниченно долгое время жизни развивающегося объекта комплексного освоения.

1. Интегрированные системы регионального программирования

ИР-системы и принципы их разработки. При решении задач комплексного освоения территорий могут одновременно использоваться сразу несколько различных Р-систем, образуя в процессе решения задачи некоторую единую систему. Так, в работе [2] приводится пример совместного использования нескольких Р-систем при решении задач планирования и проектирования развития нефтегазодобывающих районов Западной Сибири. Будем называть такие комплексы систем интегрированными системами регионального программирования (ИР-системами). Очевидно, что для различных задач должны быть сформированы различные ИР-системы. Для автоматизации разработки и сопровождения ИР-систем [2, 3] предлагается использовать средства автоматического синтеза программ [4].

Предлагаются следующие основные принципы разработки ИР-систем [2, 3, 5], важнейшими из которых мы считаем принципы самоорганизации, децентрализации, виртуальности и конкретности:

- модульность — ИР-системы формируются за счет различной компоновки отдельных Р-систем, которые разрабатываются и могут функционировать независимо друг от друга;
- интегрированность — ИР-системы обеспечивают интеграцию разнородных составляющих систем Р-систем одновременно по нескольким направлениям — по уровням управления, по отраслям экономики, в территориальном аспекте, по решаемым задачам и т. д. с возникновением синергетического эффекта;
- открытость — ИР-системы являются открытыми системами, ориентированными на длительный процесс постоянного развития, когда постоянно изменяются объекты планирования и проектирования, операционная среда, решаемые задачи и т. д., что влечет за собой изменение состава и функций составляющих Р-систем;
- интеллектуальность — непрерывное развитие ИР-систем на основе подходов искусственного интеллекта, включая различные способы представления знаний и автоматический синтез программ;
- самоорганизация — автоматическое построение и развитие ИР-систем как самоорганизующихся и адаптивных систем с заранее неизвестными свойствами, которые без участия человека могут формировать и изменять свою структуру и функции при

изменении внешней среды на основе использования средств автоматического синтеза структур ИР-систем, специальных программ-конвертеров (преобразователей) показателей и планирования вычислений;

- децентрализация — использование ориентированной на данные парадигмы к созданию сложных систем, которые строятся из независимых Р-систем-"кирпичиков" на основе анализа потоков данных и установления связей между этими системами за счет автоматического синтеза программ-конвертеров показателей; это согласуется с принципами организации сложных биологических и социальных систем, неограниченная сложность которых достигается за счет компоновки ограниченного числа элементов при отсутствии общих спецификаций всей системы;
- виртуальность — разрабатываемые ИР-системы существуют как единое целое лишь в виде информационной компьютерной модели; функционирование такой системы сводится к вызову отдельных Р-систем в соответствии с моделью;
- конкретность — построение ИР-систем непосредственно под конкретную задачу на основе использования, как уже говорилось, средств автоматического синтеза структур, программ-конвертеров и планирования вычислений;
- жизненный цикл — поддержка полного жизненного цикла ИР-систем (разработка, функционирование, сопровождение, утилизация) как единого сквозного процесса непрерывного развития, в котором чередуются этапы постепенного совершенствования системы и ее функционирования;
- программирование систем — адаптация всего арсенала методов и средств, относящихся к отдельным программам (синтез программ, схемы программ, планирование вычислений, повторное использование программ и т. д.) при разработке сложных комплексов систем;
- оптимизация — во всех необходимых случаях формируется не одна, а множество "близких" альтернатив решения конкретной задачи, из которых выбирается альтернатива, наилучшая по многим критериям оценки;
- прототипность — ИР-системы должны служить основой для создания на их базе семейств систем для других объектов и задач как за счет модификации базовых ИР-систем в направлении построения других систем, так и за счет использования "пустых" ИР-систем-оболочек, состоящих лишь из инвариантных компонентов;

- распределенность — построение ИР-систем с ориентацией на функционирование в распределенной вычислительной среде с учетом топологии и характеристик этой среды.

Инструментальная система ИРИС. Поддержку полного жизненного цикла интегрированных систем, включая разработку, функционирование и развитие, предлагается выполнять на основе интеллектуальной инструментальной системы ИРИС (Инструментарий Разработки Интегрированных Систем) [2, 3, 5].

Комплекс "ИР-система — ИРИС" работает следующим образом. Развитие осваиваемого региона изменяет состояние среды ИР-системы, а воздействия среды подаются на вход ИРИС, формируя задачу функционирования системы. Используя модели базы знаний и спецификацию задачи, ИРИС строит (изменяет) соответствующую информационную графовую модель ИР-системы (см. п. 2) и/или план решения конкретной задачи. Построенная интегрированная система решает соответствующие задачи регионального программирования, что влечет в будущем изменение среды осваиваемого региона и т. д. Таким образом, ИРИС выступает в качестве адаптирующего элемента ИР-системы, включенного в обратную связь между интегрированной системой и моделируемой средой — региональной социально-экономической системой.

В целом ИРИС решает три основные задачи: автоматизации построения ИР-системы, автоматизации сопровождения и повторного использования ИР-системы, автоматизации управления функционированием ИР-системы; при этом первые две задачи фактически сливаются в единую задачу автоматизации непрерывного развития интегрированной системы.

В комплексе "ИР-система—ИРИС" можно выделить следующие взаимодействующие компоненты [2, 3]:

- основные методологические принципы разработки и функционирования ИР-системы ("принципы");
- этапы жизненного цикла ИР-системы, поддерживаемые ИРИС ("этапы");
- методы автоматического синтеза программ, используемые в процессах ИРИС ("методы");
- входная информация для функционирования ИР-системы ("входы");
- лица, осуществляющие разработку, эксплуатацию и сопровождение ИР-системы ("пользователи");
- информационные модели ИР-системы, обрабатываемые ИРИС ("модели");
- процессы обработки информационных моделей ИР-системы ("процессы");

- база знаний, описывающая предметную область, процесс и средства развития ИР-системы ("база знаний");
- средства выбора вариантов в ситуациях неоднозначности ("выбор");
- технологические процессы обработки и анализа полученных моделей ИР-системы ("технологии");
- различные свойства разрабатываемых ИР-систем ("свойства");
- вычислительные средства, на которых работают Р-системы интегрированной системы ("сеть").

Для работы ИРИС не требуется детальных спецификаций разрабатываемых ИР-систем, поскольку:

- развивающаяся система не может быть полностью специфицирована заранее, причем детальные спецификации вступают в противоречие с растущей сложностью систем;
- возможности самоорганизации и адаптации уменьшают актуальность детальных спецификаций.

Инициализация работы ИРИС может осуществляться либо вручную либо автоматически специальным блоком слежения при появлении во внешней среде или во внутренней модели ИРИС различных изменений, которые касаются параметров внешней среды, целей функционирования, множеств входных и выходных показателей ИР-системы, ограничений на предметную область, характеристик вычислительной среды и т. д.

Модели систем. В основе функционирования ИРИС лежит обработка информационных моделей интегрированных систем — графовых моделей систем (ГМС) [2, 3]. Каждая ГМС представляет собой ориентированный граф $G(S, R)$, где вершинам из множества S (элементам ГМС) в общем случае соответствуют Р-системы интегрированной системы, а дугам из множества R — их информационные связи, реализуемые в виде специальных программ-преобразователей технико-экономических показателей (далее конвертеров).

Компонентами ГМС являются элементы, конвертеры, предикаты и указатели на другие ГМС. Если некоторая Р-система s' использует выходные данные Р-системы s'' , $s', s'' \in S$, то от s' к s'' проведена дуга $r \in R$, которой соответствует программа-конвертер, преобразующая показатели из формы для s' в форму для s'' . Каждому элементу ГМС может быть сопоставлен также некоторый предикат, задающий условия его работы.

Несмотря на простоту формальной модели ГМС являются достаточно сложными объектами.

- ◆ Элементы ГМС могут иметь различную содержательную интерпретацию: отдельный программный модуль, программа, Р-система, ссылка на другую ГМС, а также метаэлемент.
- ◆ ГМС могут образовывать многоуровневые иерархические структуры при наличии в них элементов, ссылающихся на другие ГМС.
- ◆ Каждый метаэлемент задает семейство близких в некотором смысле элементов, которые могут быть получены при его конкретизации. Возможны различные способы задания метаэлементов, такие как место для вставки конкретного элемента, ограничения пользователя, элемент-ссылка на некоторую модель, условная конструкция, набор альтернатив и т. д.
- ◆ При наличии в ГМС метаэлементов такие модели систем превращаются в метамодели, описывающие целые семейства близких в некотором смысле ИР-систем.
- ◆ Элементы ГМС могут существовать в различных вариантах (версиях), различающихся, например, особенностями реализации.
- ◆ ГМС могут иметь различные способы представления: фрейм в базе знаний, визуальное представление на экране монитора, текстовое представление на простом языке программирования ГМС, а также представление в виде сети Петри, используемое при управлении работой ИР-систем в распределенной вычислительной среде.

Отдельные Р-системы, составляющие ГМС, могут быть классифицированы с точки зрения следующих свойств:

- общесистемные свойства, такие как стохастичность, динамичность, нестационарность, адаптивность и т. д.;
- пользовательские свойства, задающие классификацию систем по различным направлениям: по типу объекта ИР-системы, в функциональном аспекте (планирование, проектирование, управление и т. д.), по уровням управления, по отраслям, в территориальном и временном аспекте, по характеру решаемых задач и т. д.;
- характеристики реализации, такие как язык программирования, требуемая память, адрес в сети Интернет и т. д.;
- характеристики ГМС как графов, например, цикломатическое число.

Каждый элемент ГМС (Р-система) может быть представлен как точка в n -мерном метрическом пространстве классификационных признаков, где расстояние между точками-системами есть число различающихся значений этих признаков. Это дает возможность исполь-

зовать такие понятия, как расстояние между системами, окрестность системы, базовые системы (прототипы) и т. д. При этом планарный граф ИР-системы $G(S, R)$ может быть заменен на изоморфный и более информативный граф в n -мерном пространстве классификационных признаков, вершины которого соответствуют точкам этого пространства.

Каждый элемент ГМС описывается своим фреймом-паспортом, который содержит, в частности, следующую информацию: тип элемента (программа, Р-система, ссылка на другую ГМС, метаэлемент), входные и выходные данные системы, ссылки на соответствующие программы-конвертеры, предикат применимости, ролевая характеристика элемента, значения пользовательских и общесистемных свойств, характеристики реализации, местонахождение системы в вычислительной среде, характеристики визуализации на экране монитора.

Эти паспорта хранятся в базе знаний (БЗ) системы ИРИС наряду с другими компонентами, описывающими цели функционирования ИР-системы, параметры среды, входные и выходные показатели, конвертеры, предикаты, планы (цепочки) вычислений, вычислительную среду, вычислительные узлы и т. д. Всего БЗ содержит около 40 компонентов, организованных в виде сети фреймов.

Процессы ИРИС. ИРИС функционирует на основе сквозной технологии обработки графовых моделей и других знаний о разрабатываемой ИР-системе с использованием процессной модели [2, 3]: функционирование ИРИС представляет собой совокупность различных взаимодействующих процессов обработки ГМС, обменивающихся сообщениями. Процессы ИРИС разделяются на следующие группы (блоки).

1. Интерфейсный блок обеспечивает взаимодействие ИРИС с пользователем и внешней средой:

- специфицирование — ввод, анализ и сохранение поступающих на вход ИРИС спецификаций задач;
- редактирование — формирование, просмотр и изменение фреймов-паспортов компонентов БЗ;
- визуализация — анализ и редактирование визуальных представлений ГМС;
- мониторинг — анализ и распознавание сигналов среды для поддержки процесса адаптации ИР-систем.

2. Блок обработки моделей проводит различные операции анализа и обработки ГМС:

- операции с моделями — объединение, пересечение, редукция, выделение и конкретизация ГМС;

- поиск — поиск в базе знаний ГМС, удовлетворяющих заданным ограничениям;
- верификация — проверка полноты и корректности связей в ГМС;
- графовый анализ — получение характеристик ГМС как графов;
- документирование — выдача документации по ИР-системе.

3. Блок настройки моделей обеспечивает настройку ГМС на конкретную задачу:

- синтез конвертеров — автоматический синтез программ-конвертеров с помощью средств автоматического синтеза расчетных программ [4];
- планирование вычислений — автоматическое выделение из ГМС плана вычислений (цепочки вызовов Р-систем и программ-конвертеров) для решения конкретной задачи.

4. Блок исполнения обеспечивает функционирование построенной ИР-системы, в том числе в распределенной вычислительной среде:

- исполнение — интерпретация плана вычислений, т. е. собственно функционирование ИР-системы;
- синхронизацию — динамическое назначение Р-системам и конвертерам вычислительных узлов при распараллеливании работы ИР-системы в распределенной вычислительной среде.

5. Блок процессов развития: семь процессов этого блока более подробно рассмотрены ниже в п. 3.

В процессе планирования вычислений автоматически строится план решения конкретной задачи нахождения целевых показателей работы ИР-системы V по заданным показателям U ; этот план представляет собой линейную последовательность Р-систем и конвертеров, упорядоченных в порядке их вызова. Из имеющейся ГМС выделяется подграф для решения этой задачи, т. е. проводится редукция тех элементов ГМС и конвертеров, которые не используются для решения данной задачи. Планирование на ГМС выполняется аналогично планированию на вычислительных моделях [6] и может осуществляться либо "снизу вверх" от U к V , либо "сверху вниз" от V к U . В процессе планирования также могут проверяться и сниматься предикаты применимости, зависящие от параметров среды, и использоваться ограничения пользователя на вид планов.

Блок исполнения ИРИС вызывает и анализирует план вычислений, проверяет предикаты применимости Р-систем и осуществляет вызов компонентов ГМС — отдельных Р-систем и программ-конвертеров как ехе-файлов. Функционирование ИР-системы может проводить-

ся в распределенной вычислительной среде, включая многопроцессорные системы, кластеры, территориально-распределенные гетерогенные грид-системы. В этом случае блок исполнения позволяет динамически распределять исполнение компонентов ГМС между свободными вычислительными узлами, запускать компоненты одного уровня плана параллельно на различных узлах и синхронизировать исполнение компонентов.

2. Автоматический синтез расчетных программ

Технико-экономические показатели и их преобразования. Для построения программ-конвертеров в процессе разработки ИР-систем используются средства автоматического синтеза расчетных программ (РП) — программ обработки технико-экономических показателей (ТЭП) [4]. Следует отметить, что РП составляют значительную часть программного обеспечения автоматизированных систем различного назначения, включая Р-системы. Каждый ТЭП характеризуется своим наименованием t и значением \bar{t} (вектор либо матрица чисел). Совокупность ТЭП для данной предметной области можно рассматривать как некоторый информационный язык фасетного типа [7].

Каждый фасет определяет некоторую ролевую характеристику языка ТЭП и задает позицию в тексте ТЭП, которая может быть заполнена некоторым термином из заданного множества значений фасетов. Пример ТЭП (в скобках указаны названия фасетов): "прирост (способ расчета) плановой (функция) численности (характеристика) работающих (предмет) по годам (разрез) в тыс. (коэффициент) человек (единица измерения)".

Вводится n -мерное семиотическое гиперпространство наименований ТЭП $T = \prod_{i=1}^n T_i$, где T_i — множество значений i -го фасета; n — число фасетов [4]. Тогда каждому ТЭП будет соответствовать точка $t \in T$ с координатами (t^1, t^2, \dots, t^n) , $t^i \in T_i$. Пространство T можно рассматривать как метрическое пространство, где расстояние $d(x, y)$ между двумя ТЭП $x, y \in D$ — число несовпадающих значений фасетов в их наименованиях; это позволяет использовать, в частности, понятия расстояния от точки до множества, шара, связного множества и т. д.

Особенности расчетных программ определяются спецификой языка ТЭП. Каждая РП представляет собой линейную последовательность программных фрагментов (вычислительных преобразований, ВП), которые сво-

дятся к циклическому выполнению поэлементных преобразований матриц — значений ТЭП. Каждому ВП значений ТЭП вида $f: \bar{X} \rightarrow \bar{y}$ соответствует некоторое семантическое преобразование (СП) их наименований $s: X \rightarrow y, X \subseteq D, y \in D$. Вводятся различные классы СП и показываются их взаимные связи [4]. Так, каждому правильному СП из класса R соответствует некоторое ВП; класс E элементарных СП включает СП из R , неразложимые на другие СП из R . Важный класс B базовых СП (БСП) включает СП из E , аргументами которых могут быть только семантически близкие (различающиеся в одном фасете) ТЭП.

Назовем фасеты определяющими для данного СП из E , если они задают семантику преобразования и однозначно определяют вид сопоставленного ВП. Систему фасетов назовем независимой, если любое СП из E имеет единственный определяющий фасет. Можно показать, что для независимых фасетов

$$B \equiv E = \bigcup_{i=1}^n B_i, B_i = \bigcup_{k=1}^{m_i} B_{ik},$$

где эквивалентные БСП B_{ik} изменяют i -й фасет одинаковым образом. Поэтому каждому B_{ik} можно сопоставить единственное базовое преобразование значений фасетов (БФП), заданное на T_i , и единственное базовое ВП (БВП) значений ТЭП, что резко уменьшает число связей ТЭП.

БВП обычно реализуют матричные арифметические и структурные операции над матрицами и векторами. Так, множеству B_{ik} , включающему БСП {"план добычи газа", "объем добычи газа"} \rightarrow "дефицит добычи газа", соответствует БФП {"план", "объем"} \rightarrow "дефицит" и БВП — поэлементное вычитание матриц.

В работе [4] было показано, что зависимые системы фасетов обычно можно преобразовать в независимые, исключив зависимые фасеты, не являющиеся определяющими ни для одного СП из E , или проведя редукцию множеств значений фасетов.

Все сказанное выше позволяет поставить задачу синтеза расчетных программ. Пусть заданы: граф $G(A, R)$ — вычислительная модель, где множеству вершин A соответствует объединение множеств допустимых ТЭП из D и их семантических отношений; входные ТЭП U , их значения \bar{U} и выходные ТЭП $V (U, V \subset D)$. Требуется построить РП $P: \bar{U} \rightarrow \bar{V}$, выделив из соответствующих отношений последовательность операторов для расчета значений \bar{V} по \bar{U} .

Теорема. Пусть для пространства ТЭП фасеты являются независимыми, и пусть U —

множество ТЭП, для которых известны их значения \bar{U} . Тогда семантическое преобразование $s: X \rightarrow y, X \subseteq U, y \in D \setminus U$, есть композиция БСП тогда и только тогда, когда $y \in U^*$, где U^* — множество достижимости для U .

Как следствие, при независимости фасетов семантическая модель программы $S: U \rightarrow V$, соответствующая расчетной программе $P: \bar{U} \rightarrow \bar{V}$, есть композиция БСП, меняющих по очереди значения различных фасетов.

Метод и алгоритмы синтеза расчетных программ. Метод синтеза РП [4] существенным образом использует следующие специфические особенности ТЭП и их преобразований: можно количественно оценить семантическую близость ТЭП, установить соответствие между СП наименований и ВП значений ТЭП, причем аргументами БСП могут быть только семантически близкие ТЭП. А сам метод синтеза РП опирается на комплексное использование идей таких методов синтеза программ, как программирование в потоках данных, композиционное программирование и синтез на вычислительных моделях [6].

Рассмотренные свойства ТЭП и их преобразований позволяют обеспечить автоматический синтез программ расчета показателей. Автоматический синтез РП проводится в два этапа: вначале строится семантическая модель программы, затем по ней генерируется текст программы на конкретном языке программирования. Это позволяет учитывать специфику ТЭП и их преобразований, проводить синтез в терминах формальных объектов, оптимизировать РП и строить их на различных выходных языках.

На основании теоремы решение задачи синтеза семантической модели преобразования $s: X \rightarrow y$ можно искать как линейную последовательность (композицию) абстрактных однородных компонентов — БСП, переходя затем к соответствующим вычислительным реализациям — БВП. Анализируется множество БСП, через которое "плывет" множество наименований ТЭП. Если имеется необходимое число семантически близких ТЭП, являющихся аргументами некоторого БСП, то данное БСП активизируется (вырабатывая наименование нового ТЭП) и добавляется к решению.

Семантическая модель РП строится в виде частично упорядоченной в порядке применения БСП последовательности дескрипторов БСП $\langle X, y, f, k \rangle$, где $f: \bar{X} \rightarrow \bar{y}$, а k — уровень вычислений. Следует отметить, что БВП и БСП представляют собой атомарные элементы РП и их семантики, выделение которых обеспечивает многоуровневую декомпозицию про-

грамм, их семантических моделей и значительно уменьшает сложность исходной задачи синтеза, позволяя искать оптимальное решение.

Алгоритм синтеза семантических моделей программ $S: U \rightarrow V$, оптимальных по сложности вычислительных операций, работает в два этапа [4]. Вначале определяется разрешимость исходной задачи, и строится явный граф G' с вершинами из U_0^* , $V \subseteq U_0^* \subseteq U^*$, где U^* есть множество достижимости для U ; одновременно формируется семантическая модель РП $S': U \rightarrow U_0^*$. Для поиска решения используется перебор на графе в ширину в прямом направлении [12]. Для каждой вершины z графа G' находится и запоминается оценка стоимости $g(z)$ ее достижения, т. е. оценка поддерева G' с корнем в z . Алгоритм заканчивает работу за линейное время $N = O(|U^*|)$.

На втором этапе из S' выделяется оптимальное решение — модель искомой РП $S: U \rightarrow V$ с минимальной сложностью вычислительных операций. Для этого граф G' просматривается в обратном порядке, от V к U , и на нем выделяется решающий подграф минимальной стоимости G'' , такой что его конечным вершинам соответствуют только ТЭП из V . Алгоритм упорядоченного эвристического перебора на И/ИЛИ графах [8] всегда находит оптимальный решающий подграф, так как для любой вершины z выполняется условие $g(z) \leq g^*(z)$, где $g^*(z)$ — действительная стоимость для z . Работа алгоритма упрощается за счет того, что существование графа G'' следует из разрешимости задачи, и этот граф ищется на заранее построенном графе G' с известными оценками вершин $g(z)$.

На этапе генерации формируется текст расчетной программы на заданном языке программирования. Вначале по семантической модели РП, сведениям о предметной области и задаче синтеза строится детальная модель РП в виде совокупности дескрипторов основных ее компонентов — типов размерности, переменных, массивов, операторов обмена с информационной базой, циклов, расчетных формул и т. д. Одновременно проводится частичная оптимизация генерируемой РП по памяти и быстрдействию — минимизируется число обменов с базой знаний и циклов.

На втором этапе генерации непосредственно строится текст РП на некотором подмножестве языка программирования блочного типа; сведения о языке задаются правилами вывода контекстно-свободной грамматики языка. В процессе генерации при раскрытии правил вывода срабатывают сопоставленные им семантические правила, которые, извлекая ин-

формацию из детальной модели программы и базы знаний, формируют значения семантических атрибутов, определяющих конкретный вид порождаемой РП (включая идентификаторы, имена файлов, вид БВП и т. д.).

Рассмотренные методы и алгоритмы были реализованы в Системе автоматического синтеза расчетных программ (САС/РП) [4], которая позволяет автоматически формировать программы расчета технико-экономических показателей на заданном языке программирования. Для постановки задачи синтеза требуется минимальная информация — тексты входных и выходных показателей расчетной программы.

3. Автоматизация разработки интегрированных систем

Синтез программ-конвертеров. Автоматизация разработки и сопровождения ИР-систем проводится средствами ИРИС. При построении ИР-систем из независимых Р-систем следует обеспечить информационные связи между их входами и выходами. В различных системах данные с одинаковой семантикой могут представляться по-разному, поэтому для каждой пары систем s_1 и s_2 таких, что $In(s_2) \cap Out(s_1) \neq \emptyset$, необходима программа-конвертер, преобразующая часть выходных данных $Out(s_1)$ системы s_1 (из формы для s_1) во входы $In(s_2)$ системы s_2 (в форму для s_2) [2, 5].

Для автоматического синтеза таких программ необходимо, чтобы обрабатываемыми данными были технико-экономические показатели с единым словарем значений фасетов. Назовем ТЭП x_1 и x_2 семантически эквивалентными, $x_1 \approx x_2$, если их наименования имеют одинаковые значения для определяющих семантику ТЭП фасетов, хотя могут иметь различные значения в служебных фасетах, таких как "функция", "разрез", "единица измерения" и т. д.

Для каждой пары Р-систем s_1 и s_2 программа-конвертер есть совокупность вычислительных преобразований показателей $f: \bar{x}_1 \rightarrow \bar{x}_2$ для каждой пары ТЭП x_1 и x_2 , таких что $x_1 \approx x_2$, $x_1 \in Out(s_1)$, $x_2 \in In(s_2)$. Тогда при использовании автоматического синтеза программ [4] каждое такое преобразование f строится как композиция БВП, а соответствующая семантическая модель $s: x_1 \rightarrow x_2$ — как композиция БСП, таких что каждое из них меняет значение некоторого единственного фасета с номером j в наименованиях ТЭП x_1 и x_2 .

Таким образом, автоматическое построение конвертеров обеспечивает интеграцию разно-

родных Р-систем в единое целое и поддержку постоянного развития (самоорганизации) сложных ИР-систем.

Процессы развития интегрированной системы.

Процессы развития реализуются через процессы обработки ГМС и других компонентов базы знаний средствами ИРИС для решения задач построения и сопровождения развивающихся ИР-систем. Рассматриваются следующие процессы развития ИР-систем [2, 3].

Автоматический синтез моделей обеспечивает автоматическое построение структуры ГМС за счет установления связей по данным между несвязными элементами множества Р-систем S , т. е. фактически обеспечивает самоорганизацию ИР-системы. Автоматический синтез конвертеров проводится на основе анализа семантики наименований показателей, обрабатываемых Р-системами из S , следующим образом:

- для каждой пары Р-систем $s_1, s_2 \in S$ попарно сравниваются все ТЭП из $Out(s_1)$ и $In(s_2)$;
- для каждого ТЭП $x_1 \in Out(s_1), x_2 \in In(s_2)$ пофасетно сравниваются их наименования для проверки $x_1 \approx x_2$;
- при $x_1 \approx x_2$ информация о получении x_2 по x_1 добавляется к модели конвертера I для систем s_1 и s_2 ;
- если после окончания всех сравнений модель I не пуста, то s_1 и s_2 соединяются дугой на ГМС и выполняется автоматический синтез расчетной программы-конвертера, обеспечивающей преобразование показателей из формы системы s_1 в форму системы s_2 .

Если $|S| = n$, а среднее число входов/выходов систем из S есть m , то требуется провести $O(n^2m^2)$ сравнений наименований показателей. Поэтому для сокращения перебора могут использоваться роли Р-систем в ИР-системе, ограничения пользователя, информация самообучения и т. д.

Автоматическая модификация моделей выполняется при сопровождении, развитии, адаптации ИР-систем к изменениям среды и новым задачам, а также при использовании ИР-систем в качестве прототипов при разработке новых систем. При модификации ГМС могут изменяться состав, связи, характеристики и интерпретация компонентов модели и обрабатываемых данных. Автоматическая модификация ГМС сводится к их повторному автоматическому синтезу и инициируется автоматически резидентным блоком. Этот блок работает в режиме слежения и периодически сравнивает текущее состояние входов ИРИС с тем состоянием, при котором была построена модифицируемая модель.

Диалоговое редактирование моделей обеспечивает построение/модификацию ГМС с использованием средств машинной графики. Пользователь в диалоговом режиме может добавлять, заменять и удалять компоненты ГМС, просматривать их характеристики, объединять, разделять ГМС, формировать метамоделли и т. д.

Задание текстовых описаний моделей может проводиться на простом языке программирования ГМС декларативного типа. Для метаэлементов могут быть заданы ограничения на класс элементов, условные выражения и наборы альтернатив.

Обобщение моделей обеспечивает получение из нескольких ГМС модели более общего вида (метамоделли), которая в будущем может быть использована в качестве базы (прототипа) при разработке новых моделей и модификации уже существующих. После формирования метамоделей исходные модели можно удалить.

Конкретизация метамоделей (моделей с метаэлементами) позволяет строить семейства ГМС с идентичной структурой за счет конкретизации метаэлементов до элементов. Конкретизация каждого метаэлемента может быть задана пользователем, выбрана им среди нескольких вариантов, получена автоматически в зависимости от параметров среды и т. д.

Утилизация моделей обеспечивает повторное использование ранее созданных ГМС, их фрагментов (подграфов) и компонентов (конвертеров, предикатов) при создании моделей новых ИР-систем. Проблемы повторного использования во многом снимаются при использовании формализованных описаний ГМС, что обеспечивает однозначную идентификацию, эффективное хранение и поиск повторно используемых фрагментов.

Заключение

Особенность предложенного подхода состоит в том, что он обеспечивает самоорганизацию и адаптацию сложных программных комплексов, состоящих из независимых систем за счет автоматического синтеза программ-конвертеров, обеспечивающих информационные связи между этими системами. Интегрированная система строится непосредственно под конкретную задачу и существует как единое целое только виртуально, в виде компьютерной модели, что снижает трудоемкость разработки ИР-системы и повышает эффективность ее функционирования. Полученные результаты могут быть использованы не только при раз-

работке ИР-систем, но и при создании производственных комплексов систем, обрабатывающих технико-экономические показатели.

Список литературы

1. Хачатуров В. Р. Математические методы регионального программирования. М.: Наука, 1989. 304 с.
2. Хачатуров В. Р., Соломатин А. Н., Злотов А. В. и др. Планирование и проектирование освоения нефтегазодобывающих регионов и месторождений: Математические модели, методы, применение / Под ред. В. Р. Хачатурова. М.: УРСС: ЛЕНАНД, 2015. 304 с.
3. Соломатин А. Н. Автоматизация разработки интегрированных региональных систем. М.: ВЦ РАН, 1993. 32 с.

4. Соломатин А. Н. Автоматический синтез расчетных программ на основе анализа семантики показателей // Кибернетика и системный анализ. 1994. № 5. С. 133–142.
5. Solomatin A. N. Integrated systems of regional programming // Tenth International Conference "Management Of Large-Scale System Development" (MLSD) / Moscow (October 2017). IEEE Conference Publications, IEEE Xplore Digital Library. URL: <http://ieeexplore.ieee.org/document/8109689> (дата обращения 12.08.2019).
6. Тыгу Э. Х. Решение задач на вычислительных моделях // ЖВМ и МФ. 1970. Т. 10, № 5. С. 716–734.
7. Главатская Н. Г., Петухова О. В., Ясин Е. Г. Базовый информационный язык — система формализованного описания экономических данных // Прикладная информатика. М.: Финансы и статистика, 1983. С. 35–48.
8. Нильсон Н. Проблемы искусственного интеллекта. М.: Радио и связь, 1985. 280 с.

A. N. Solomatin, Ph.D., Leading Researcher, e-mail: a.n.solomatin@bk.ru,
Federal Research Center "Computer Science and Control" of Russian Academy of Sciences,
Moscow, Russian Federation

The Integrated Systems of Regional Programming and Automation of their Development

Features and automation of development of the integrated systems of regional programming (IR-systems) are considered. IR-systems are the complexes of the systems consisting of separate independent systems (R-systems) of regional planning, design, ecological safety and intended for solving problems of complex development of territories. Each integrated system exists as a whole only virtually in the form of a computer model and is built each time for a specific task by automatically establishing information links between P-systems. Support for the full life cycle of IR-systems, including the development and operation is provided by the IRIS tool system. Models of IR-systems processed by IRIS and different groups of IRIS processes (operating on the basis of the process model) are described. The concept, method, and algorithms of automatic synthesis of calculation programs — programs for calculation of technical and economic indicators are discussed. The hyperspace of names of indicators is introduced, various classes of semantic transformations of names are introduced and investigated. The method and algorithms of synthesis of semantic models of programs as a sequence of semantic transformations of special form are described, generation of program text on some subset of a block-type programming language is considered. The issues of automation of the development of IR-systems using IRIS tools are discussed. To build a model of an IR-system for solving a specific problem by means of existing P-systems, information links between systems are automatically established based on the analysis of the semantics of the processed indicators. And then, using a system of the automatic synthesis of calculation programs converter programs are generated that implement these relationships and transform semantically equivalent indicators of P-systems from one form of presentation to another. The development processes of IR-systems implemented by IRIS are listed, including processes of automatic synthesis and modification of models of IR-systems, generalization, utilization of models, etc. The proposed approach can be used to automate the development of fairly arbitrary systems processing technical and economic indicators.

Keywords: complex development of territories, integrated systems of regional programming, planning and design systems, systems complex model, process model of functioning, planning of calculations, distributed functioning environment, automatic synthesis of calculation programs, indicator converters, development automation tools

DOI: 10.17587/it.26.107-115

References

1. Khachaturov V. R. Mathematical methods of regional programming, Moscow, Nauka, 1989, 304 p. (in Russian).
2. Khachaturov V. R., Solomatin A. N., Zlotov A. V. et al. Planning and designing the development oil and gas production regions and fields: Mathematical methods, models, applications, Moscow, URSS, LENAND, 2015, 304 p. (in Russian).
3. Solomatin A. N. Automation of development of the integrated regional systems, Moscow, Published CC RAS, 1993, 32 p. (in Russian).
4. Solomatin A. N. Automatic synthesis of calculation programs based on the analysis of semantics of indicators, *Cybernetics and System Analysis*, 1994, no. 5, pp. 133–142 (in Russian).

5. Solomatin A. N. Integrated systems of regional programming, *Tenth International Conference "Management Of Large-Scale System Development" (MLSD) / Moscow (October 2017)*, IEEE Conference Publications, IEEE Xplore Digital Library, available at: <http://ieeexplore.ieee.org/document/8109689> (date of access: 12.08.2019).
6. Tyugu E. H. The solution of problems on computational models, *ZhVM and MF*, 1970, vol. 10, no. 5, pp. 716–734 (in Russian).
7. Glavatskaya N. G., Petukhova O. V., Yasin E. G. Basic information language — the system of formalized description of economic data, *Applied Informatics*, Moscow, Finance and statistics, 1983, pp. 35–48 (in Russian).
8. Nielson N. Problems of artificial intelligence, Moscow, Radio and communication, 1985, 280 p. (in Russian).

Ю. Б. Кузьмин, аспирант, e-mail: info@data-analytic.org,
ИПУ РАН, Москва

Шкалирование уровня автоматизации на производстве

Статья посвящена обзору проблемы оценки уровня автоматизации в зарубежной и отечественной литературе. Даны классификация задач автоматизации и сравнительная характеристика некоторых шкал — их сходство и различие. Приводится ряд стандартов и методов определения уровня автоматизации в сравнении с классической шкалой Т. Шеридана и В. Вепланка, с другим типом оценки — вектором приоритетов коэффициента автоматизации K_a . Обсуждается уровень автоматизации для систем технологических процессов, бизнес-процессов, подготовки производства и др. Рассматривается принцип объединения шкал с различными классами задач автоматизации и дальнейшее исследование в целях оптимизации уровня автоматизации в зависимости от стоимости и других показателей производства. Отмечена возможность дальнейшего исследования при формализации различных процессов производства и учета уровня автономности динамических систем.

Ключевые слова: уровень автоматизации, LOA, коэффициент автоматизации, степень автоматизации, шкала автоматизации, шкала Т. Шеридана и В. Вепланка, классификация автономности

Введение

Проблема оценки уровня автоматизации недостаточно раскрывается на страницах русскоязычной печати, хотя в иностранной литературе этому вопросу уделяется немало внимания в журналах, монографиях и книгах. Существует постоянное развитие этого понятия, начиная с середины прошлого века, когда Р. М. Fitts в 1951 г. обратил внимание на распределение задач между человеком, машиной и компьютером — отмечается в обзорной работе "Levels of Automation in Manufacturing" [1]. С 1978 г. за рубежом получила широкое распространение десятиуровневая шкала Т. Шеридана и В. Вепланка (ШиВ) [6], также другими исследователями рассматриваются шкалы как с более, чем десятью уровнями, так и с меньшим числом уровней [1]. Немаловажным моментом является создание новых методик и подходов для расчета уровня автоматизации в различных сферах деятельности. В настоящий момент некоторые организации, одна из них международная — сообщество автомобильных инженеров (Society of Automotive Engineers, SAE), уже стандартизовали шесть уровней автономности (Standard J3016) [2], так как понимание уровня автоматизации (уровня автономности) опреде-

ляет класс выпуска автоматизированных или автоматических машин, это становится неотъемлемой частью производства для автомобилестроения, аэрокосмической индустрии и др.

Среди стандартов также есть упоминания об уровне автоматизации при проектировании (в ГОСТ 23501.108—85) и об уровне автоматизации технологических процессов в ГОСТ 23004—78 (ГОСТ 14.309—74 — старая версия) [7, 14], где приведены числовые оценки для восьми уровней и даны краткие методические пояснения по расчету. В отечественной литературе часто встречается понятие коэффициента автоматизации [17, 18].

В РФ в связи с уменьшением производства, отсутствием комплексного перевооружения предприятий и, как следствие, отсутствием аудита программно-технических средств (ПТС) и исследований автоматизации с количественными результатами такие оценки и методики не очень востребованы. В академической среде такие исследования не прекращались, так как вопросы модернизации на основе оценки текущего и планируемого состояния систем подразумевают эффективное использование ресурсов, более четкую постановку целей и, в конечном итоге, конкретное отражение изменения заданных показателей с обоснованием

повышения эффективности. Для системы автоматизированной технологической подготовки производства (ТПП) российские исследователи предлагают метрики в пространстве автоматизации (ПА) по трем координатам, позволяющие всесторонне оценивать проект системы по числу автоматизируемых задач, объемам видов обеспечения и этапам проекта [3]. Для систем управления производством в общепринятых нотификациях применяется оценка уровня автоматизации бизнес-процессов [4], в работе [12] приводится описание информационной модели АСУ ТП в диаграммах UML-нотаций [12]. Для оценки уровня автоматизации иерархических систем управления, в частности на производстве для систем АСУ ТП, используется расчетный показатель K_a [5], а для автоматизированных информационных систем (АИС) — интегральный показатель [24].

В настоящее время отраслевыми холдингами часто применяются свои разработанные стандарты, но, как правило, решения о выборе автоматизированной системы принимаются на интуитивном уровне при рекламной поддержке производителями своих разработок, а не на основании сравнительных характеристик предлагаемых продуктов. Формальные методы оценки с применением ранжирования, а тем более количественные шкалы — интервальные, отношений и др. — используются крайне редко. Хотя шкалы с более возможными допустимыми (математическими) преобразованиями отражают более высокий уровень измерений и учитывают не только то, в каком отношении находятся измеряемые объекты, но и степень их различия [22, 23]. Тип шкалы также предопределяет, какие методы статистического анализа можно будет использовать [23]. Более того, такого рода градации используют как атрибуты для векторизации данных и дальнейшей обработки.

Поэтому при приближении хотя бы в перспективе производства к цифровым технологиям СЮ-руководителям и ИТ-менеджерам необходимо иметь первоначальный инструмент оценки состояния и влияния на уровень автоматизации как программно-технических средств с описанием когнитивных задач (человек—компьютер), так и механического/технического оборудования с описанием физических задач (человек—машина). Объединение этих понятий в данной работе при шкалировании уровня автоматизации будет являться продолжением работы в этом направлении.

Шкала уровня автоматизации (LOA)

Шкалирование, или оценка уровня автоматизации Level of Automation (LOA), как отмечается в работах [1, 10], исследуется с 50-х годов прошлого столетия многими авторами и основные концепции представлены в работах Т. В. Sheridana, Н. Kerna & М. Schumanna, С. E. Billingsa, М. R. Endsley & D. B. Kabera, Р. Satchella, R. Parasuramana, М. P. Groovera, Т. J. Williamsa.

Уровень автоматизации (LOA) — это распределение физических и когнитивных (информационных) задач между людьми и техникой, описываемое как непрерывный процесс от полностью ручного до полностью автоматизированного [1]. При количественном подсчете уровень автоматизации можно охарактеризовать как формализованную, выраженную числом степень роботизации выполняемых операций или, по-другому, независимость автоматизированного комплекса от человека [4].

В своей работе 1978 г. Т. Шеридан и В. Веспланк (Массачусетский технологический институт) [6] предложили ставшее классическим 10-уровневое шкалирование по уровню компьютеризации и оценивали степень автоматизации в зависимости от того, кто принимает решение, вида операции и того, кто ее выполняет — человек или техническая система/компьютер (табл. 1) [1, 4]. Каждая операция по Т. Шеридану и В. Веспланку (ШИВ) может выполняться в абсолютно ручном режиме человеком — уровень автоматизации "1" или полностью автоматически компьютером (роботизированный режим), без участия человека — уровень автоматизации "10". Промежуточные уровни автоматизации оцениваются от 1 до 10.

Систематизация ШИВ включает вопросы обратной связи, определение и выбор вариантов, связанных с распределением задач между человеком и технической системой, компьютером. Эта шкала также является одной из наиболее подробных классификаций, которые можно найти в литературе с точки зрения определения того, **что** оператор (человек) и техническая система (компьютер для информации и контроля) должны выполнять на разных уровнях LOA, и **как** они должны взаимодействовать [1].

Чтобы сделать производственную систему максимально надежной, необходимо решить вопрос не только о том, как определить правильное функционирование элементов системы при решении определенных задач, но и о том, какие человеческие ресурсы и автомати-

Таблица 1

10 уровней автоматизации (LOA)
по Т. Шеридану и В. Вепланку [1]

LOA	Описание
10	Компьютер решает все и действует автономно, игнорируя человека
9	Компьютер выполняет операции автоматически и информирует человека, только если он (компьютер) решит
8	Компьютер выполняет операции автоматически и информирует человека, только если компьютер "спросит"
7	Компьютер выполняет операции автоматически, обязательно информируя человека
6	Компьютер предлагает одну альтернативу и выполняет это предложение, если человек в течение ограниченного времени не накладывает вето на автоматическое выполнение операции
5	Компьютер предлагает одну альтернативу и автоматически выполняет это предложение, если человек соглашается
4	Компьютер предлагает одну альтернативу
3	Компьютер предлагает полный набор решений/действий, альтернативы и сужает выбор до нескольких вариантов
2	Компьютер предлагает человеку полный набор решений /действий, альтернативы
1	Компьютер не предлагает помощь: человек должен принимать решения и выполнять все действия сам

зированные элементы могут поддерживать друг друга на разных уровнях автоматизации, чтобы сделать производственную систему максимально удобной и гибкой [1]. К примеру, в случае предоставления оператору ручных или автоматических инструментов (например, гаечного ключа или гидравлического болта) повышается уровень технической поддержки. При дальнейшей замене электрического или гидравлического ручного инструмента машиной или роботом можно достигнуть почти полной автоматизации. Важно понимать, что концепция оценки уровня автоматизации лежит в плоскости понятий: техника—человек—компьютер и учитывает степень их взаимодействия при распределении функций (что делает человек, что машина и как осуществляется контроль).

Автоматизацию в производстве можно разделить на два основных класса: механизацию и компьютеризацию [1, 10]. Под компьютеризацией понимается решение когнитивных (познавательных) задач, возникающих в результате чувственных процессов и умственной деятельности человека. Примером могут быть сбор, хранение, анализ и использование информации для управления производственным

процессом. Механизация аналогично связана с компьютеризацией, но определяется как замена физического труда человека механизмами с преобразованием материала и энергии, а уровень механизации можно определить как технический уровень производственной системы [1]. Механический/физический уровень LOA определяет, **чем** производить, собирать, обрабатывать и т. п., в то время как когнитивный уровень LOA — **как** производить (на нижних уровнях) и как управлять ситуацией (на более высоком уровне) [10]. Рост уровня автоматизации возрастает по мере того, как все больше задач выполняется автоматически. Поэтому по мнению западных исследователей [1, 10] представляется целесообразным выделить следующие уровни автоматизации:

а) автоматизация физических задач, связанных с технологией производства;

б) автоматизация информационных задач контроля, мониторинга и поддержки принятия решений;

в) автоматизация при комбинировании задач (совместное распределение) между машинами/технологией и информированием/контролем человеком.

В РФ не так давно, согласно ГОСТ 23004—78 [7], как уже упоминалось, выделяли восемь категорий механизации и автоматизации технологических процессов в зависимости от диапазона изменений временного уровня механизации средств технологического оснащения (СТО) $K_{\text{СТО}}$ (табл. 2). Расчет базировался на отношении машинного времени ко всему штучному времени (включает основное, вспомогательное

Таблица 2

Категории механизации и автоматизации технологических процессов согласно ГОСТ 23004—78

Номер категории	Наименование категории	Диапазон изменений временного уровня механизации (автоматизации) СТО
0	Нулевая	0 — при отсутствии механизации (автоматизации) процесса
1	Низшая	0,0...0,25
2	Малая	0,25...0,45
3	Средняя	0,45...0,60
4	Большая	0,60...0,75
5	Повышенная	0,75...0,90
6	Высокая	0,90...0,99
7	Завершенная	1,00

и время на обслуживание и отдых). Интерес в этом стандарте представляет тот факт, что устанавливалась взаимосвязь единства двух подсистем: качественных и количественных характеристик.

Очевидно, что рост уровня автоматизации связан и коррелирует с повышением технической поддержки или механического оснащения и с добавлением средств управления. Техническое оснащение автоматизированной производственной системы, к примеру АСУ ТП, которая управляет механическими устройствами технологического процесса (ТП), тесно связано с программно-техническим обеспечением (ПТО) и программно-техническими средствами (ПТС) — функциональностью и числом контрольно-измерительных приборов и автоматики (КИПиА), которые непосредственно связаны с оборудованием. Какая это связь, и как они коррелируют? Вопрос, который предстоит исследовать.

Следует отметить, что некоторые западные специалисты базируются в своих оценках на экспертных мнениях, основанных на шкалировании и описании когнитивных задач и их связи с физическим трудом [1, 10]. В отечественной литературе часто встречается понятие коэффициента автоматизации, либо используют другой расчетный показатель, на основе которых делается вывод о различных уровнях автоматизации [7, 17, 18]. Из этого можно сделать вывод — существуют два подхода, которые рассматривают проблему с разных сторон: учет степени участия человека путем экспертных оценок по определенным градациям (уровням), либо учет машинного времени (самого распространенного показателя) по отношению ко всему времени, либо какого-либо другого функционального показателя.

Связь интегрального коэффициента уровня автоматизации K_a и LOA

В работе [5] уровень автоматизации для систем типа АСУ ТП ассоциирован как с уровнем (функциональностью) ПТС, так и со степенью автоматизации функций контроля и управления на основе сигналов, поступающих непосредственно с агрегатов. Представление о степени автоматизации таких функций основано на учете числа сигналов и задействованных приборов и средств автоматики. Эти параметры нашли свое отражение в интегральном коэффициенте автоматизации согласно формуле

$$K_a = \sum_l v^l K_a^l, \quad (1)$$

где v^l — вектор приоритетов (весов) для l -уровней, $v = (v^0, v^1, \dots, v^l)$, $\sum_l v^l = 1$; K_a^l — с одной стороны отражает для всех уровней функциональность ПТС, ПО и технологичность решений, и с другой, только для 0-го уровня — степень полноты автоматизации, т. е. доли сигналов для контроля и управления от возможного числа [5]. Подобная формула для среднего интегрального показателя уровня автоматизации, основанная на оценках ИТ-аспектов АИС, приводится в работе [24].

Таким образом, была формализована концепция оценки уровня автоматизации в количественном виде от 0 до 100 % с учетом функциональности применяемых ПТС и числа функций контроля/управления.

Если говорить подробнее, то были получены численные оценки вклада в уровень автоматизации для АСУ ТП, построенной по четырехуровневой иерархии: 0-й уровень — уровень КИПиА; 1-й уровень — уровень ПЛК; 2-й уровень — уровень диспетчерского управления и сбора данных (уровень SCADA от англ. Supervisory Control and Data Acquisition); 3-й уровень — уровень системы исполнения производства (MES от англ. Manufacturing Execution System). Среднее значение вектора приоритетов уровней $v_{cp} = (0,53; 0,26; 0,13; 0,08)$ [5]. Диапазон изменения вклада всех уровней в общий показатель автоматизации был равен: 0-го уровня (КИПиА) — 50...56 %; 1-го (ПЛК) — 23...29 %; 2-го (SCADA) — 12...15 %; 3-го (MES) — 6...10 %. Значения вектора приоритетов были получены в работе [5] по двум методикам: уровневой модели весовых коэффициентов, предложенной в работе, и методом анализа иерархий (МАИ) Т. Саати. Эти значения соответствовали, по мнению экспертов, идеализированному уровню — самой высокой степени вклада каждого уровня автоматизации, если бы они оставались без изменения в окончательной формуле (1) для K_a . Другими словами, идеализированная оценка ПТС и число автоматизируемых функций контроля и управления технологических элементов, связанных с КИПиА, отвечает самому высокому уровню автоматизации при использовании в системе управления последних достижений науки и техники, что также косвенно подтверждается увеличением возлагаемых функций на ПТС, которые ранее мог выполнять человек (этот уровень в работе [5] указан как перспективный).

При другом подходе, как упоминалось в предыдущем разделе, согласно стандарту [7] рассматривается коэффициент $K_{СТО}$ как доля машинного времени от всего штучного времени, который принимает значение также от 0 до 1 (0—100 %). Иными словами, вычисляется сразу прямая величина относительно распределенного времени, связанная с уровнем средств технологического оснащения для каждой операции, учитываемой при автоматизации. К примеру, для категорий "повышенного" и "высокого" уровней автоматизации доля машинного времени $K_{СТО}$ должна составлять 75...90 % и 90...99 % соответственно (табл. 2).

Интересно было сопоставить численные значения коэффициентов K_a с градациями LOA шкалы ШИВ и с описанием в стандарте уровней автоматизации. Для этого мы привели шкалу ШИВ к значениям от 0 до 1 и записали степень уровня автоматизации, взяв десятичный логарифм $\lg(LOA)$ от номера LOA, преобразовав ранговую (порядковую) шкалу в относительную (интервальную). Это означает, что мы рассматриваем отношение текущего уровня автоматизации к уровню, когда автоматизация

отсутствует, т. е. к $LOA = 1$ (табл. 3). Для перехода к количественной шкале также часто используют математический аппарат нечетких множеств.

Заметим, что первые этапы автоматизации (модернизация при внедрении ПТС) вносят более значительный вклад в изменения по сравнению с последующими. При сопоставлении преобразований (см. выше) со значениями K_a мы также сможем убедиться в их корректности или в каких-либо расхождениях в оценках.

Из описания уровней шкалы ШИВ видно, что с уровня LOA5 появляется автоматическое исполнение операций, т. е. система становится автоматизированной и при обязательном участии человека может уже работать самостоятельно. Мы можем говорить, что с уровня LOA5 (70 %-ного уровня автоматизации) у нас появляется АСУ, ЧПУ, машина с вычислителем и т. п. с набором всех технических средств и далее, с ростом LOA идет только их совершенствование.

Чтобы связать когнитивное понимание уровня автоматизации с уровнем ПТО и функциями контроля и управления, выражаемыми

Таблица 3

Шкалирование уровней АСУ ТП по шкале Т. Шеридана и В. Вепланка

Уровень автоматизации (LOA)	Степень LOA $\lg(LOA)$	Описание по ШИВ	КИПиА + ПЛК	SCADA	MES
1	0	Компьютер/система не предлагает помощь: человек должен принимать решения и выполнять все действия сам			
2	0,3	Компьютер/система предлагает человеку полный набор решений/действий, альтернативы (пример — работа с электронным справочником)			✓
3	0,48	Компьютер/система предлагает полный набор решений /действий, альтернативы и сужает выбор до нескольких вариантов			✓
4	0,6	Компьютер/система предлагает одну альтернативу			✓
5	0,7	Компьютер/система предлагает одну альтернативу и автоматически выполняет это предложение, если человек соглашается		✓	✓
6	0,78	Компьютер/система предлагает одну альтернативу и выполняет это предложение, если человек в течение ограниченного времени не накладывает вето на автоматическое выполнение операции			
			Ручное управление		
			Автоматическое управление		
7	0,84	Компьютер/система выполняет операции автоматически, обязательно информируя человека	✓	✓	✓
8	0,9	Компьютер/система выполняет операции автоматически и информирует человека, только если компьютер "спросит"	✓	✓	
9	0,95	Компьютер/система выполняет операции автоматически и информирует человека, только если он (компьютер) решит	✓	✓	
10	1	Компьютер/система решает все и действует автономно, игнорируя человека		✓	

через K_a , в табл. 3 в соответствии с уровневый построением АСУ ТП, мы отметили типичную функциональность системы для данных уровней в терминах ШИВ. Затем отметили полученный диапазон значений для вектора приоритетов K_a жирной стрелкой.

На уровнях КИПиА — ПЛК осуществляется измерение, автоматический контроль и управление ТП в реальном времени, как правило, без участия человека, в зависимости от настроек ПТС (отмеченные, как LOA-7...9 в терминологии ШИВ). Вклад, оцененный по сумме средних значений вектора приоритета v_{cp} по методике расчета для K_a , этих двух уровней (КИПиА—ПЛК) в конечную автоматизацию равен 79 % ($v^0 + v^1 = 0,53 + 0,26 = 0,79$) [5], что практически попадает на границу уровней LOA-6...7 (верхняя отметка стрелки табл. 3).

Следующий уровень SCADA соответствует добавлению функций автоматического сбора, регистрации, хранения в БД, представления информации, формирования отчетов и взаимодействия с другими сетями. Этот уровень отмечен в шкале ШИВ как LOA-6...9 (табл. 3). Для всех трех уровней, с 0-го по 2-й, оценка вклада по сумме средних значений вектора приоритета v_{cp} равна 92 % ($v^0 + v^1 + v^2$) [5], что шкалируется по LOA-9 (нижняя отметка для стрелки). Функция диспетчерского управления SCADA (отмеченная как LOA-5), что соответствует, к примеру, случаям нештатных ситуаций или редкой подстройки параметров для достижения оптимальной производительности малоинформативно для нас в контексте общей автоматизации. Полученные выше оценки показаны в табл. 3 в диапазоне 79...92 %.

Далее видно, что уровень MES, описанный функциональной моделью из 11 функций (8 функций с 2004 г.), разработанной MESA International, отвечает данному шкалированию для типовых задач управления производством [13] (к примеру, составление расписаний LOA-2...3, управление людскими ресурсами LOA-2...3, управление качеством LOA-3...4, управление производственными процессами LOA-5...7, сбор и хранение данных LOA-7 и т. д.). На уровне MES могут решаться некоторые вопросы оптимизации для принятия более эффективных решений [9] как при участии человека (LOA < 6), так и без участия человека (LOA > 5). Рассматривая MES как уровень автоматизации бизнес-процессов оперативного управления производством, путем логирования (отражение логики бизнес-процесса) всей цепочки операций можно применять оцен-

ку уровня автоматизации, при этом каждая операция должна быть формально описана и должна получить свой номер LOA [4]. В целом, при описании таких процессов уровень автоматизации может устанавливаться по самому минимальному LOA [4], или, как показано далее, по соотношению числа операций с соответствующим LOA и затем суммированием по средневзвешенному числу операций. В общей иерархии АСУ ТП говорится о вкладе этого уровня в автоматизацию, составляющим 6...10 %, и доля этих процессов в формуле (1) для интегрального K_a не велика [5]. Уровень MES повышает функциональность для решения когнитивных задач и в данном шкалировании при вкладе в общий уровень автоматизации ТП практически незначителен и может только дополнять существующий уровень путем автоматизации производственных процессов. Вместе с тем в результате внесения новых функций оптимизации процессов производства, их учета и применения дополнительных ПТС уровень автоматизации по шкале ШИВ снижается из-за привлечения дополнительных ресурсов и необходимости контроля исполнения. В данном случае для уровня MES шкалирование по ШИВ более информативно, чем при подсчете в K_a , и есть расхождение в оценках, поэтому в дальнейшем уровень MES мы не принимаем во внимание.

Несмотря на это идеализированная оценка для K_a по значению вектора приоритетов v_{cp} рассмотренных выше трех уровней с 0-го по 2-й корректно согласуется со шкалой ШИВ. Также при расчете коэффициента автоматизации K_a дается более точная информация в количественном выражении об уровне автоматизации данной системы. Таким образом, учет функциональности ПТС и ПО в K_a коррелирует со шкалированием LOA для информационных задач (взаимодействия человек—компьютер), а учет доли сигналов говорит о числе приборов для контроля/измерения и средств автоматизации (наличие механических устройств). Такой же механизм рассмотрения двух типов задач: механизации и информационно-управляющих отражена в объединенной шкале по работе [1] (см. следующий раздел).

Также рассмотренные выше уровни автоматизации АСУ ТП на основе оценки K_a , лежащей, как показано выше, в диапазоне 79...92 %, будут соответствовать количественным оценкам по шкале ШИВ степени LOA-6...9 (0,78...0,95), а также предпоследним уровням автоматизации "5—6" (75...99 %) в вышеупомянутой шкале из

ГОСТ 23004—78 (табл. 2) [7]. В качественном описании, в соответствии с ГОСТ 23004—78 эти уровни отражены как "повышенный" (75...90 %) уровень автоматизации для КИПиА — ПЛК и как "высокий" (90...99 %) для всех трех уровней. Отметим, что соответствие оценок по вектору приоритетов уровней КИПиА — ПЛК "повышенному" уровню автоматизации и "высокому" вместе с уровнем SCADA в соответствии с ГОСТ 23004—78 действительно отвечает комплексной автоматизации и высокому уровню АСУ ТП на производстве. Оснащение трех выше перечисленных уровней автоматизации по последнему слову техники современными ПТС с реализацией на нем ПО, выполняющего требуемые функции согласно ТП, о чем, помимо высокого показателя K_a , косвенно еще может говорить шкалирование по уровням LOA-7...9 шкалы ШИВ (табл. 3), действительно позволит реализовать передовой и высокий уровень автоматизации.

Заметим, что в формуле (1) для интегрального K_a напрямую не учитывается оценка работы алгоритмов и контуров управления, но косвенно также учитывается управление по числу задействованных точек контроля и регулирования коэффициентом полноты автоматизации для 0-го уровня КИПиА по всем агрегатам ТП. При проектировании АСУ ТП в качественном описании отмечают до четырех степеней развитости информационных функций по типу контроля и измерения, включая распознавание ситуаций, и до семи степеней развитости управляющих функций по типу регулирования и управления до самообучения и изменения алгоритмов [19]. Очевидно, что высокий технический уровень ПТС, связанный с определенными затратами, подразумевает реализацию как достаточно устойчивых и надежных алгоритмов, не приводящих к авариям, так и надежных конфигураций безопасности. Оценка надежности работы алгоритмов и программной конфигураций ПТС выходит за рамки данного исследования. Недостатком расчета K_a является отсутствие показателей, учитывающих эти функции, и в большей степени качество управляющих функций — тип регулирования, интеллектуальные функции алгоритмов. Также не всегда корректен принцип аккумуляции уровня автоматизации в формуле (1), как было показано выше на примере с уровнем MES.

Теперь рассмотрим пример количественного выражения уровня автоматизации АСУ процессами производства (АСУПП) с использова-

нием показателя LOA по шкале ШИВ. Путем присвоения всем операциям соответствующего LOA и затем разделения всех операций была проведена количественная оценка бизнес-процессов в процентном отношении с ручным ($LOA < 6$) или автоматическим ($LOA > 5$) исполнением операций [4]. Результатом явилось получение в процентах числа операций АСУПП с ручным и автоматическим исполнением. Следует отметить, что результаты не уточнялись по разным LOA, а учитывались только по двум группам, как $LOA > 5$ и $LOA < 6$. Уровень автоматизации не коррелировал с оценкой для операций рядом стоящих LOA в каждой из групп и был одинаковым для LOA-6, LOA-7 и LOA-8. При детальном рассмотрении АСУПП видно, что показатель LOA в одной группе отличается по разным функциям контроля и рискам взаимодействия, которые возлагаются на человека и компьютер. Возможным выходом из данной ситуации может быть присвоение значения степени автоматизации $\lg(LOA)$ для каждого уровня согласно табл. 3. (0; 0,3; 0,48; 0,6; 0,7; 0,78; 0,84; 0,9; 0,95; 1) и/или присвоение веса каждому LOA аналогично вектору приоритетов, рассмотренному выше, т.е. с увеличением LOA будет рост автоматизации на какую-то определенную долю при расчете в итоговом K_a . К примеру, по LOA-7 устанавливается уровень автоматизации, равный 0,84 (84 %), а по LOA-8 — уровень 0,9 (90 %) при суммировании в интегральном показателе по средневзвешенным значениям числа операций по каждому LOA. Это позволит точнее представлять уровень автоматизации, в результате будет более точным представление уровня автоматизации для систем с процессами, имеющими близкий уровень LOA.

Еще один подход для получения оценок коэффициента автоматизации K_a и коэффициента готовности G_a , учитывающего готовность к запуску автоматизированных систем в трехмерном ПА по соответствующим координатам (задачи, этапы проекта, виды обеспечения) для ТПП при проектировании, приводится в работе [3]. Помимо сопоставления полученных значений с ГОСТ 23501.108—85 [14], где $K_a < 25$ % соответствует низкому, K_a от 25 до 50 % — среднему, а $K_a > 50$ % — высокому уровню автоматизации, авторы предлагают оценивать и планировать стратегию развития и функционирования автоматизированной системы ТПП [3]. В основе метода лежит получение вектора развития системы и введение весов для различных элементов ПА по трем

координатам при суммировании в интегральном показателе готовности G_a . Интегральный показатель готовности G_a отражает ближайшую перспективу автоматизации предприятия и учитывает не только действующие системы, но и все незавершенные проекты.

Можно отметить, что показатели функциональности и степень полноты автоматизации, представленные в формуле (1) для K_a , нормированные вектором приоритетов v_{cp} , согласуются численно со шкалой ШИВ, соответствуя когнитивному описанию функций для данных уровней в табл. 3. Качественные и количественные оценки из ГОСТ 23004—78 находят свое подтверждение в сравнении с численными значениями вектора приоритетов v_{cp} идеализированной оценки. Таким образом, два типа рассмотренных шкал и оценка значения K_a взаимосвязанны, характеризуют уровень автоматизации для производства с различных сторон и соответствуют близким как качественным описаниям, так и количественным значениям. Также оценка по шкале LOA производственных операций может являться основой для количественного расчета и вывода об уровне автоматизации. Трехмерное представление систем автоматизации с показателем G_a может послужить перспективой для отражения на цифровой платформе перспективы развития систем предприятия.

Другие шкалы, выбор LOA и оптимизация затрат

Как говорилось выше, существует множество различных шкал и подходов к оценке уровня автоматизации. В работах [1, 16] было приведено объединение в одной семиуровневой шкале физического и когнитивного понимания уровня автоматизации, т. е. задач, описывающих механизацию — применение технического оборудования, и задач информации/контроля. Описание этих задач использовалось в качестве основы для оценок от полностью ручного управления и контроля до полностью автоматического или автономного (табл. 4). Объединение двух типов шкал позволяет рассматривать процесс автоматизации с точки зрения выполнения задач из двух классов одновременно, определяя величину физических возможностей и информационной поддержки [1]. Начиная с уровня LOA = 4 (предупреждение), затем LOA = 5 (контроль), и LOA = 6 (вмешательство) и т. д. видно сходство перечисленных задач для уровней КИПиА — ПЛК — SCADA по аналогии с рассмотренной выше системой АСУ ТП (LOA-5...9) (табл. 3).

Далее было замечено, что для каждой рабочей операции (рабочего места) существуют максимальный и минимальный пределы автоматизации, их можно варьировать, подбирая

Таблица 4

Объединенная семиуровневая шкала LOA для информационных задач и механизации производства [1]

LOA	Механизация и оборудование	Информация и управление
1	Полностью ручная работа, никаких инструментов не используется, используется только собственная сила мышц, например собственная сила	Полностью ручной уровень: пользователь создает свое собственное понимание ситуации и развивает свой курс действий на основе его/ее более раннего опыта и знаний
2	Статический ручной инструмент — ручная работа с поддержкой статического инструмента, например отвертки	Уровень принятия решения: пользователь получает информацию о том, что делать, или предложение о том, как задача может быть достигнута. Пример — рабочий заказ
3	Гибкий ручной инструмент — ручная работа с поддержкой гибкого инструмента, например регулируемого гаечного ключа	Уровень обучения: пользователь получает инструкции о том, как выполнить задачу, например, инструкции, руководства
4	Автоматический ручной инструмент — ручная работа с поддержкой автоматизированного инструмента, например гидравлического болта	Уровень предупреждения: техническая система выдает запрос, если исполнение отклоняется от подходящего режима. Пример — проверка перед действием
5	Статическая машина / рабочая станция — автоматическая работа машины, предназначенной для конкретной задачи, например токарного станка	Уровень контроля — техника требует внимания пользователей и указывает на проблему. Пример — сигнализация
6	Гибкая машина / рабочая станция — автоматическая работа машины, которая может быть переконфигурирована для различных задач, например станка с ЧПУ	Уровень самовмешательства — техника берет на себя и корректирует действие, если исполнение отклоняется от варианта, который техника считает подходящим. Пример — термостат
7	Полная автоматизация — полностью автоматическая работа, машина решает все проблемы, которые возникают. Пример — автономные системы	Уровень полная автоматизации — вся информация и управление обрабатываются техникой. Пользователь никогда не участвует. Пример — автономные системы

различные уровни при непосредственном выполнении (табл. 5). Эти пределы отображаются в матрице LOA 7x7 (с физическими и когнитивными задачами), образуя площадь возможных улучшений Square of Possible Improvements (SoPI) [10]. SoPI иллюстрирует диапазон возможностей, где данные задачи и сборочные операции могут быть успешно автоматизированы или выполнены вручную. Данная методика получила название DYNAMO [10, 16].

Также исследователи обратили внимание на сборочном типе производства на различную стоимость мероприятий по автоматизации и других затрат в зависимости от уровня LOA [1, 11]. Поэтому в дальнейшем переключились на поиск оптимального баланса между затратами и автоматизацией, т. е. варианта, когда стоимость инвестиций не должна превышать ценность преимуществ, достигаемых посредством автоматизации (при высоком LOA система становится жесткой и не гибкой) (см. рисунок). Соответствующий минимум LOA определялся как благоприятный, когда работа может быть выполнена в подходящем темпе по приемлемой цене без ущерба для здоровья

Таблица 5

Градации уровня автоматизации (LOA) от минимального до максимального для каждой рабочей операций ТП [1]

LOA	Механизация и оборудование						
	Рабочие операции						
	1	2	3	4	5	6	7
7							
6	max	max		max		max	
5				min			
4					max	min	max
3							
2	min		max		min		
1		min	min				min

LOA	Информация и управление						
	Рабочие операции						
	1	2	3	4	5	6	7
7							
6							
5	max	max	max	max		max	max
4		min		min			
3					max	min	
2			min		min		
1	min						min

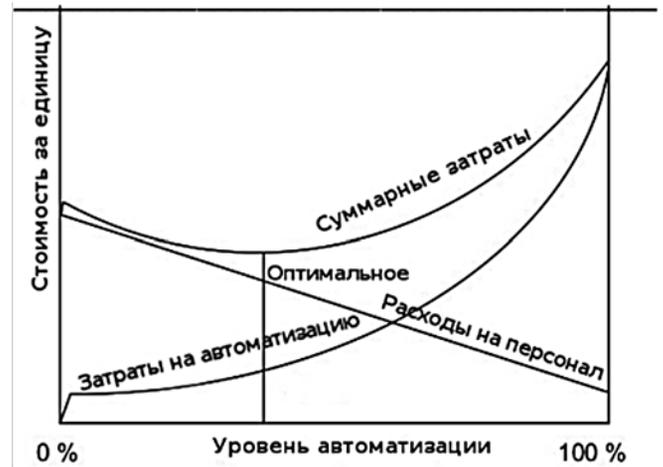


График зависимости стоимости от уровня автоматизации [11]

работника. Достигают этого путем разбиения технологического процесса каждого рабочего места на выполняемые операции с возможным максимальным и минимальным уровнями автоматизации в рамках принятых уровней объединенной шкалы и затем детальным рассмотрением каждой операции (табл. 5) [1].

При дальнейшем рассмотрении сборочного производства автомобилей из-за различий в используемых рабочей силе и оборудовании на одной и той же сборке будут разные общие затраты для разных уровней автоматизации (табл. 6) [11]. Самый оптимальный исходя из стоимости LOA будет не обязательно самым высоким. Различные значения LOA будут также для показателей "качество" и "производительность/гибкость" на различных участках сборки (УС) автомашин, как показано в табл. 7 по сравнению с фактическим уровнем автома-

Таблица 6

Затраты (€) на автоматизацию по участкам в зависимости от LOA по [11]

Уровень автоматизации (LOA)	Сборочный участок		
	1	2	3
1	1,30	1,20	1,30
2	1,20	1,30	1,20
3	1,00	1,10	1,10
4	1,10	1,00	1,00
5	1,40	1,40	1,40
6			1,50
7			1,60

Выбор оптимального LOA по ряду критериев [11]

Показатель	Модель автомобиля								
	Гольф А5, г. Вольфсбург			Туран 5000, г. Вольфсбург			Гольф А5, г. Эйттенхахе		
	УС1	УС2	УС3	УС1	УС2	УС3	УС1	УС2	УС3
Стоимость	1	4	4	3	4	4	5	5	7
Качество	3	4	4	3	4	1	3	4	1
Производительность/гибкость	1	2	4	2	3	4	4	4	6
Фактический LOA	1	1	1	4	4	4	5	5	7
Рекомендованный LOA	1	4	4	3	4	3	5	5	7

тизации. Определение оптимального LOA исходя из показателей "стоимость", "качество" и "производительность", на каждом участке сборки с рекомендованным уровнем автоматизации как минимум по двум показателям, главным из которых является "стоимость", не представляет трудностей. Окончательный выбор LOA в табл. 7 указан как "Рекомендованный LOA".

В итоге автоматизация производства только выигрывает от дифференцированного подхода, а полуавтоматическое исполнение становится неотъемлемой частью производства и резервом для повышения производительности [10, 11].

Перспективы шкалирования

Вышеупомянутым международным сообществом SAE international была стандартизована шестиуровневая шкала для определения уровня автономности в транспортной сфере от ручного до полностью автоматического контроля и управления, включая сценарии интеллектуального поведения движущихся объектов [2]. Уровень автономности не является уровнем автоматизации на производстве, но по смыслу эти понятия очень близки, и рост уровня автоматизации является прямым следствием повышения уровня ПТС, используемых для беспилотных машин и устройств. Основанная на современной методологии надежного техобслуживания Reliability-Centered Maintenance (RCM), своевременная диагностика этих средств позволит в оперативном режиме говорить о поломках и изменившемся уровне автономности [20]. Если говорить о производстве, то это могут быть роботизированные средства предприятия с текущим K_a , характеризующим изменившийся уровень автоматизации, аналогично показателю автономности выпускаемых изделий. В бизнес-системах

можно говорить о программах-роботах для роботизированной автоматизации процессов Robotic Process Automation (RPA), для которых характерна имитация взаимодействия человека с информационными системами при исполнении процессов [15]. В перспективе оценка систем предприятия, диагностика изделия, устройства и т. п. с предписанным уровнем автоматизации (автономности) по любой предложенной шкале аналогично Standard J3016, ГОСТ 23004—78 или табл. 3 даст более краткую и содержательную характеристику класса, к которому относится рассматриваемый объект. Это также будет являться дополнительной информацией для систематизации информационного пространства по соответствующему уровню автоматизации при создании цифровой платформы предприятия, оперативно информирующей об изменившемся уровне автоматизации. Иными словами, применение более современных ПТС, ПО и технологических решений позволяет говорить о более высоком уровне автоматизации по шкале LOA, а в количественном выражении об этом будет свидетельствовать соответствующий коэффициент автоматизации K_a . Увеличение K_a говорит о том, что данные средства и ПО выполняют больше функций, и оценка может занять более высокое место в шкале LOA, а стоимость ПТС, как и проекта, будет повышаться в соответствии с нормативными документами [19]. Данная информация может быть использована для обоснования инвест-проектов по модернизации систем [5] и сведена в более общую шкалу.

Дальнейшее исследование задачи определения уровня автоматизации АСУ ТП может идти путем формализации процессов подобно описанию информационной модели в диаграммах UML-нотаций [12], а для бизнес-процессов уровня MES и выше — в диаграммах функциональной модели по методологии IDEF0 [4]. Это

позволит выявить отдельные процессы и соответствующие ПТС на которых они запущены с возможным значением K_a . Выбор уровня автоматизации от максимального до минимального может дать дополнительную информацию с точки зрения оптимизации затрат.

Заключение

В работе был сделан анализ по систематизации вопросов, связанных с уровнем автоматизации, рассмотрены и сопоставлены различные шкалы. Показаны основные принципы деления, которые распространены в западной научной литературе, в некоторых подходах отечественных авторов и в нормативных документах. В качестве основной шкалы рассматривалась известная шкала Шеридина — Вепланка с введенным понятием уровня автоматизации LOA в применении к технологическим и производственным процессам автоматизации, в сравнении с коэффициентом автоматизации K_a и затем с показателем машинного времени средств оснащения $K_{СТО}$ шкалы ГОСТ 23004—78. Был сделан вывод, что шкалы и коэффициенты описывают процесс автоматизации с различных сторон и соответствуют близким по смыслу качественным и количественным описаниям. Расчетное значение K_a также может дать дополнительную более точную информацию при определении уровня автоматизации как при проектировании, так и при оценке бизнес-процессов, и может найти свое применение в обосновании инвест-проектов по модернизации систем. В дальнейшем существуют возможность оптимизации затрат в зависимости от уровня автоматизации (LOA), как и возможность определения изменения уровня автономности для динамических и других систем в зависимости от их состояния и применяемых ПТС. При развитии информационного пространства описанные методы определения уровня автоматизации могут сыграть свою положительную роль для сравнения и описания новых и существующих систем, послужить основой для разработки методик оценки различных типов процессов, а также могут использоваться для классификации систем и их стандартизации при создании цифровой платформы предприятия.

Список литературы

1. Frohm J., Stahre J. O., Winroth M. P. Levels of Automation in Manufacturing // International Journal of Ergonomics and Human Factors. 2008. Vol. 30, Iss. 3.

2. SAE international, Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems // Standard J3016. 2018. 35 p.

3. Куликов Д. Д., Яблочников Е. И. Применение оценочных метрик для анализа технологической подготовки производства // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2011. № 6. С. 109—112.

4. Кораблев И. Г. Оценка уровня автоматизации бизнес-процессов предприятия // Вестник Череповецкого государственного университета. 2016. № 1. С. 17—24.

5. Кузьмин Ю. Б. Моделирование степени автоматизации иерархических систем управления на примере АСУ ТП предприятия // Промышленные АСУ и контроллеры. 2017. № 6. С. 13—22.

6. Sheridan T. B., Verplank V., William L. Human and computer control of undersea teleoperators. Cambridge, Mass.: Massachusetts Institute of Technology, Man-Machine Systems Laboratory, 1978.

7. ГОСТ 23004—78. Механизация и автоматизация технологических процессов в машиностроении и приборостроении. Основные термины, определения и обозначения. М.: Изд-во стандартов, 1978. 29 с.

8. Саати Т. Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1993, 278 с.

9. Кузьмин Ю. Б. Мессирование как оно есть. Непрерывные процессы в газовой отрасли // ERPnews. 2008. № 1. С. 54—59.

10. Asth Å., Stahre J., Dencker K. Level of Automation Analysis in Manufacturing Systems. Chalmers University of Technology, Gothenburg, SE-41296, Sweden-2010.

11. Gortlach I. A., Wessel O. Optimal Level of Automation in the Automotive Industry // Engineering Letters. 2008. № 16.

12. Зорина К. В., Соколыч П. Ю. Описание АСУТП как изделия для разработки систем управления ЖЦ АСУТП // ВЕСТНИК ПНИПУ. Химическая технология и биотехнология. 2017. № 1. С. 53—60.

13. Загидуллин Р. Р. Управление машиностроительным производством с помощью систем MES, APS, ERP. Старый Оскол: ТНТ, 2011. 372 с.

14. ГОСТ 23501.108—85. Системы автоматизированного проектирования. Классификация и обозначения. М.: Изд-во стандартов, 1985. 15 с.

15. Лавров В. С., Петюк С. И. Роботизированная автоматизация процессов // Научные записки молодых исследователей. 2017. № 6. С. 43—45.

16. From J. Levels of automation in production systems. Thesis // Chalmers University of Technology, SE-412 96, Göteborg, Sweden, 2008.

17. Королев М. А., Фигурнов Э. Б. Статистика и экономический анализ в управлении народным хозяйством. М.: Экономика, 1985.

18. Власов А. В., Мартынов А. П., Кудояров Г. Ш. Экономика, организация управления и планирования нефтепродуктов. М.: Недры, 1975. 280 с.

19. СБЦП 81-2001-22. Государственный сметный норматив. Справочник базовых цен на проектные работы в строительстве. Автоматизированные системы управления технологическими процессами (АСУТП). М. 2016. 34 с.

20. SAE international, A Guide to the Reliability-Centered Maintenance (RCM). Standard JA1012. 2011. 62 p.

21. Панцагль И. Теория измерений. М.: Мир, 1976. 165 с.

22. Мицкевич А. А. Управленческая теория измерений. Ч. 2. Шкалы и накладываемые ими ограничения // Управляем предприятием. 2016. № 5 (65). URL: <http://upr.ru>.

23. Елисеева И. И., Курьшева С. В., Костеева Т. В. и др. Эконометрика. М.: Финансы и статистика, 2007. 576 с.

24. Камшилов С. Г. Определение уровня автоматизации бизнес-процессов на промышленном предприятии // Вестник Челябинского государственного университета. 2015. № 1 (356). Управление. Вып. 10. С. 56—61.

Taxonomy the Level of Automation in Production

The article is devoted to a review of the problem of assessing the level of automation in foreign and domestic literature. The analysis of classification of automation tasks and comparative characteristics of some taxonomies, their similarity and difference are given. Give a range of standards and methods for determining the level of automation are compared with the classical scale of T. Sheridan and V. Veplanka, with a another type of estimation — the vector of priorities of the automation ratio. Discussed is the level of automation for process control systems, business processes, production preparation and etc. The principle of combining taxonomy with different classes of automation tasks and further research with the purpose of optimizing the level of automation depending on the cost and other indicators of production are considered. The possibility of further research in formalizing various production processes and taking into account the level of autonomy of dynamic systems.

Keywords: level of automation, LOA, ratio of automation, degree of automation, taxonomy of automation, taxonomy of T. Sheridan and V. Veplank, taxonomy of autonomy

DOI: 10.17587/it.26.116-127

References

1. Frohm J., Stahre J. O., Winroth M. P. Levels of Automation in Manufacturing, *International Journal of Ergonomics and Human Factors*, 2008, vol. 30, iss. 3.
2. SAE international, Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, Standard J3016, 2018, 35 p.
3. Kulikov D. D., Yablochnikov E. I. The use of assessment metrics for the analysis of technological preparation of production, *Scientific and Technical Bulletin of the St. Petersburg State University of Information Technologies, Mechanics and Optics*, 2011, no. 6, pp. 109–112 (in Russian).
4. Korablev I. G. Assessment of the level of automation of business processes of an enterprise, *Bulletin of the Cherepovets State University*, 2016, no. 1, pp. 17–24 (in Russian).
5. Kuzmin Yu. B. Modeling the Degree of Automation of Hierarchical Control Systems by the Example of the Enterprise Process Control System, *Promyshlennye ASU i kontrolyery*, 2017, no. 6, pp. 13–22 (in Russian).
6. Sheridan T. B., Verplank V., William L. Human and computer control of undersea teleoperators. Cambridge, Mass., Massachusetts Institute of Technology, Man-Machine Systems Laboratory, 1978.
7. GOST 23004–78. Mechanization and automation of technological processes in mechanical engineering and instrument making. Basic terms, definitions and designations, Moscow, Publishing house of standards, 1978, 29 p. (in Russian).
8. Saati T. Making decisions. Method for analyzing hierarchies, Moscow, Radio i svyaz, 1993, 278 p. (in Russian).
9. Kuzmin Yu. B. Mesirovanie as it is. Continuous processes in the gas industry, *ERPnews*. 2008, no. 1, pp. 54–59 (in Russian).
10. Fasth Å., Stahre J., Dencker K. Level of Automation Analysis in Manufacturing Systems. Chalmers University of Technology, Gothenburg, SE-41296, Sweden-2010.
11. Gorlach I. A., Wessel O. *Optimal Level of Automation in the Automotive Industry*, *Engineering Letters*, 2008, no. 16.
12. Zorina K. V., Sokol'chik P. Yu. Description PCS as products for the development of control systems life cycle PCS, *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Khimicheskaya i biotekhnologiya*, 2017, no. 1, pp. 53–60 (in Russian).
13. Zagidullin R. R. Management of machine-building production using systems MES, APS, ERP, Stary Oskol, TNT, 2011, 372 p. (in Russian).
14. GOST 23501.108–85. Computer aided design systems. Classification and designations, Moscow, Izdatelstvo standartov, 1985, 15 p. (in Russian).
15. Lavrov V. S., Petyuk S. I. Robotic automation of processes, *Scientific notes of young researchers*, 2017, no.6, pp. 43–45 (in Russian).
16. From J. Levels of automation in production systems. Thesis, Chalmers University of Technology, SE-412 96, Göteborg, Sweden, 2008, 200 p.
17. Korolev M. A., Figurnov E. B. Statistics and economic analysis in the management of the national economy, Moscow, Economics, 1985, 235 p. (in Russian).
18. Vlasov A. V., Martynov A. P., Kudoyarov G. Sh. Economics, organization of management and planning of petroleum products, Moscow, Nedra, 1975, 280 p. (in Russian).
19. SBСР 81-2001-22. State estimated norm. Reference book of basic prices for design work in construction. Automated process control systems (APCS), Moscow, 2016, 34 p. (in Russian).
20. SAE international, A Guide to the Reliability-Centered Maintenance (RCM), Standard JA1012, 2011, 62 p.
21. Pfanzagl J. Theory of measurement, Moscow, Mir, 1976, 165 p. (in Russian).
22. Mitskevich A. A. Management Theory of Measurement Part 2. Scales and restrictions imposed by them, *We manage the enterprise*, 2016, no. 5 (65), available at: <http://upr.ru> (in Russian).
23. Eliseeva I. I., Kurysheva S. V., Kosteeva T. V. et al. Econometrics, Moscow, Finance and statistics, 2007, 576 p. (in Russian).
24. Kamshilov S. G. Determining the level of automation of business processes in an industrial enterprise, *Bulletin of Chelyabinsk State University*, 2015, no. 1 (356), *Control*, vol. 10, pp. 56–61 (in Russian).



ПОЗДРАВЛЯЕМ ЮБИЛЯРА!

Научному руководителю Института проблем проектирования в микроэлектронике РАН, члену Президиума РАН, Лауреату Государственной премии и Премии Правительства РФ, кавалеру Ордена Дружбы, Ордена Почета, Ордена Александра Невского, Главному редактору журнала "Информационные технологии", Председателю Совета Центрального дома ученых, академику РАН, доктору технических наук, профессору

Александр Леонидович Стемповскому

исполнилось 70 лет!

Александр Леонидович Стемповский — ведущий российский ученый в области автоматизации проектирования микроэлектронных схем и систем и новых вычислительных технологий, крупный организатор, создатель отечественной научной школы в области автоматизации проектирования микро- и нанoeлектронных систем.

Основные направления исследований академика А. Л. Стемповского — автоматизация проектирования СБИС и систем-на-кристалле, методы проектирования помехо- и отказоустойчивых

интегральных схем, отказоустойчивость параллельных вычислительных структур, разработка компьютерных систем на базе потоковой модели вычислений и применение модулярной арифметики для быстрых и надежных вычислений. А. Л. Стемповский является автором рекурсивной модулярной арифметики.

Александр Леонидович Стемповский имеет большой опыт организации международного сотрудничества, является одним из учредителей и членом Правления крупнейшей ежегодной европейской конференции по комплексной проблеме автоматизации проектирования и опыту разработки микроэлектронных систем DATE (Design, Automation and Test in Europe), много лет представляет Россию и РАН в международных организациях IFIP (Международная федерация по обработке информации) и IEEE (Институт инженеров по электротехнике и радиоэлектронике). Он организатор и председатель Оргкомитета регулярно проводимой с 2005 года ведущей Всероссийской конференции в области микроэлектроники «Проблемы разработки перспективных микро- и нанoeлектронных систем» (МЭС), научные труды которой входят в Перечень ВАК.

Талантливый ученый и организатор, высококвалифицированный специалист, чуткий, отзывчивый и обаятельный человек, А. Л. Стемповский пользуется глубоким уважением учеников, коллег, товарищей.

Редакционный совет, редакционная коллегия и коллектив редакции журнала сердечно поздравляют Александра Леонидовича Стемповского с юбилеем и желают ему здоровья, благополучия и дальнейших творческих успехов на благо российской науки и техники!

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Н. В. Яшина*.

Сдано в набор 09.12.2019. Подписано в печать 24.01.2020. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ220. 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. Сайт: www.aov.ru

Рисунки к статье Г. В. Данилова, А. А. Потапова, И. Н. Пронина, М.А. Шифрина, В. А. Васенина, А. Е. Подопригоры, В. В. Назарова, Н. В. Ласунина, Т. Е. Пронкиной, Ю. В. Струниной, Ю. А. Дорофеюк

«МЕТОДОЛОГИЯ ОЦЕНКИ АКТИВНОСТИ И РЕЗУЛЬТАТИВНОСТИ НАУЧНОЙ ДЕЯТЕЛЬНОСТИ В МЕДИЦИНСКОЙ ОРГАНИЗАЦИИ: ОПЫТ НАЦИОНАЛЬНОГО МЕДИЦИНСКОГО ИССЛЕДОВАТЕЛЬСКОГО ЦЕНТРА НЕЙРОХИРУРГИИ ИМЕНИ АКАДЕМИКА Н.Н. БУРДЕНКО»

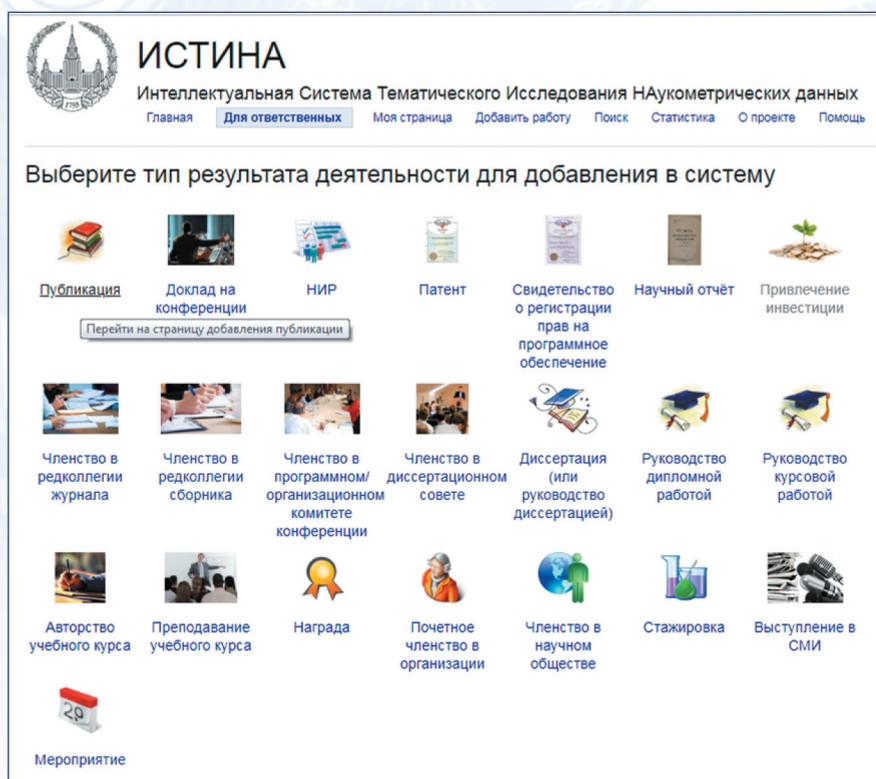


Рис. 3. Снимок экрана графического интерфейса пользователя информационной системы «ИСТИНА»

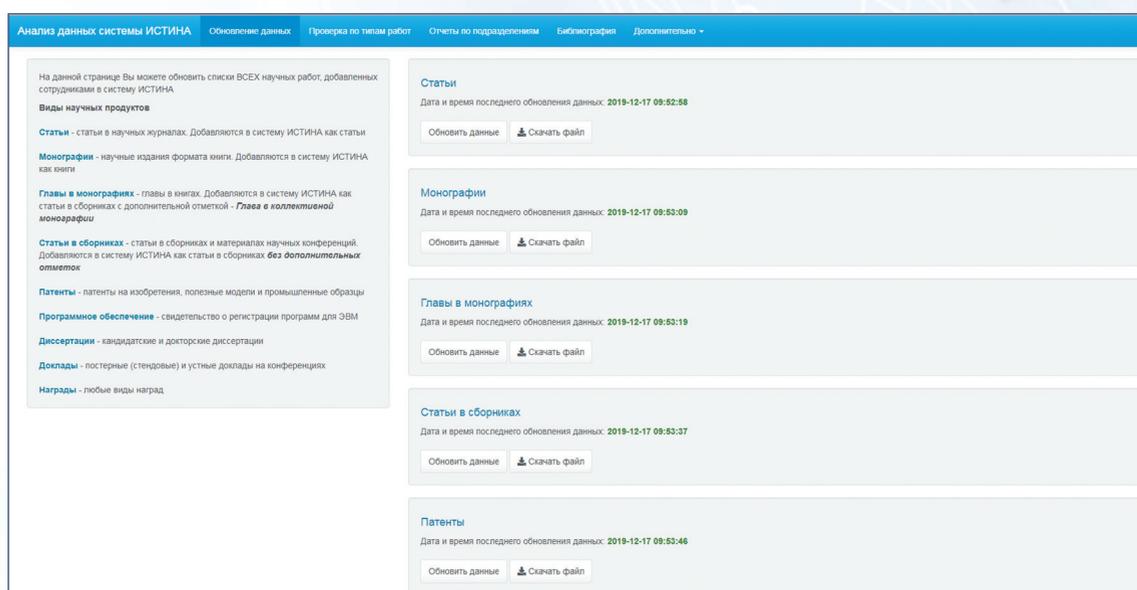
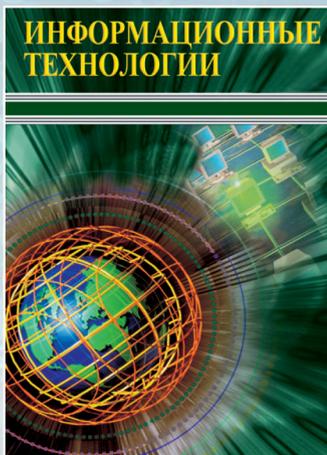


Рис. 4. Интерфейс аналитического модуля для системы мониторинга и оценки научной результативности сотрудников НМИЦ нейрохирургии

Издательство «НОВЫЕ ТЕХНОЛОГИИ»

выпускает научно-технические журналы



Ежемесячный теоретический
и прикладной научно-технический журнал
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

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

Подписной индекс по Объединенному каталогу
«Пресса России» – 72656



Научно-практический
и учебно-методический журнал

**БЕЗОПАСНОСТЬ
ЖИЗНЕДЕЯТЕЛЬНОСТИ**

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

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Ежемесячный
междисциплинарный
теоретический и прикладной
научно-технический журнал

**НАНО- и
МИКРОСИСТЕМНАЯ
ТЕХНИКА**

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



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Ежемесячный теоретический
и прикладной
научно-технический журнал

**МЕХАТРОНИКА,
АВТОМАТИЗАЦИЯ,
УПРАВЛЕНИЕ**

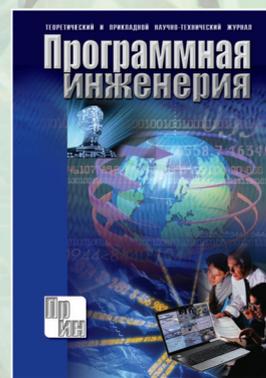
В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автоматического и автоматизированного управления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

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

Теоретический
и прикладной
научно-технический журнал

**ПРОГРАММНАЯ
ИНЖЕНЕРИЯ**

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



Подписной индекс по
Объединенному каталогу
«Пресса России» – 22765

Адрес редакции журналов для авторов и подписчиков:

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