

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Программная инженерия

Том 14. № 6. 2023





N* Новосибирский
государственный
университет
*НАСТОЯЩАЯ НАУКА

МАТЕМАТИЧЕСКИЙ
ЦЕНТР В АКАДЕМГОРОДКЕ

Институт математики им. С. Л. Соболева СО РАН
Новосибирский национальный исследовательский государственный университет
Математический центр в Академгородке
Российская Ассоциация Искусственного Интеллекта
Российская Инженерная Академия
Institute of Electrical and Electronics Engineers (IEEE)

IX Международная конференция ЗНАНИЯ – ОНТОЛОГИИ – ТЕОРИИ

2 – 6 октября 2023 г. Новосибирск

Целью конференции является ознакомление с новейшими научными достижениями, обмен знаниями и передовым опытом в области математических методов представления и анализа данных, извлечения знаний и построения теорий предметных областей, анализа формальных понятий и извлечения информации из текстов естественного языка. Сборник трудов конференции будет проиндексирован в РИНЦ, избранные статьи будут опубликованы в IEEE Xplore и проиндексированы в Scopus.

Тематика конференции отражает основные стадии процесса познания

- **Обнаружение закономерностей и извлечение знаний**, скрытых в структурированных и неструктурированных данных. Машинное обучение. Распознавание образов, анализ данных. Прогнозирование. Индуктивный вывод.
- **Систематизация знаний**. Инженерия знаний. Управление знаниями. Извлечение знаний из текстов на естественном языке. Разработка онтологий предметных областей, технологии создания и применения онтологий.
- **Построение теорий предметных областей**. Разработка семантических и онтологических моделей предметных областей. Анализ формальных понятий. Логическая семантика естественного языка. Нечёткие логики.

Работа конференции планируется в виде пленарных, секционных и стендовых докладов и круглых столов по тематике конференции. Рабочие языки конференции – русский и английский.

Контактные данные для переписки: zont@math.nsc.ru

Подробности: <http://old.math.nsc.ru/conference/zont/23/>

Программная инженерия

Том 14
№ 6
2023
Пр
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

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

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

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

Редакция

Чугунова А.В.

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

СОДЕРЖАНИЕ

- Васенин В. А., Чечкин А. В., Пирогов М. В.** Три принципа интеллектуализации автоматизированных систем 263
- Гостев И. М., Голосов П. Е.** Анализ эффективности облачной вычислительной системы, обслуживающей поток заданий с директивными сроками выполнения при множественных отказах серверов 278
- Терёхина И. Ю.** Выявление аномалий с использованием построенной модели процессов в виде ациклического ориентированного графа 285
- Кулаков К. А., Москин Н. Д., Рогов А. А., Воронов Р. В.** Программный комплекс СМАЛТ как инструмент для исследования теоретико-графовых моделей текстов 292
- Palchevsky E. V.** Method for Improving the Accuracy of Predictive Values of Time Series Based on the Imputation of Historical Data 301

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в подписных агентствах (индекс по Объединенному каталогу "Пресса России" — 22765) или непосредственно в редакции (для юридических лиц).

Тел.: (499) 270-16-52.

Http://novtex.ru/prin/rus E-mail: prin@novtex.ru

Журнал включен в Российский индекс научного цитирования (РИНЦ) и Russian Science Citation Index (RSCI).

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

© Издательство "Новые технологии", "Программная инженерия", 2023

SOFTWARE ENGINEERING

PROGRAMMAYA INGENERIA

Vol. 14

N 6

2023

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

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

Editor-in-Chief:

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

Editorial Board:

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

Editors:

CHUGUNOVA A.V.

CONTENTS

Vasenin V. A., Chechkin A. V., Pirogov M. V. Three Principles
of Intellectualization of Automated Systems 263

Gostev I. M., Golosov P. E. Analysis of the Efficiency of the
Simulation Model of Cloud Computing in Case of Multiple Server
Failures 278

Teryokhina I. Yu. Anomaly Detection Using the DAG Process
models 285

Kulakov K. A., Moskin N. D., Rogov A. A., Voronov R. V.
SMALT Software Package as a Tool for the Study of Graph-Theoretic
Models of Texts 292

Palchevsky E. V. Method for Improving the Accuracy of Predictive
Values of Time Series Based on the Imputation of Historical Data 301

В. А. Васенин, д-р физ.-мат. наук, проф., vasenin@msu.ru, МГУ им. М. В. Ломоносова,
А. В. Чечкин, д-р физ.-мат. наук, почетный профессор, A.Chechkin@mail.ru,
Военная академия ВВС им. Петра Великого, Балашиха,
М. В. Пирогов, канд. физ.-мат. наук, инженер 2 категории, pmv_mvpr@mail.ru,
НПО им. С. А. Лавочкина, Москва

Три принципа интеллектуализации автоматизированных систем

Поступила в редакцию 16.02.2024

Принята к публикации 12.04.2023

Целью исследования, результаты которого представлены в статье, является изложение концептуальных положений создания и развития новой технологии моделирования и программирования средств автоматизации процессов управления сложноорганизованными системами с использованием методологии интеллектуализации, которая проиллюстрирована на примере ее использования на всех этапах жизненного цикла. В контексте настоящей статьи такая технология именуется радикальной.

Рассмотрены три принципа интеллектуализации: учет разнохарактерного целеполагания при разработке таких систем; принцип необходимой информационно-системной избыточности; принцип достаточной целостности в предоставлении средств автоматизации. В основе новой технологии лежат математические объекты — радикалы и схемы радикалов, нацеленные на устранение неудобств, связанных с необходимой избыточностью математических моделей и реализующих их программно-технических средств в области сложных автоматизированных систем.

Обсуждены вопросы стандартизации технологии создания и сопровождения автоматизированных систем на основе применения схем радикалов. Обоснована необходимость применения таких схем для представления соответствующих частных моделей систем во многих областях их применения.

Положения предлагаемой технологии в кратком изложении основаны на анализе опыта их использования при разработке программ для работы целевой аппаратуры космических аппаратов, в том числе для синтеза, эксплуатации и модификации соответствующего прикладного программного обеспечения.

Ключевые слова: программное обеспечение, информационное обеспечение процессов интеллектуализации технических систем, информационно-системная безопасность автоматизированной системы, космическая съемка, среда радикалов, схема радикалов, радикальное моделирование, радикальное программирование, язык RADICAL, информационная форма

Для цитирования:

Васенин В. А., Чечкин А. В., Пирогов М. В. Три принципа интеллектуализации автоматизированных систем // Программная инженерия. 2023. Том 14, № 6. С. 263—277. DOI: 10.17587/prin.14.263-277.

Введение

Вопросы разработки программного обеспечения средств и систем автоматизации процессов управления перманентно развивающимися и усложняющимися объектами всегда были и остаются в центре внимания специалистов в области инфор-

матики. Особое место в перечне таких объектов и приложений занимают сложноорганизованные, многоагентные системы авиационно-космического комплекса, системы, автоматизирующие процессы в сфере промышленного производства, медицины и др. В связи с необходимостью автоматизации и интеллектуализации процессов сопровождения

такого класса объектов в реальном окружении в России и в других странах развиваются новая методология и технология инженерии разработки и сопровождения программ [1]. Проблемная область сложных автоматизированных систем, к которым относятся в том числе космические системы, охватывает большое число разнообразных составляющих таких систем, их свойств и связей. Для подобного класса автоматизированных систем характерны: изменчивость; архитектурно-технологические расширения функциональных возможностей; повсеместные потребности в применении быстро развивающихся программно-технических средств различного назначения [2, 3].

Проблемной области автоматизированных систем присущи как значительные достижения, так и существенные недостатки, которые способны привести к масштабным, невосполнимым потерям, авариям и катастрофам. Наличие проблемных вопросов верификации и обеспечения высокой степени безотказности, трудности модификации и возможность семантических ошибок — все это характерно для современных автоматизированных систем, для их программно-технических средств.

Причина недостатков сложноорганизованных автоматизированных систем — широкое использование при их разработке технологий, которые не охватывают в должной мере этапы предпроектных исследований, включая моделирование. Для такого подхода характерны ситуации, в которых проблемная область автоматизированной системы, ее составляющие и их связи описываются недостаточно полно. В этом случае даже разработчики программного обеспечения, не говоря о пользователе, не могут оперативно и с достаточной полнотой пояснить алгоритмику и схему работы отдельных элементов программного обеспечения и, соответственно, не могут качественно и достаточно быстро ее модифицировать. Для документирования таких программных средств характерно широкое использование представления информации на основе контекстов естественного языка. Автоматизация процессов разработки и сопровождения такой документации, оценка ее полноты, правильности с точки зрения используемых данных и знаний существенно затруднены. Математическая формализация проблемной области автоматизированных систем, как правило, недостаточна. Она не соответствует объективно высокой сложности и ответственности подлежащих решению задач, особенностям применяемых программно-технических средств.

В настоящей статье предложена новая, единообразно описанная на основе понятия *радикала*

математическая стандартизация формального описания проблемной области автоматизированных систем и протекающих в них процессов. При такой стандартизации все значимые для проблемной области составляющие рассматриваются как формально представленные с помощью математического аппарата объекты, преобразуемые по принятым правилам.

Наиболее известным из подходов, основанных на методологии инженерии программ применительно к автоматизированным системам, является представленный в 2010 г. международный проект SEMAT (*Software Engineering Method and Theory*), основателями которого являются Ивар Якобсон, Бертран Майер и Ричард Соули [4]. Следует отметить, что исследования и развитие этого проекта проводят и российские ученые. В рамках концепции SEMAT в настоящее время ведутся активные исследования и прикладные работы, направленные на формирование его ядра [5]. Они проводятся в рамках стандартов OMG (*Object Management Group*) — консорциума, занимающегося разработкой и продвижением объектно-ориентированных технологий и стандартов как совокупности продиктованных практикой атомарных элементов (сущностей), именуемых альфами. Ядро и язык описания модели предметной области составляют базовую часть проекта, именуемую Essence [6]. Существенным требованием к ее составляющим является требование возможности адекватного и наглядного для разработчиков описания и адекватной (объективной) оценки состояния целевого программного продукта на всех этапах его производства. В качестве основных принципов, которые декларируются в рамках проекта Essence, получившего статус стандарта OMG, рассматриваются: **расширяемость**; **применимость** на практике (на разных этапах жизненного цикла продукта), включая поэтапную апробацию (моделирование) и оценку состояния; **компонуемость**, подразумевающая возможности использования уже зарекомендовавших себя на практике решений; **адаптируемость** как возможность использования уже апробированных на каком-то этапе решений в целевом продукте; **масштабируемость** как возможность применения продукта для более сложных архитектур с учетом их размеров.

Предложенный в настоящей статье проект, основанный на модели *радикального программирования и радикального моделирования*, является аналогом отмеченного выше проекта SEMAT/Essence. Модель, рассмотренная в настоящей статье, основана на единообразной, математически формализуемой стандартизации описания любой

проблемной области и ее окружения. В качестве базовых атомарных элементов, составляющих ядро данного подхода, рассматривают так называемые *радикалы* и механизмы их взаимодействия в ходе реализации автоматизированных процессов. В рамках этой модели разработан специальный язык моделирования RADICAL, учитывающий *избыточность* описания модели. С помощью этого языка обеспечивается выразительное, гибкое и адекватное представление проблемной области, визуально удобное разработчику для оценки состояния целевого продукта на разных этапах его жизненного цикла. В качестве **радикалов** рассматривают отдельные атомарные модули, *функционально замкнутые на используемые операции в среде окружения автоматизированной системы*, которые либо подлежат разработке, либо уже прошли апробацию на практике и рекомендованы к использованию по назначению.

Основными концептуальными принципами, по которым развивается проект, основанный на радикалах, являются следующие: **принцип учета разнохарактерного целеполагания**, который обеспечивается путем участия разных по функциональному признаку групп разработчиков при формировании требований к модели; **принцип необходимой избыточности**, который обеспечивает возможности расширения ядра целевой модели за счет включения в его состав избыточных радикалов, которые потенциально могут быть востребованы и по мере необходимости включены в модель на тех или иных этапах функционирования целевого продукта, в том числе в критических и нештатных ситуациях; **принцип достаточной целостности** проблемной области, гомеостаза автоматизированной системы.

На основе этих принципов обеспечиваются: *защита* от деструктивных информационных воздействий, *контроль* за ходом выполнения автоматизированных процессов, *интеллектуализация* модели, принятие решений *применения* дополнительных радикалов для реагирования на нештатные (ранее не предусмотренные, критические) ситуации.

Математически формализованное описание проблемной области должно достигаться с помощью подлежащей созданию и развитию технологии моделирования автоматизированных систем на основе понятия радикала, а также на основе опыта его практического применения. Разработка и совершенствование методов математически строгого описания автоматизированных систем и их среды окружения на основе такой технологии представляется стратегически важной задачей.

В настоящей статье отражено текущее состояние поисковых исследований и прикладных работ, направленных на развитие проекта, именуемого радикальным моделированием и радикальным программированием. Представленный в публикации материал, включая краткое изложение результатов применительно к автоматизации объектов в авиакосмической области, может рассматриваться также как призыв к новым исследователям, готовым к участию в работах на направлениях развития проекта.

1. Принцип учета разнохарактерного целеполагания в среде окружения автоматизированной системы для адекватного и выразительного представления проблемной области

Вопросы целеполагания (описания целевого назначения) автоматизированной системы тесно связаны со средой (сферой) ее окружения, с реакцией на ее проявления. Математически формализованная избыточная (радикальная) модель среды окружения автоматизированной системы должна учитывать многочисленные вопросы ее функционирования, принимая во внимание как внешние, так и внутренние факторы. Предлагается строить модель среды автоматизированной системы с учетом отмеченного выше разнохарактерного целеполагания как модель *проблемной* области системы. Для такой проблемной области характерно наличие как положительных качеств, так и недостатков. К недостаткам, в первую очередь, относятся вопросы верификации (оценки) полноты принятых и описанных в модели факторов, как следствие, уровня обеспечения безотказности системы в рамках принятой модели. Таким образом, могут быть востребованы новые средства для устранения отмеченных недостатков, а значит, механизмы для повышения уровня интеллектуализации автоматизированной системы. Такие механизмы с элементами искусственного интеллекта призваны обеспечивать решение не только запланированных на этапе проектирования модели штатных, но и нештатных задач. Принцип учета разнохарактерного целеполагания автоматизированной системы подчеркивает тот факт, что тактико-техническое обоснование для нее всегда остается за человеком (разработчиками). Автоматизированная система является человеко-центричной системой и осуществляет целеполагание и планирование для решения тактических и оперативных задач, обеспечивающих выполнение тактико-технического задания *под наблюдением* человека.

2. Принцип необходимой избыточности — основа радикального моделирования проблемной области

Отметим, прежде всего, что каждая сложная целенаправленная система должна рассматриваться как автоматизированная система с элементами искусственного интеллекта. Решение задач автоматизированными системами в рамках принятых проектом положений должно осуществляться на основе применения технологии радикального моделирования, опирающейся, в свою очередь, на математическую стандартизацию их проблемной области. Принцип необходимой избыточности автоматизированных систем требует наличия большого дополнительного (избыточного) информационного и системного ресурса, представленного в ранее отмеченной радикальной форме (среды радикалов). Концепция, основанная на понятии радикала, позволяет устранять негативные стороны избыточности, при активном использовании ориентированного на это информационно-системного ресурса. Сложная человеко-центричная автоматизированная система должна рассматриваться как сервисно-ориентированная система с наличием первой сигнальной (сенсорной) системы и второй языковой сигнальной системы с элементами искусственного интеллекта и общения на естественном языке.

2.1. Методы и средства технологии радикального моделирования

Основой стандартизированного представления (описания) проблемной области автоматизированных систем должны служить математические объекты — **радикалы** и **среда радикалов**. Такие объекты, как отмечалось ранее, являются средством формально строгого представления избыточной (радикальной) на случай появления нештатных (непредвиденных на этапе проектирования) ситуаций проблемной области автоматизированных систем. Основанные на понятии радикала и концепции среды радикалов [7, 8], а также на понятии информационно-системной безопасности новые технологии радикального моделирования и программирования рассматриваются авторами настоящей статьи как основа адекватного и выразительного описания проблемной области автоматизированных систем, как эффективное средство разрешения целого ряда вопросов, существующих на настоящее время [9] в области моделирования автоматизированных систем, создания их программно-технических средств. Под *радикалом* понимается [10, 11] любая *функционально*

ориентированная (предназначенная выполнять функции по своему назначению) *программно реализуемая сущность*, имеющая доступные *извне механизмы изменения ее состояния*. Эти состояния должны быть двух типов: **активные** (радикал «включен» и функционирует согласно своему назначению) и **пассивные** (радикал «выключен» и не функционирует согласно своему назначению, находится в *ожидании*, в *резерве*). Все радикалы образуют среду радикалов, составляющих основу радикальной модели проблемной области автоматизированных систем. Такая среда в пассивном состоянии является потенциальной средой для выделения путем активирования любой частной модели проблемной области автоматизированной системы. Все значимые составляющие проблемной области автоматизированной системы, процессы ее взаимодействия с внешним окружением представляются с помощью схем радикалов — формальных описаний на универсальном языке радикалов RADICAL. Семантика схем радикалов — составляющих проблемной области и их связей такова, что в активном состоянии они являются «действующими» частными моделями отдельных объектов или процессов их взаимодействия со средой окружения автоматизированной системы, а в пассивном состоянии они находятся в резерве — в пассивном состоянии. Организация модели проблемной области в формализме среды радикалов обеспечивает полное исключение негативного влияния огромной избыточности модели. Каждый раз пользователь модели активизирует и использует только *системоквант модели*, *выделенную часть модели из активных радикалов*. Все остальные радикалы находятся в пассивном состоянии, в резерве, в ожидании. Они составляют потенциал возможностей модели для расширения и коррекции рабочего системокванта.

Следует отметить, что в основе новой технологии *радикального моделирования и программирования* лежит реализация идеи «**лексикона программирования**» академика А. П. Ершова [12—14]. Требование **информационно-системной безопасности** автоматизированной системы к среде радикалов включает двойственную безопасность: **информационную** и **системную**. Последняя означает сохранение **достаточной системной целостности** автоматизированной системы для обработки последствий всех изменений ее проблемной области. В процессе развития информационно-коммуникационных технологий развивается трактовка аспектов информационно-системной безопасности автоматизированных систем. Например, в силу открытости системы растет число сущностей, целостность ко-

торых необходимо обеспечить [15]. Одна из основ радикального программирования, его лексикон (по А. П. Ершову) — RADICAL — является языковой средой, средством описания объектов проблемной области. Он содержит нотацию для построения соответствующих баз данных и знаний. Программа на языке RADICAL — «радикальная» программа — содержит подробное описание своей семантики, как это предполагалось в лексиконе. Внутреннее единство подлежащей формальному представлению сложной проблемной области, имеющей единую цель (цели), приводит к необходимости **единой информационной формы** такого представления.

Для обеспечения информационно-системной безопасности автоматизированных систем соответствующая среда радикалов рассматривается как средство поддержки следующих аспектов (направлений) создания и функционирования автоматизированных систем, видов ее обеспечения и отдельных компонентов по ГОСТ: **эргономического, организационного, методического, технического, математического, программного, информационного, лингвистического и правового** [16]. В перспективе может рассматриваться также ресурсное (включая финансовое) обеспечение. Жизненный цикл автоматизированной системы рассматривается как процесс модификации компонентов сопровождения автоматизируемого объекта в среде радикалов.

Сформулируем авторскую точку зрения на предлагаемый подход: **радикальное моделирование и программирование, язык RADICAL должны составлять основу единообразного, формально строгого описания проблемной области любой сложной неорганизованной автоматизированной системы.**

Такая формализация создает объективное более полное описание всей проблемной области, способствующее решению целевых задач. Средства сопровождения автоматизированной системы, их компоненты должны быть формально описаны на языке RADICAL и реализованы с помощью программно-технических средств. С помощью языка RADICAL должно обеспечиваться представление в схемах радикалов всех значимых составляющих проблемной области автоматизированной системы, включая их свойства и связи (как в статике, так и в динамике), а также представление преобразований проблемной области. При этом должны охватываться как штатные, так и (по мере появления) нештатные ситуации, возникающие в процессе функционирования системы. Конструкции языка RADICAL подлежат реализации с помощью программно-технических средств.

Формализованные знания о проблемной области предназначены для их использования

в процессе создания, сопровождения и развития автоматизированной системы. Для проблемной области такой системы должны быть определены соответствующие ей радикалы-уникумы и радикалы-контейнеры, представляющие свойства и связи уникалов. Должны быть также определены ультраконтейнеры, представляющие правила. Имена радикалов-уникалов начинаются символом «u», первый символ радикалов-контейнеров — «с». Из уникалов, контейнеров и ультраконтейнеров строятся более сложные схемы радикалов. Среда радикалов при этом представляется на языке RADICAL в виде схем радикалов. Схемами радикалов можно описать, например, космический аппарат, объект наблюдения и их характеристики. В языке RADICAL предусмотрены разнообразные стандартные представления схем радикалов, включая **символьные, геометрические, табличные, проективные**. На основе таких представлений обеспечивается возможность автоматизированного формирования практически любых, удобных по тем или иным причинам, дополнительных, выразительных представлений схем радикалов. В зависимости от решаемой задачи можно формировать различные **сечения среды радикалов**, управлять визуализацией тех или иных схем радикалов, «вырезать» из среды радикалов фрагменты и «склеивать» их, использовать в представлениях схем цвет и движение. Под сечением среды радикалов будем понимать ее часть — выделенную схему *активных радикалов*, являющуюся частной моделью рассматриваемой проблемной области.

Схемы радикалов могут изменяться во времени, что должно автоматически фиксироваться с помощью специальных радикалов представления времени. Для представления иерархических структур проблемной области автоматизированной системы применяются контейнеры, объединяющие целостности и их части.

Схемами радикалов, например, можно представлять следующие сущности (объекты) и факты проблемной области [9]:

- собственно автоматизированную систему, которая включает в себя космический аппарат и среду его окружения;
- космическая система, которая содержит космический и взаимодействующие с ней наземные комплексы;
- комплекс, объединяющий космический аппарат и ретранслятор;
- аппаратуру съемки внутри космического аппарата;
- наземный комплекс, включающий пункты приема информации, передаваемой из космоса.

Анализ проблемной области позволяет обоснованно предполагать, что с помощью схем радикалов могут быть описаны все значимые сущности, задействованные в процессах автоматизированного наблюдения внешних объектов с применением средств космических систем. Все объекты, факты и правила автоматизированной системы космической съемки должны быть представлены формально с помощью схем радикалов на основе логики принятой и представленной в модели проблемной области.

С помощью контейнеров и вложенных в них уникамов реализуется основная навигационная и системообразующая идея координатизации среды радикалов — идея **координатной системы контейнеров**, согласно которой каждый уникам ассоциирован (вкладывается) в различные и разнотипные контейнеры. Вложение уникамов в контейнеры инициируется специалистами автоматизированной системы при работе с радикальной моделью ее проблемной области с применением соответствующих ультраконтейнеров — правил в автоматизированном или автоматическом режиме.

В языке RADICAL предусмотрены средства для выделения («окрашивания») и активации схем радикалов. Предоставляется возможность формулировать запросы к радикальной модели проблемной области автоматизированной системы, получать и анализировать соответствующие ответы. Другими словами, *автоматизированные системы, работающие по технологии радикального моделирования и программирования, это интеллектуальные, умные системы*, когнитивно общающиеся с пользователями. Например, возможен запрос обо всех объектах наблюдения (объектах обработки), попавших в так называемую «мертвую зону» — область, расположенную на поверхности Земли непосредственно под космическим аппаратом, в которой съемка технически невозможна.

Конфликты в среде радикалов представляются недопустимыми заполнениями контейнеров. Когнитивное прогнозирование конфликтов, их разрешение, уход от конфликтов при решении разноплановых задач среды радикалов должны быть обеспечены с помощью правил-ультраконтейнеров. Преобразования схем радикалов должны рассматриваться с точки зрения учета приближения среды радикалов к возможным конфликтам и удаления от них.

Представляется, что разные типы задач проблемной области автоматизированных систем и разные методы их решения могут быть рассмотрены и проанализированы с помощью схем радикалов.

Предлагается общий метод решения задач в среде радикалов — *двунаправленный метод по-*

строения целевого уникама, решающего очередную задачу с помощью избыточной (на случай нештатных ситуаций) библиотеки радикалов. Введен *основной оператор радикального моделирования*, использующий двунаправленный метод для решения основной задачи управления и развития автоматизированной системы. Востребованность в решении такой задачи характерна для всех иерархических уровней модели системы, на которых осуществляется радикальное моделирование в целях реализации одних целевых радикалов с помощью других.

Выводы:

1. С помощью схем радикалов могут быть представлены как дискретные, так и непрерывные математические объекты. Последние представляются с помощью специальных уникамов. Схемы радикалов применимы в том числе для компонентов программного обеспечения (программные классы, функции и т. п.), информационного обеспечения (базы данных, таблицы и т. п.) и технического обеспечения автоматизированных систем. Для решения задач проблемной области таких систем все виды ресурсного обеспечения, все их компоненты могут и должны быть представлены схемами радикалов, реализованными с помощью программно-технических средств.

2. Все сложные автоматизированные системы, несмотря на их разнообразие по предмету назначения, схожи в возможностях применения схем радикалов для описания соответствующих проблемных областей.

3. Введено понятие представимой схемами радикалов типовой автоматизированной системы, в рамках которой должны обеспечиваться модификация и развитие стандартной технологии радикального моделирования, в том числе поэтапная *структуризация радикальной модели*:

- *первый этап* — *нормализация*, всеобъемлющая координатизация радикальной модели — фиксирование применяемых уникамов и контейнеров;
- *второй этап* — *ультраоснащение* радикальной модели, когда в модель добавляются ультраконтейнеры-правила;
- *третий этап* — *иерархическая организация* радикальной модели, осуществляемая по соответствующим правилам.

4. С помощью языка RADICAL и его выразительных возможностей в среде радикалов автоматизированной системы поддерживаются *режимы одновременных трех стандартных информационных процессов*:

- *целевого*, отвечающего за решение задач по основному назначению системы;

- *сенсорного*, отвечающего за постоянную ситуационную коррекцию, полноту и адекватность радикальной модели, за ее динамическую адаптацию в складывающихся ситуациях;

- *сертификационного*, отвечающего за решение гомеостатических задач сохранения *достаточной целостности* автоматизированной системы и ее проблемной области.

5. В рамках технологии радикального моделирования язык RADICAL рассматривается как изменяющаяся стандартизованная языковая среда, развитие которой осуществляется в ходе решения основной задачи радикального моделирования и радикального программирования — *задачи достижения целевого состояния* проблемной области автоматизированной системы из ее исходного состояния вместе со всеми подзадачами такой задачи и обязательным обеспечением информационно-системной безопасности.

6. Согласно *тезису о выразительности языка RADICAL*, любая математическая модель любой значимой составляющей проблемной области автоматизированных систем, описываемой в терминах сущностей, свойств, связей, событий, преобразований, правил, выделения, активации и построения, реализации одних математических объектов с помощью других математических объектов, может быть представлена (как в статике, так и в динамике) схемами радикалов. Под решением задачи в среде радикалов понимается построение «выходных» радикалов на основе исходных радикалов и радикалов, представляющих доступные ресурсы.

Для иллюстрации тезиса о выразительных возможностях языка RADICAL далее приведены четыре примера представления схем радикалов. Однако следует отметить, что все приведенные в данной статье примеры и рисунки имеют упрощенный, демонстрационный характер.

На рис. 1–3 приведена в различных представлениях одна и та же схема радикалов, в которую входят радикалы-уникимы, радикалы-контейнеры и служебные радикалы $d[1]$ и $d[2]$, идентифицирующие направления в контейнерах.

Первое представление — символьное представление схемы радикалов (рис. 1), на котором присутствуют два контейнера — $cInteractingSystems$ (взаимодействующие системы) и $cSystems_System$ (множество систем и входящая в его состав система).

Первый из этих контейнеров — $cInteractingSystems$ — связывает два множества взаимодействующих систем — множество всех естественных систем $uAllNaturalSystems$ (доступное

```
cInteractingSystems →
{
    d[1]Systems→ uAllNaturalSystems;
    d[2]Systems→ uAllArtificialSystems;
}
cSystems_System→
{
    d[1]Systems→ uAllArtificialSystems;
    d[2]System→ uNewArtificialSystem;
}
```

Рис. 1. Символьное представление схемы радикалов

в рассматриваемом контейнере по направлению $d[1]$) и множество всех искусственных систем $uAllArtificialSystems$ (доступное в рассматриваемом контейнере по направлению $d[2]$).

Второй контейнер — $cSystems_System$ — объединяет множество всех искусственных систем $uAllArtificialSystems$ и новую искусственную систему $uNewArtificialSystem$, которая должна войти в это множество. В контейнере $cSystems_System$ к этим радикалам ведут направления $d[1]$ и $d[2]$.

Второе представление — геометрическое представление рассматриваемой схемы радикалов — приведено на рис. 2.

На плоскости введена прямоугольная система координат, оси которой направлены вправо (ось H) и вниз (ось V). На оси V зафиксированы радикалы, представляющие типы используемых контейнеров, а также используемые уникамы:

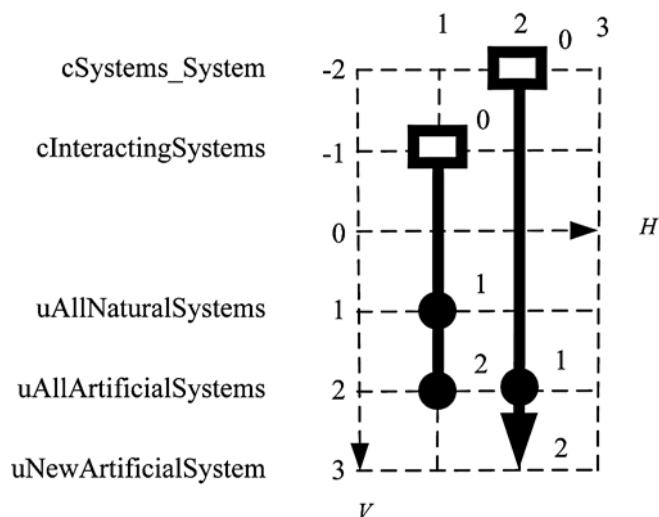


Рис. 2. Геометрическое представление схемы радикалов

cInteractingSystems ($V = -1$); cSystems_System ($V = -2$); uAllNaturalSystems ($V = 1$); uAllArtificialSystems ($V = 2$); и uNewArtificialSystem ($V = 3$).

На вертикалях $H = 1, 2$ представлены экземпляры контейнеров типов cInteractingSystems ($V = -1$) и cSystems_System ($V = -2$). Точки ($H = 1; V = -1$) и ($H = 2; V = -2$) пересечения вертикалей $H = 1$ и $H = 2$ с горизонталями типов контейнеров $V = -1$ и $V = -2$ помечены для наглядности прямоугольниками, а также символом «нулевого» направления «0».

Выделенная точка ($H = 1; V = 1$) означает, что имеет место вложение в рассматриваемый экземпляр контейнера cInteractingSystems (вертикаль $H = 1$) уникала uAllNaturalSystems, расположенного на горизонтали $V = 1$, а также, что уникал uAllNaturalSystems доступен в этом экземпляре контейнера по направлению «1» — точка ($H = 1; V = 1$) помечена этим символом.

Точка ($H = 1; V = 2$) помечена символом 2, т. е. в рассматриваемом контейнере доступен также по направлению 2 уникал uAllArtificialSystems, который зафиксирован на горизонтали $V = 2$.

Возможно также представление доминирования одних уникалов другими. В нашем примере в контейнере вертикали $H = 2$ уникал uAllArtificialSystems, лежащий на горизонтали $V = 2$, доминирует уникал uNewArtificialSystem, лежащий на горизонтали $V = 3$, для представления чего точка ($H = 2; V = 2$) помечена кругом, а в точку ($H = 2; V = 3$) входит стрелка. В остальном, «устройство» вертикали $H = 2$ аналогично «устройству» вертикали $H = 1$.

Третье представление — табличное представление той же схемы радикалов — приведено на рис. 3.

Числовое поле (столбец) V таблицы — это поле ее первичного ключа. Поле (столбец) B может со-

V	B	C	H1	H2
-2	c	Systems_System		0
-1	c	InteractingSystems	0	
1	u	AllNaturalSystems	1	
2	u	AllArtificialSystems	2	1
3	u	NewArtificialSystem		2

Рис. 3. Табличное представление схемы радикалов

держат либо символ «с» (и тогда строка характеризует радикал-контейнер), либо символ «u» (в этом случае строка соответствует радикалу-уникуму). Перейдем к столбцу C. Тип контейнера представлен в верхней строке таблицы, уникамы находятся в нижней ее части. Экземпляры контейнеров представляются в столбцах H1 и H2 аналогично рассмотренному выше геометрическому представлению.

Представления схем радикалов, приведенные на рис. 1–3, относятся к **L-представлениям** (L от logic, логика). Они позволяют сконцентрироваться на логике проблемной области, на ее составляющих, их свойствах и связях. В языке RADICAL используются также представления четвертого типа — **P-представления** (P от projection, проекция).

Четвертое представление — проективное представление той же схемы радикалов. На рис. 4 приведен пример таблицы P-представления, состоящей из «элементарных» уникалов-квадратов u11, ..., u33.

Приведенный пример показывает, что с помощью P-таблиц можно, например, представлять любые твердые тела проблемных областей сложных систем (их проекции), как в начертательной геометрии. Например, объекты наблюдения (их проекции), «окрашивая» нужные уникамы-квадраты, как это показано на рис. 4. Можно показать, что P- и L-представления схем радикалов связаны между собой. Обработку целевых объектов проблемной области можно выполнять по частям. Например, планирование космической съемки объектов наблюдения можно осуществлять на основе «послойного» разбиения множества таких объектов по важности на конечное число подмножеств, представляемых P-таблицами. При проведении съемки геометрические проекции полосы обзора космического аппарата зависят от ограничений, налагаемых целевой аппаратурой, и также могут

u11	u12	u13
u21	u22	u23
u31	u32	u33

Рис. 4. Таблица P-представления

быть представлены с помощью Р-таблицы. Расчет возможности обработки (съемки) объектов наблюдения проводится на основе системы правил, учитывающих временные и энергетические характеристики модели. Вначале определяют возможность съемки наиболее важных объектов наблюдения, затем переходят к менее важным объектам.

2.2. Интеллектуализация и автоматизация технологии радикального моделирования и радикального программирования

На радикальном моделировании основана технология радикального программирования — технология пошаговой разработки, апробации и сертификации программно-технических средств с применением библиотеки схем радикалов. На практике возможны различные виды (варианты, режимы) автоматизации технологии радикального программирования, расположенные между «минимальной» (технологией ручного режима) и «максимальной» (технологией автоматического режима). В режиме минимальной технологии схемы радикалов существуют отдельно от программного кода — текстов на языках программирования. Специалисты при этом работают и со схемами, и с кодом. Не следует считать этот режим неполноценным, неработоспособным. Специалист получает возможность разрабатывать формальные описания проблемной области и анализировать их, что сразу дает качественно более высокий уровень наглядности и полноты ее представления в виде схем радикалов.

Отметим, что радикальное моделирование и программирование неконфликтны по отношению к другим средствам автоматизации проблемной области автоматизированных систем. Необходимое «взаимопонимание» и взаимодействие всегда могут быть отражены на основе применения языка RADICAL.

Интенсивность использования схем радикалов можно увеличивать постепенно (поэтапно), что, конечно, не исключает возможности в отдельных случаях использовать этот формализм вплоть до уровня его реализации в автоматическом режиме. Программные коды при этом должны генерироваться автоматически на основе принятых схем. Что касается программно-технических средств реализации схем радикалов, то они могут быть разными по тем или иным причинам. В результате научно-технического развития одни из них могут заменяться другими, в то время как сама технология работы со схемами радикалов в основе своей остается неизменной. Таким образом, авторы считают, что язык RADICAL на настоящее время

является одним из последних формальных языков программной инженерии в рассматриваемом смысле. Применение схем радикалов для решения целевых задач в ходе развития технологии радикального моделирования от начального режима ее реализации до «максимума» (реализации всех определенных техническим заданием функций) позволит на единой основе развивать формализацию проблемной области. Такой подход сделает эту область визуально доступной для пользователя в различных сечениях, с разных «точек зрения», сделает ее удобной для перманентного анализа, контроля, модификации и сравнения вариантов решения задач, начиная с самых ранних этапов жизненного цикла автоматизированной системы. Благодаря применению языка RADICAL появляется принципиальная возможность систематического тестирования и «многоуровневой» сертификации получаемых решений, обеспечения ответственности в ходе их реализации.

Предлагаемые технологии могут и должны развиваться на основе результатов постоянно проводимых научных исследований и прикладных разработок, относящихся к разнообразным автоматизированным системам и их программно-техническим средствам. Соответствующие результаты должны представляться на языке RADICAL, анализироваться и оцениваться с помощью систем ультраконтейнеров-правил. Таким образом, должно быть сформировано мнение (объективная оценка) о каждом средстве автоматизации, о каждом результате соответствующих исследований с точки зрения возможного использования таких средств для тех или иных автоматизированных систем в рамках технологии радикального моделирования и программирования.

Важно подчеркнуть объективно существующую потребность общих механизмов для разных автоматизированных систем. Казалось бы, далекие друг от друга по своему функциональному назначению и среде окружения автоматизированные системы могут оказаться похожими в плане востребованности в их составе одних и тех же функциональных механизмов и, как следствие, реализующих их радикалов. Это обстоятельство предоставляет возможность многоцелевого использования радикалов. Конечно, в моделях проблемных областей автоматизированных систем много разнообразия (специфики), что приводит к сложностям при обеспечении их сосуществования и взаимодействия. Представляется, что положение дел здесь может быть существенно улучшено на основе применения технологии радикального моделирования. Технология радикаль-

ного моделирования и технология радикального программирования предлагают единый подход к решению задач проблемной области автоматизированных систем, основанный на применении радикалов — уникамов, контейнеров, ультраконтейнеров (правил), запросов и ответов на них, на применении L- и P-представлений схем радикалов и двунаправленного метода построения целевых уникамов, а также основного оператора, использующего этот метод. Такой подход предоставляет возможность реализации элементов интеллектуализации, контролируемого построения и преобразования унифицированных формальных описаний проблемной области для решения разных целевых задач, а также для организации взаимодействия различных автоматизированных систем.

3. Информационная форма в радикальном программировании как реализация принципа достаточной целостности

В этом разделе представлены основные понятия, используемые при формировании информационного описания проблемной области в радикальном программировании. Некоторые схемы радикалов проиллюстрированы на примере космической съемки как проблемной области.

3.1. Понятие информационной формы и единой системы информационных форм

Принцип достаточной целостности проблемной области автоматизированных систем заключается в обеспечении **целостного (с учетом текущих знаний) описания сущностей радикальной модели этой области (включая объект автоматизации, среду его окружения и саму автоматизированную систему)** посредством универсального языка технологии радикального моделирования — языка RADICAL. При этом целостность должна быть достаточной для достижения поставленной цели. Целостность описания сущностей достигается посредством информационных форм этих сущностей. Информационная форма — это схема радикалов особого назначения, сечения которой полностью определяют все внутренние и внешние значимые связи уникама сущности с уникамами-компонентами всех видов обеспечений проблемной области.

Информационные формы должны позволять, находясь в рамках уникама сущности, переходить от его базовых иерархических уровней к *внутренним микроуровням*, а также к *внешним макроуровням* радикальной модели. В этом состоит идея многоуровневости среды радикалов проблемной области автоматизированных систем. При решении целевых

задач проблемной области *достаточная целостность* ее должна быть обеспечена на всех уровнях. Необходимая для этого навигация в среде радикалов обеспечивается горизонтальными и вертикальными *лифтами* (понятие «лифт» будет представлено позже), реализуемыми с помощью информационных форм.

Добавим к этому идею изменяющихся «лабиринтов», которые могут располагаться как в пределах одних уровней, так и между ними. «Лабиринты» составляют системы контейнеров, относительно которых «перемещаются» (изменяются) уникамы в процессе достижения поставленной цели, потребляя для этого необходимые ресурсы. Средства реализации и изменения «лабиринтов» должны основываться на информационных формах. Целевые информационные формы строятся с помощью библиотечных информационных форм. *Информационные формы — математические объекты глубинного смыслового уровня*, служащие для единообразного представления всех составляющих среды радикалов проблемной области любой автоматизированной системы. Информационные формы — это составляющие базы знаний проблемной области и средства построения, эволюции такой базы знаний. Каждая информационная форма должна быть реализована с помощью программно-технических средств, сертифицирована и включена в радикальную модель проблемной области как ее часть. Каждая информационная форма должна быть полностью готова к работе в составе программно-технических средств автоматизированной системы после ее выбора в базе знаний, извлечения, размещения в нужных контейнерах и настройки. Информационная форма, представленная схемами радикалов, должна быть удобна для эксплуатации, контроля, модификации, демонтажа и хранения.

Единая система информационных форм — это специальное представление модели проблемной области в среде радикалов **с учетом принципа ее достаточной целостности** как совокупности связанных между собой информационных форм, как результат реализации радикальной модели с помощью информационных форм. Одновременно единая система информационных форм представлена как часть такой модели автоматизированных систем различного назначения. Среда радикалов, радикальная модель проблемной области должна быть представлена с помощью единой системы информационных форм в рамках математической стандартизации проблемной области. С помощью единой системы информационных форм должны моделироваться все составляющие проблемной области автоматизированных систем, все их

свойства и связи (как в статике, так и в динамике), должны быть реализованы (с применением программно-технических средств) все виды обеспечения автоматизированных систем, все их компоненты. Все манипуляции с данными и знаниями в среде радикалов проблемной области, все их преобразования должны быть представлены и осуществимы с помощью единой системы информационных форм и программно-технических средств автоматизированных систем. Единая система информационных форм должна обеспечивать охват всех значимых ситуаций, аспектов, типов задач проблемной области автоматизированных систем, в том числе следующих:

- *работа с исходными радикалами*, например, списком объектов обработки (поиск, проверка на наличие ошибок согласно имеющимся правилам-ультраконтейнерам);
- *определение доступных объектов*, обработка которых возможна (при учете имеющихся ресурсов и ограничений);
- *согласование формальных описаний*, генерируемых участниками работ.

Частные модели проблемной области автоматизированных систем могут учитывать различные ее аспекты, включая: дискретность и непрерывность; статический и динамический характер; упругость и пластичность в среде окружения; вероятность; нечеткость, а также автоматизированную обучаемость. Эти, а также все другие значимые аспекты моделей должны обеспечиваться с помощью сертифицированных информационных форм из состава единой системы информационных форм, реализованных с помощью программно-технических средств и принадлежащих базе знаний автоматизированных систем.

3.2. Примеры некоторых схем радикалов применительно к космической съемке

Приведем краткий перечень некоторых радикалов проблемной области космической съемки, сформированный на практике в результате ее исследования.

Контейнеры, связывающие систему космической съемки в целом с ее надсистемой; а также систему космической съемки в целом и отдельные ее подсистемы верхнего уровня.

Контейнер подсистем верхнего уровня, связывающий главную подсистему (подсистему управления), рабочую подсистему (непосредственно обеспечивающую решение целевой задачи), подсистему помощи, подсистему архива, подсистему обучения.

Главная подсистема осуществляет: формирование описаний постановок задач, настройку и активацию подсистем (соответствующих запускаемых

модулей), протоколирование и анализ процессов верхнего уровня системы, анализ сформированной рабочей подсистемой и представленного с помощью схемы радикалов решения поставленной задачи, отправку решения надсистеме.

Рабочая подсистема осуществляет: формирование и анализ исходных данных, используемых задачей; решение задачи, сопровождающееся протоколированием и анализом соответствующего процесса; формирование итоговой схемы радикалов — решения поставленной задачи.

Таким образом, в *составе единой системы информационных форм* необходимы информационные формы, предназначенные для постановок задач, для их решения и протоколирования соответствующих процессов. Для функционирования рабочей подсистемы необходимы связываемые контейнерами запускаемые модули, различные источники и приемники данных. Необходимы файлы данных, в том числе — базы данных, их псевдонимы и таблицы, поля таблиц и их характеристики, процедуры для работы с данными (с привязкой к запускаемому модулю и событиям, соответствующим этим процедурам). Необходимы также контейнеры, связывающие головные и вспомогательные таблицы. *Все это должно быть представлено стандартно схемами радикалов с помощью единой системы информационных форм.*

Приведем последовательность схем, направленную от начальных описаний проблемной области автоматизированной системы к описаниям последующих уровней (и, в итоге, к решению поставленной задачи). Рассмотрение будет упрощенным. Итак, вначале будем считать, что на верхнем уровне требуемые функциональные возможности будут обеспечиваться подсистемами управления и рабочей подсистемой, а также подсистемами помощи, архива и обучения (рис. 5).

Рассмотрим наиболее важные подсистемы — подсистему управления uSysControl и подсистему uSysWork. Их объединяет контейнер cSysSys, в котором: по направлению 1 доступна активирующая

c	cSysSys	0
u	uSysControl	1
u	uSysWork	2
u	uSysHelp	
u	uArchive	
u	uSysTraining	

Рис. 5. Подсистемы автоматизированной системы

c	cExeExe			0
c	cSysExe	0	0	
u	uSysControl	1		
u	uSysWork		1	
u	uExeControl	2		1
u	uExeWork		2	2

Рис. 6. Подсистемы и запускаемые модули

подсистема uSysControl, а по направлению 2 — активируемая подсистема uSysWork.

Уточнять описание подсистем uSysHelp, uSysArchive и uSysTraining пока не будем.

С помощью контейнеров cSysExe и cExeExe уточним уникамы uSysControl и uSysWork, введя в рассмотрение уникамы — запускаемые модули (рис. 6).

Контейнеры cSysExe объединяют подсистему uSysControl и программно реализующий ее модуль uExeControl, а также подсистему uSysWork и ее модуль.

Кроме того, имеется контейнер cExeExe, в который вложены запускающий модуль uExeControl (доступен в контейнере по направлению 1) и запускаемый модуль uExeWork (доступен по направлению 2).

Модуль uExeWork должен обеспечить работу с базами данных, которые необходимы для решения целевой задачи возможности съемки объектов наблюдения (рис. 7).

Здесь в контейнер источника и приемника данных cSourceReceiver вложены:

- по направлению 1 — уникам запускаемого модуля uExeWork обеспечения работы с базами данных;
- по направлению 2 — уникам источника данных — базы данных uDBOut;
- по направлению 3 — уникам приемника данных — базы данных uDBWork.

Пусть известен состав таблиц рассматриваемых баз данных и их полей. Тогда состав источников и приемников данных можно уточнить (рис. 8, поля таблиц не показаны).

Здесь первый экземпляр контейнера cDbTb базы данных и ее таблиц объединяет базу данных

c	cSourceReceiver	0
u	uExeWork	1
u	uDBOut	2
u	uDBWork	3

Рис. 7. Запускаемый модуль, базы данных — источник и приемник данных

c	cSourceReceiver			0
c	cDbTb	0	0	
u	uExeWork			1
u	uDBOut	1		
u	uDBWork		1	
u	uTBOut1	2		
u	uTBOut2	3		
u	uTBOut3	4		2
u	uTBWork1		2	
u	uTBWork2		3	3

Рис. 8. Запускаемый модуль, таблицы — источник и приемник данных

uDBOut (направление 1) и ее таблицы — уникамы uTBOut1, uTBOut2 и uTBOut3 (направления 2, 3, 4).

Второй экземпляр контейнера cDbTb объединяет базу данных uDBWork (направление 1) и ее таблицы — уникамы uTBWork1 и uTBWork2 (направления 2, 3).

Экземпляр контейнера cSourceReceiver связывает с помощью запускаемого модуля uExeWork (направление 1) источник данных — таблицу uTBOut3 (направление 2) и приемник данных — таблицу TBWork3 (направление 3).

Для активации запускаемого модуля uExeWork необходимо совершение некоторого события, обозначенного как uEventSmth (рис. 9).

Здесь в контейнер cEventSourceReceiver вложены следующие радикалы:

- по направлению 1 — уникам — событие uEventSmth, которое пока «полностью» не определено;
- по направлению 2 — уникам — запускаемый модуль uExeWork;
- по направлению 3 — таблица — источник данных uTBOut3;
- по направлению 4 — таблица — приемник данных uTBWork2.

В следующей статье авторы планируют рассмотреть приведенный выше пример более подробно.

c	cEventSourceReceiver	0
u	uEventSmth	1
u	uExeWork	2
u	uTBOut3	3
u	uTBWork2	4

Рис. 9. Событие, запускаемый модуль, таблицы — источники и приемники данных

Заключение

В настоящей статье на концептуальном уровне описано состояние исследований, направленных на внедрение в практику нового подхода, основанного на идее использования схем радикалов для сопровождения систем автоматизации процессов управления сложными объектами на всех этапах их жизненного цикла.

Выделены наиболее важные принципы интеллектуализации автоматизированных систем: *учет разнохарактерного целеполагания, принцип необходимой избыточности, принцип достаточной целостности.*

Опыт практического применения языка RADICAL как универсального средства для описания схем взаимодействия радикалов в такой среде, полученный при решении задач автоматизации управления приборной базой космических аппаратов в разных целях, позволяет утверждать об эффективности предлагаемого подхода, о необходимости разработки и развития единой постоянно расширяющейся библиотеки стандартных информационных форм, предназначенных для решения отмеченных в статье, востребованных на практике задач. Такая расширяемая библиотека информационных форм (библиотеки радикалов) должна заложить основу развивающейся базы знаний нового подхода в условиях быстро прогрессирующего разнообразия программно-технических средств.

Список литературы

1. **Липаев В. В.** Программная инженерия. Методологические основы. М.: ТЕИС, 2006. 608 с.
2. **Васенин В. А., Кривчиков М. А.** Формальные модели программ и языков программирования. Часть 1. Библиографический обзор 1930—1989 гг. // Программная инженерия. 2015. № 5. С. 10—19.

3. **Васенин В. А., Кривчиков М. А.** Формальные модели программ и языков программирования. Часть 2. Современное состояние исследований // Программная инженерия. 2015. № 6. С. 24—33.
4. **Пак Дж. С., Якобсон И., Майбург Б., Джонсон П.** SEMAT вчера, сегодня и завтра: перспективы промышленного использования // Программная инженерия. 2014. № 11. С. 6—16.
5. **Змеев Д. О., Змеев О. А., Иванова Л. С.** Практика работы с антипаттернами для Essence Practice Library // Программная инженерия. 2022. Том 13, № 7. С. 311—321. DOI: 10.17587/prin.13.311-321.
6. **Позин Б. А.** SEMAT — Software Engineering Method and Theory. О чем, зачем и кому это нужно? // Программная инженерия. 2014. № 11. С. 3—5.
7. **Пирогов М. В.** Радикальное программирование // Программная инженерия. 2013. № 4. С. 2—15.
8. **Васенин В. А., Пирогов М. В., Чечкин А. В.** Радикальное моделирование и инженерия сложных программных систем // Программная инженерия. 2014. № 10. С. 3—10.
9. **Васенин В. А., Пирогов М. В., Чечкин А. В.** Информационно-системная безопасность критических систем: монография. М.: КУРС, 2018. 352 с.
10. **Чечкин А. В.** Математическая информатика. М.: Наука. 1991. 412 с.
11. **Соболева Т. С., Чечкин А. В.** Дискретная математика. Углубленный курс. М.: ИНФРА-М: Курс, 2016. 276 с.
12. **Ершов А. П.** Предварительные соображения о лексиконе программирования // Избранные труды. Новосибирск. 1994. С. 395—406.
13. **Котов В. Е.** Расширяющаяся вселенная информатики // Андрей Петрович Ершов — ученый и человек. Новосибирск. 2006. С. 158—166.
14. **Чечкин А. В., Пирогов М. В.** Лексикон программирования А. П. Ершова и моделирующая среда широкой проблемной области целенаправленных систем в форме среды радикалов // Нейрокомпьютеры: разработка, применение. 2016. № 11. С. 3—14.
15. **Галатенко В. А., Костюхин К. А., Левченкова Г. Л.** Целостность как аспект информационной безопасности: обзор современных подходов // Программная инженерия. 2021. Том 12, № 8. С. 420—424. DOI: 10.17587/prin.12.420-424.
16. **ГОСТ 34.003—90.** Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
17. **Потюпкин А. Ю., Чечкин А. В.** Искусственный интеллект на базе информационно-системной избыточности: монография. М.: КУРС, 2019. 384 с.
18. **Колобов А. Ю., Пирогов М. В., Рожков В. В.** Принципы и основные схемы радикалов для построения базы данных системы планирования и управления работой целевой аппаратуры космического аппарата // Вестник НПО им. С. А. Лавоочкина. 2021. № 2/52. С. 73—79. DOI: 10.26162/LS.2021.52.2.010.

Three Principles of Intellectualization of Automated Systems

V. A. Vasenin, Professor, vasenin@msu.ru, Lomonosov Moscow State University, Moscow, 119991, Russian Federation,

A. V. Chechkin, Honorary Professor, A.Chechkin@mail.ru, Military Academy of the RVSN named after Peter the Great, Balashikha, 143900, Russian Federation,

M. V. Pirogov, Engineer of the 2nd category, pmv_mvp@mail.ru, S. A. Lavochkin NGO, Khimki, 141402, Russian Federation

Corresponding author:

Alexsandr V. Chechkin, Honorary Professor, Military Academy of the RVSN named after Peter the Great, Balashikha, 143900, Russian Federation

E-mail: A.Chechkin@mail.ru

Received on February 16, 2023

Accepted on April 12, 2023

The purpose of the research, the results of which are presented in the article, is to present the conceptual provisions of the new modeling and programming technology, called radical. It is intended for all stages of the life cycle of complex automated systems using the concept of a broad problem area and the methodology of intellectualization. Within the framework of the new radical technology, three principles of intellectualization are considered: diverse goal setting; necessary information and system redundancy; sufficient integrity in the provision of automation tools. Radical technology is based on mathematical objects — radicals and radical schemes, with the help of which the modeling of the problem area of complex automated systems should be carried out. The issues of standardization of technology for the creation and maintenance of automated systems based on the use of radical schemes are discussed. The necessity of using such schemes in the life cycle of an automated system is substantiated. The provisions of the proposed technology are briefly illustrated on the basis of the experience of their use in the development of programs for the operation of the target equipment of spacecraft. The issues of software development for automation of control processes of permanently developing and increasingly complex objects have always been and remain in the focus of attention of specialists in the field of computer science. A special place in the list of such objects is occupied, for example, by complexly organized systems of the aerospace complex. Due to the need to automate and intellectualize the processes of maintaining such a class of objects in a real environment, a new methodology and technology of software development and maintenance engineering is being developed in Russia and other countries.

The problem area of complex automated systems, which include, *inter alia*, space systems, covers a large number of diverse components of such systems, their properties and relationships. This class of automated systems is characterized by: variability; expansion of functionality; the need for the use of rapidly developing software and hardware for various purposes. The problem area of automated systems is characterized by both significant achievements and significant shortcomings. Disadvantages can (and do) lead to large-scale, irreparable losses, accidents and catastrophes. The presence of problematic issues of verification and ensuring a high degree of reliability, difficulties of modification and the possibility of semantic errors — all this is characteristic of modern automated systems, for their software and hardware.

The reason for the disadvantages of complex automated systems is the widespread use of technologies in their development that do not adequately cover the stages of pre-project research, including modeling. The old approach is characterized by situations in which the problem area of an automated system, its components and their connections are not fully described. Hence the objective difficulties of software development and modification. The documentation of such software tools is characterized by the widespread use of information representation based on natural language. Automation of the processes of development and maintenance of such documentation, assessment of its completeness, correctness is significantly difficult. Mathematical formalization of the problem area of automated systems, as a rule, is insufficient. In this article, a new mathematical standardization of the formal description of the problem domain of automated systems and processes occurring in them, uniformly described on the basis of the concept of radical, is proposed. The most well-known of the approaches based on the methodology of software engineering applied to automated systems is the SEMAT (Software Engineering Method and Theory) project. The core and the domain model description language make up the basic part of the project, called Essence. An essential requirement for its components is the requirement of the possibility of an adequate and visual description for developers and an adequate (objective) assessment of the state of the target software product at all stages of its production. The project proposed in this article, based on the model of radical programming and radical modeling, is an analogue of the SEMAT/Essence project noted above. The model considered in this article is based on a uniform mathematically formalized standardization of the description of any problem area and its environment. The so-called radicals and the mechanisms of their interaction during the implementation of automated processes are considered as the basic atomic elements that make up the core of our approach. Within the framework of this model, a special modeling language, RADICAL, has been developed, taking into account the redundancy of the model description. With the help of such a language, an expressive, flexible and adequate representation of the problem area is provided, visually convenient for the developer to assess the condition of the target product at different stages of its life cycle.

The development and improvement of methods for mathematically rigorous description of automated systems and their environment based on such technology is a strategically important task. This publication attempts to describe the current state of exploratory research and applied work aimed at developing a project called «radical modeling and radical programming». The material presented in the publication, including a summary of the results in relation to the automation of objects in the aerospace field, can also be considered as an invitation to new researchers to participate in the work on the development of the project. The experience of practical application of the RADICAL language, as a universal one for describing schemes of interaction of radicals in such an environment, obtained when solving problems of automation of control of the instrument base of spacecraft for various purposes, suggests the urgent need to develop and develop a single, constantly expanding library of standard information forms designed to solve different classes of tasks. Such an expandable library of information forms (libraries of radicals) should lay the foundation for a developing knowledge base of radical technology in the context of a rapidly progressing variety of software and hardware.

Keywords: software, information support for the intellectualization of technical systems, information and system security of an automated system, satellite imagery, radical environment, radical scheme, radical modeling, radical programming, RADICAL language, information form

For citation:

Vasenin V. A., Chechkin A. V., Pirogov M. V. Three Principles of Intellectualization of Automated Systems, *Programmnaya Ingeneriya*, 2023, vol. 14, no. 6, pp. 263—277. DOI: 10.17587/prin.14.263-277.

References

1. **Lipaev V. V.** *Software engineering. Methodological foundations*, Moscow, THEIS, 2006, 608 p. (in Russian).
2. **Vasenin V. A., Krivchikov M. A.** Formal Models of Programming Languages and Programs. Part 1. Literature Review: 1930–1989, *Programmnaya Ingeneria*, 2015, no. 5, pp. 10–19 (in Russian).
3. **Vasenin V. A., Krivchikov M. A.** Formal Models of Programming Languages and Programs. Part 2. Present State of Research, *Programmnaya Ingeneria*, 2015, no. 6, pp. 24–33 (in Russian).
4. **Pak J. S., Jacobson I., Mayburg B., Johnson P.** SEMAT Yesterday, Today and Tomorrow. An Industrial Perspective, *Programmnaya Ingeneria*, 2014, no. 11, pp. 6–16 (in Russian).
5. **Zmeev D. O., Zmeev O. A., Ivanova L. S.** Antipattern Practice for Essence Practice Library, *Programmnaya Ingeneria*, 2022, vol. 13, no. 7, pp. 311–321. DOI: 10.17587/prin.13.311-321 (in Russian).
6. **Posin B. A.** SEMAT — Software Engineering Method and Theory. About what, what for and to whom is it addressed? *Programmnaya Ingeneria*, 2014, no. 11, pp. 3–5 (in Russian).
7. **Pirogov M. V.** Radical Programming, *Programmnaya Ingeneria*, 2013, no. 4, pp. 2–15 (in Russian).
8. **Vasenin V. A., Pirogov M. V., Chechkin A. V.** Radical Modeling and Engineering of Complex Program System, *Programmnaya Ingeneria*, 2014, no. 10, pp. 3–10 (in Russian).
9. **Vasenin V. A., Pirogov M. V., Chechkin A. V.** *Information and system security of critical systems*: monograph, Moscow, COURSE, 2018, 352 p. (in Russian).
10. **Chechkin A. V.** *Mathematical informatics*, Moscow, Nauka, 1991, 412 p. (in Russian).
11. **Soboleva T. S., Chechkin A. V.** *Discrete mathematics*. Advanced course, Moscow, INFRA-M: Course, 2016, 276 p. (in Russian).
12. **Ershov A. P.** Preliminary considerations on the programming lexicon, *Selected works*, Novosibirsk, 1994, pp. 395–406 (in Russian).
13. **Kotov V. E.** The expanding universe of computer science, *Andrey Petrovich Ershov is a scientist and a man*, Novosibirsk, 2006, pp. 158–166 (in Russian).
14. **Chechkin A. V., Pirogov M. V.** The programming lexicon of A. P. Ershov and the modeling environment of a wide problem area of targeted systems in the form of a radical environment, *Neurocomputers: development, application*, 2016, no. 11, pp. 3–14 (in Russian).
15. **Galatenko V. A., Kostyukhin K. A., Levchenkova G. L.** Integrity as an Aspect of Information Security: an Overview of Modern Approaches, *Programmnaya Ingeneria*, 2021, vol. 12, no. 8, pp. 420–424. DOI: 10.17587/prin.12.420-424 (in Russian).
16. **GOST 34.003–90.** Information technology. A set of standards for automated systems. Automated systems. Terms and definitions (in Russian).
17. **Potyupkin A. Yu., Chechkin A. V.** *Artificial intelligence based on information-system redundancy*: monograph, Moscow, COURSE, 2019, 384 p. (in Russian).
18. **Kolobov A. Yu., Pirogov M. V., Rozhkov V. V.** Principles and basic schemes of radicals for building a database of the system of planning and control of the operation of the target equipment of the spacecraft, *Bulletin of S. A. Lavochkin NGO*, 2021, no. 2/52, pp. 73–79. DOI: 10.26162/LS.2021.52.2.010 (in Russian).

ИНФОРМАЦИЯ

Продолжается подписка на журнал «Программная инженерия» на второе полугодие 2023 г.

Оформить подписку можно через подписные агентства
или непосредственно в редакции журнала (для юридических лиц).

Подписной индекс по Объединенному каталогу

«Пресса России» — 22765

Сообщаем, что с 2020 г. возможна подписка
на электронную версию нашего журнала:

ООО «ИВИС»: тел. (495) 777-65-57, 777-65-58; e-mail: sales@ivis.ru;
ООО «УП Урал-Пресс Округ». Для оформления подписки (индекс 013312)
следует обратиться в филиал по месту жительства — <http://ural-press.ru>

Адрес редакции: 107076, Москва, Матросская Тишина, д. 23, с. 2, оф. 45,

Издательство «Новые технологии»,
редакция журнала «Программная инженерия»

Тел.: (499) 270-16-52. E-mail: prin@novtex.ru

И. М. Гостев, д-р техн. наук, вед. науч. сотр., igostev@gmail.com,
Институт проблем передачи информации имени А. А. Харкевича РАН, Москва,
П. Е. Голосов, канд. техн. наук, декан, pgoosov@gmail.com,
Российская академия народного хозяйства и государственной службы при Президенте
РФ (РАНХиГС), Москва

Анализ эффективности облачной вычислительной системы, обслуживающей поток заданий с директивными сроками выполнения при множественных отказах серверов

Поступила в редакцию 28.03.2023

Принята к публикации 11.04.2023

В настоящее время использование распределенных (облачных) ресурсов приобретает все более широкое распространение. Несмотря на большую надежность оборудования, применяющегося в кол-центрах, отказы случаются. Если системы долговременного хранения имеют высокую надежность за счет многократного дублирования, то отказ сервера, на котором выполняется задача, в специализированных вычислительных системах может соответствовать отказу всей системы. В целях устранения такого отказа исследована имитационная модель специализированной вычислительной системы, построенной в среде Simulink (SimEvent), классифицируемая в теории массового обслуживания как $G/G/n/\infty$. Она характеризуется множеством входных потоков с бесконечной очередью, имеет две обратные связи, отражающие ситуации повторной обработки в случае отказа сервера или отсутствия решения при первой попытке обслуживания. Архитектура системы ориентирована на параллельное выполнение определенного класса задач, разделяемых на подзадачи, независимые по данным. Главным критерием функционирования такой системы является прохождение всех поступивших задач за время, не превышающее директивный срок их выполнения. Целью работы, результаты которой представлены в статье, является исследование надежности функционирования такой системы при различных значениях вероятности отказа серверов. Поскольку производительность и эффективность работы такой системы уже были исследованы в предыдущих работах авторов, то основное направление исследований будет сосредоточено на влиянии отказов серверов на общую продуктивность и эффективность работы при различных законах входных потоков.

Ключевые слова: имитационное моделирование, облачные и распределенные системы, планирование, эффективность выполнения, директивный срок выполнения, отказы серверов

Для цитирования:

Гостев И. М., Голосов П. Е. Анализ эффективности облачной вычислительной системы, обслуживающей поток заданий с директивными сроками выполнения при множественных отказах серверов // Программная инженерия. 2023. Том 14, № 6. С. 278—284. DOI: 10.17587/prin.14.278-284.

Введение

В настоящей статье рассмотрена специализированная вычислительная система (СВС), решающая ресурсоемкие задачи, относящиеся к специальному классу, методы выполнения которых можно определить как случайный перебор с неизвестным

исходом [1, 2]. Получение решения такой задачи основано на переборных алгоритмах и сводится к поиску фрагмента с наперед заданными свойствами в большом массиве исходных данных. В работах [1, 2] такой специализированный класс задач именуется как *citu*-задания, *citu*-задачи; от английского термина *computation* — *intensive task*

under uncertainty. Следует отметить целый ряд существенных особенностей, присущих данному классу: выполнение всех situ-задач должно происходить в режиме реального времени. Они допускают распараллеливание по данным и являются ресурсоемкими. Их решения должны быть получены как можно быстрее и в отдельности, и в совокупности.

В настоящей публикации анализируются результаты отказов серверов в имитационной модели высокопроизводительной СВС, выполняющей задачи такого типа. Существуют примеры успешных реализаций таких систем, исследованные в работах [3, 4]. Однако по ряду причин они не получили широкого применения для решения такого класса задач. Тем не менее, поскольку класс таких задач постоянно расширяется, вопрос получения решения за директивное время выполнения (ДВВ) при возможных отказах серверов остается актуальным.

Принципы управления такими задачами были исследованы авторами неоднократно в работах [5–9], однако вопросы надежности работы таких систем оставались без внимания.

Здесь необходимо отметить, что вопросы надежности работы вычислительной техники многократно рассматривались в литературе [10–13]. В классическом варианте при исследовании отказов обычно рассматривают такие характеристики, как надежность серверов, линий передачи информации, бесперебойных источников питания и т. п., с точки зрения различных метрик, например, MTBF (*mean time between failures* — среднее время между отказами) или MTTR (*mean time to repair* — среднее время восстановления, исправления, реагирования или устранения и др.), изложенных в многочисленных стандартах, например [14, 15].

Однако в данной модели все серверы совершенно равноправны. Отказ одного из них приводит к повторному выполнению подзадачи на другом сервере. Поэтому отказ одного сервера при выполнении одной из подзадач незначительно влияет на работу всей модели. Поскольку в нашем случае исследуется имитационная модель СВС, для которой чрезвычайно сложно построить аналитическое описание, то применение каких-либо аналитических методик оценки надежности не представляется возможным. Более того, так как все серверы в такой системе одинаковы по своим параметрам, отказ одного или нескольких из них не приводит к отказу всей системы до тех пор, пока каждая задача выполняется за время, не превышающее ДВВ.

Постановка задачи

Для проведения исследований в данной имитационной модели специально разработан механизм искусственного введения отказов. Этот механизм предусматривает следующие действия:

- при возникновении отказа сервера в процессе выполнения некоторой подзадачи она незамедлительно возвращается в очередь на повторное выполнение;
- серверу, на котором возник отказ, необходимо некоторое время для перезагрузки, в течение которого он заблокирован для приема задач на выполнение;
- после перезагрузки сервер продолжает принимать и обрабатывать поступающие задачи;
- отказы серверов в настоящей модели случаются по закону равномерного распределения в интервале $[0, 1]$ и наступают при превышении некоторого порога (параметра), определяющего надежность работы сервера, например, при значении порога 0,9 сервер считается вышедшим из строя при генерации случайного числа (параметра), превышающего этот порог.

Таким образом, целью настоящей работы является исследование влияния отказов одного или более серверов на изменение производительности и эффективности работы данной модели при различных законах входных потоков задач на основе их архитектуры и алгоритмов функционирования данной модели.

При этом необходимо выполнение ряда формальных условий:

- все задачи должны строго выполняться в рамках заданного ДВВ;
- задачи (сразу после генерации) разделяются на несколько подзадач и выполняются параллельно на нескольких вычислителях (подзадачи независимы по данным);
- функционирование системы возможно при множестве входных потоков подзадач с различными законами их поступления (равномерный, экспоненциальный, пуассоновский и т. д.);
- эффективность работы такого комплекса должна быть максимальной, т. е. желательно, чтобы все серверы были загружены на 100 %;
- в случае нахождения решения задачи в одной из подзадач выполнение всех остальных подзадач должно быть прекращено;
- в случае отсутствия решения некоторой задачи при заданных условиях она должна быть автоматически перезапущена с измененными начальными условиями;
- при отказе одного из серверов при решении некоторой подзадачи она должна быть выполнена повторно.

Архитектура модели

Структура и алгоритм функционирования имитационной модели облачной системы уже были неоднократно рассмотрены в работах [7–9]. Модель была разработана специально для решения задач изложенного выше типа и схематично представлена на рис. 1.

Основные компоненты модели:

- входной поток задач различного типа и случайных интервалов поступления задач;
- очередь задач на выполнение — Queue;
- 32 сервера, выполняющих задачи;
- подсистема управления продвижением подзадач — Control;
- две обратные связи, возникающие при отказе сервера и при отсутствии решения;
- подсистема Prod, предназначенная для определения производительности и эффективности работы модели.

Подробно все компоненты этой системы были изложены в работах [8, 9]. Для оценки эффективности работы имитационной модели был разработан блок Prod, отсутствующий в предыдущих работах (рис. 2).

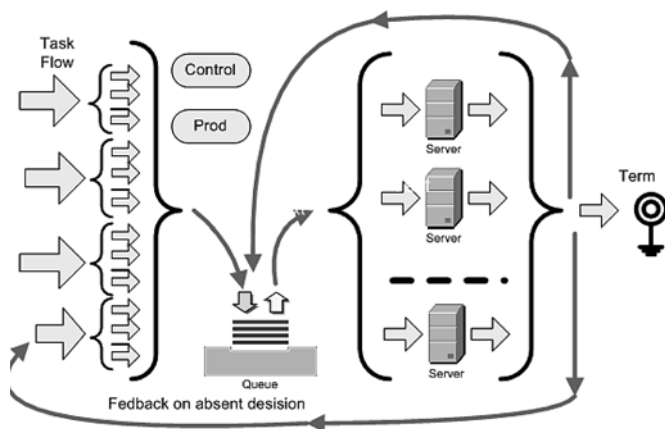


Рис. 1. Укрупненная схема имитационной модели СВС

Рассмотрим состав блока Prod, представленного на рис. 2:

- генератор Group 1, определяющий интервал, в котором будут измеряться производительность и эффективность;
- импульсный генератор Pulse Generator1, по сигналам которого происходят измерения;
- блок вызова функции Function Caller2, который по сигналу генератора вызывает функцию CalcEff(), которая, в свою очередь, и вычисляет производительность системы (более подробно механизм изложен ниже).

Дисплей res, начиная с 300-й секунды отображает вычисленные значения на «скользящем» интервале времени в 300 с.

Для введения численного параметра производительности системы введем некоторые определения. Предположим, что имеется N параллельно работающих независимых серверов. Пусть на входе системы имеется поток задач, каждая из которых имеет единичное время обслуживания. Тогда в единицу времени можно обслужить N задач, а за промежуток времени t число решенных задач будет Nt . Таким образом, можно определить среднюю продуктивность системы как сумму ДВВ решенных задач за некоторый промежуток времени. В данном случае использовался промежуток времени, равный 300 единицам условного времени (далее будем считать их секундами). Тогда, например, для рассматриваемой модели продуктивность при 32 серверах за 300 с будет составлять 9600 условных единиц продуктивности.

Вместе с тем эффективность работы системы с жестко заданными требованиями будет определяться отношением числа всех невыполненных задач к числу выполненных за ДВВ. Будем рассчитывать такую эффективность в процентах. Выполнение полностью всех задач в ДВВ за некоторый (большой временной интервал, достаточный для установления равновесного режима работы, при котором в очереди находится приблизительно

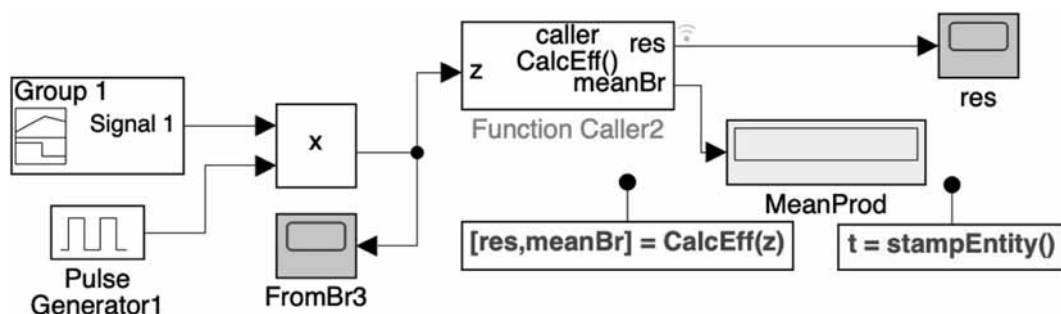


Рис. 2. Схема блока расчета продуктивности

равное число подзадач за некоторый промежуток времени) период моделирования будем считать 100 %-ной эффективностью работы. Если, например, за этот период из 100 задач 10 не уложились в ДВВ, то эффективность работы такой системы составляет только 90 %.

Вычисление продуктивности в блоке Prod осуществляется функцией CalcEff() через промежуток в 10 с на смещаемом по временной оси интервале в 300 с как сумма ДВВ всех выполнившихся задач на этом интервале. Индикатор MeanProd отражает итоговое среднее значение продуктивности на всем интервале моделирования.

Результаты моделирования

Для моделирования были выбраны четыре типа задач со временем выполнения, равным 160, 100, 64 и 32 условных единиц (далее — секунд). В качестве интервала моделирования был взят промежуток длительностью 1500 с. Интервал моделирования был увеличен втрое по сравнению с интервалом в работе [9]. Это было сделано, чтобы процесс моделирования проходил в некотором «условно равновесном» состоянии, определяемом отсутствием возрастания числа подзадач в очереди. При этом все генераторы работали только в промежутке времени 0...1400. Были проведены две серии экспериментов. В одной из серий использовался экспоненциальный закон распределения генерируемых задач, во второй серии — равновероятный закон.

В каждой серии для задания вероятности отказов работы серверов использовалась глобальная переменная Disrep, которая принимала одно из значений (0,997, 0,9, 0,8, 0,75, 0,7). Для искусственной инициализации отказов серверов в структуру их описаний была встроена функция генератора случайных чисел с равновероятным распределением на интервале [0, 1]. При поступлении задачи на сервер происходит вызов этой функции

и результат сравнивается со значением Disrep. Если значение, возвращенное этой функцией, превышает Disrep, то происходит отказ сервера. В этом случае подзадача повторно поступает в очередь Queue. Таким образом, искусственно вызываются отказы серверов с тем большей вероятностью, чем меньше значение Disrep.

В первой серии (экспоненциальное распределение) средние значения интервалов генерации задач были $\lambda = 14, 10, 7, 4$. Это означает, что задачи с большим временем выполнения генерировались реже, чем короткие. Во всех экспериментах этой серии была сгенерирована 1021 задача: 114, 177, 245 и 485 по типам соответственно. Результаты моделирования сведены в табл. 1, где P — значение параметра Disrep; M — общая средняя продуктивность системы; S — число задач, не выполнившихся за ДВВ; Eff — эффективность работы системы; InQ — число задач, оставшихся в очереди после окончания времени моделирования (1500); $LenQ$ — средняя длина очереди в подзадачах; $DepQ$ — общее число обслуженных подзадач; $QWait$ — среднее время ожидания подзадачи в очереди; Tm — время окончания выполнения всех сгенерированных задач; Err — число отказов серверов.

Во второй серии экспериментов с равномерным распределением входных потоков были использованы те же длительности всех типов задач и равновероятный закон распределения времени генерации задач с интервалами [0...18], [0...12], [0...8], [0...6] и общим числом 1220 (155, 237, 352, 476) соответственно.

Полученные результаты представлены в табл. 2.

Кроме этих серий были проведены два контрольных эксперимента при тех же самых условиях и значении Disrep = 0,997. Однако в этом случае были использованы всего 24 сервера. Результаты представлены в табл. 3. Строка 1 соответствует экспоненциальному распределению, строка 2 — равновероятному.

Таблица 1

Экспоненциальное распределение

P	M	S	$Eff, \%$	InQ	$LenQ$	$DepQ$	$QWait$	Tm	Err
0,997	14 762,8	0	100	0	45,5	29 557	2,155	1402	72
0,9	14 715,6	0	100	0	60,1	31 712	2,660	1416	2847
0,8	14 701,8	0	100	0	80,52	33 856	3,388	1433	5857
0,75	14 609,2	0	100	0	103,8	34 469	4,291	1431	7823
0,7	14 312,3	6 (4, 2, 0, 0)	99,4	0	160	31 617	7,299	1456	8652

Равномерное распределение

P	M	S	$Eff, \%$	InQ	$LenQ$	$DepQ$	$QWait$	Tm	Err
0,997	18 973,7	0	100	0	53,53	35 602	2,107	1407	97
0,9	18 895,6	0	100	0	90,47	36 660	3,497	1423	3423
0,8	18 677,8	0	100	0	150,7	34 824	6,154	1437	6085
0,75	18 321	6 (6, 0, 0, 0)	99,3	0	252,4	28 029	12,75	1429	6370
0,7	17 905	23 (18, 5, 0, 0)	97,7	0	303,3	25 290	17,67	1491	7032

Таблица 3

Результаты с использованием 24 серверов

№	M	S	$Eff, \%$	InQ	$LenQ$	$DepQ$	$QWait$	Tm	Err
1	14 672	0	100	0	80,18	27 822	4,064	1419	87
2	18 597	0	100	0	145,4	29 374	7,098	1439	87

Обсуждение

Проведенные серии экспериментов в дополнении к работе [10] ожидаемо показали высокую эффективность управлением прохождения задач в исследуемой модели СВС. Необходимо еще раз внести дополнительные пояснения относительно того факта, что практическая продуктивность в табл. 1—3 существенно превышает теоретическую (20000 по сравнению с 9600). Для объяснения этого необходимо напомнить механизм работы серверов. После генерации задачи она разделяется на подзадачи, число которых задается динамически на основании работы интеллектуальных агентов. При этом понятно, что положительное решение будет находиться в одной из подзадач (независимость по данным). Необходимо также напомнить, что все неуспешные подзадачи выполняются за время T/n , где T — время выполнения всей задачи; а n — число подзадач. Успешная подзадача будет выполняться с равной вероятностью в интервале $[0 \dots T/n]$. Как только такой фрагмент задачи будет выполнен, все остальные подзадачи снимаются с вычислений на серверах. Если же они стоят в очереди, то как только они поступают на сервер, то без вычислений заканчивают работу. Таким образом, по сравнению с теоретической продуктивностью практическая должна возрастать, на основании того, что время выполнения успешной подзадачи в среднем будет $T/2n$.

Теперь необходимо прокомментировать результаты, занесенные в табл. 1 и 2. Понятно, что уменьшение порога $Disrep$ будет приводить к возрастанию числа ошибок серверов, а это приведет к увеличению средней длины очереди и времени ожидания подзадач в ней. Эксперименты показывают, что средняя

продуктивность работы системы при уменьшении $Disrep$ до некоторого значения остается постоянной. При значении $Disrep = 0,75$ и ниже имитационная модель практически перестает соответствовать своим основным требованиям, поскольку число задач с превышением ДВВ сильно возрастает.

Таким образом, предложенная в работе [9] схема управления на основе интеллектуальных агентов (элементов искусственного интеллекта) остается устойчивой к высокому числу отказов серверов и продолжает выполнять свои функции практически до значения $Disrep = 0,75$ независимо от законов распределения входных потоков. Другими словами, система сохраняет свою работоспособность при практическом отказе четверти серверов (25 %).

На основании представленных выше результатов фактически определена максимальная производительность системы в установившемся режиме, независимо от типа распределений входных потоков.

Таким образом, имеем некоторую числовую характеристику имитационной модели, которая может оценить ее предельную нагрузку при некоторой интенсивности входных потоков. Этот результат можно считать достаточно убедительным, поскольку для данной системы крайне затруднительно предложить аналитическое описание.

Поскольку теоретически невозможно установить максимальную производительность имитационной модели облачной (распределенной) системы в установившемся режиме работы, т. е. режиме, в котором достигается максимальная продуктивность, однако при этом эффективность системы остается равной 100 %, то можно рекомендовать методику, основанную на умышленном введении отказов серверов и фиксации момента времени, при котором система перестает работать (момент не вы-

полнения ДВВ). При этом после определения максимальной эффективности можно использовать такое количество серверов, которое необходимо для максимальной эффективности работы некоторой реальной СВС при условии выполнения всех задач за ДВВ.

Предложенную методику оценки производительности систем можно применять и для других имитационных моделей сетей массового обслуживания, как с различными законами поступления входных заданий, так и с различными временами обслуживания на основании искусственного механизма введения отказов.

Использование такой методики позволяет определить количество оборудования, необходимое для наиболее эффективного использования СВС при максимально возможных входных потоках с полным сохранением их работоспособности.

Заключение

На основании проведенного исследования можно сделать следующие выводы:

- разработан метод определения максимальной эффективности работы облачной (распределенной) СВС на основе искусственно вводимых ошибок серверов;
- исследования данной имитационной модели в условиях искусственного введения ошибок подтвердили, что предложенный в работе [9] тип управления заданиями на основе интеллектуальных агентов является высокоэффективным;
- поскольку данная имитационная модель построена на основе сетей массового обслуживания, то предложенный метод оценки производительности моделей можно рекомендовать в других направлениях исследований, таких как коммуникационные и телекоммуникационные сети, системы управления производством, логистические цепи и др.

Список литературы

1. Малащенко Ю. Е., Назарова И. А. Модель управления разнородными вычислительными заданиями на основе гарантированных оценок времени выполнения. // Изв. РАН. ТиСУ. 2012. № 4. С. 29–38. DOI: 10.1134/S1064230712040090.

2. Купалов-Ярополк И. К., Малащенко Ю. Е., Назарова И. А., Ронжин А. Ф. Методы оценки эффективности и директивных сроков выполнения ресурсоемких вычислительных заданий // Информатика и ее применение. 2013. Том 7, № 2. С. 17–25.

3. Баранов А. В., Киселев А. В., Корнеев Б. В. и др. Программный комплекс «Пирамида» организации параллельных вычислений с распараллеливанием по данным // Тр. междунар. суперкомпьютерной конф. и конф. молодых ученых «Научный сервис в сети Интернет: Суперкомпьютерные центры и задачи». М.: МГУ, 2010. С. 299–302.

4. Забродин А. В., Левин В. К., Корнеев В. В. Массово-параллельные системы МВС-100 и МВС-1000. // Сб. тр. науч. сессии МИФИ — 2000. М.: МИФИ, 2000. Т. 2. С. 194–195.

5. Brucker P. Scheduling Algorithms. 5th ed. Springer-Verlag Berlin and Heidelberg, 2007. 383 p.

6. Голосов П. Е., Гостев И. М. О некоторых имитационных моделях планировщиков операционных систем // Телекоммуникации. 2021. № 6. С. 10–21.

7. Голосов П. Е., Гостев И. М. Об имитационном моделировании функционирования операционной системы с вытесняющим планированием // Телекоммуникации. 2021. № 8. С. 2–22.

8. Голосов П. Е., Гостев И. М. Имитационное моделирование серверов с прерываниями в больших многопроцессорных системах // Известия вузов. Приборостроение. 2021. Том 64, № 11. С. 879–886. DOI: 10.17586/0021-3454-2021-64-11-879-886.

9. Golosov P. E., Gostev I. M. About one cloud computing simulation model // Systems of Signals Generating and Processing in the Field of on Board Communications, Conference Proceedings. 2021. P. 9416100. DOI: 10.1109/IEEECONF51389.2021.9416100.

10. Голосов П. Е., Гостев И. М. Имитационная модель облачных вычислений со спорадическим механизмом управления параллельным решением задач // Научно-технический вестник информационных технологий, механики и оптики. 2022. Том. 22, № 2. С. 269–278. DOI: 10.17586/2226-1494-2022-22-2-269-278.

11. Надежность и эффективность в технике: Справочник. В 10 т. / Ред. совет: В. С. Авдеевский и др. М.: Машиностроение, 1988.

12. Похабов Ю. П. Проблемы надежности и пути их решения при создании уникальных высокоответственных систем // Надежность. 2019. Том 19, № 1. С. 10–17. DOI: 10.21683/1729-2646-2019-19-1-10-17.

13. Рябинин И. А. Надежность и безопасность структурно-сложных систем. СПб.: Политехника, 2000. 248 с.

14. Строгонов А., Жаднов В. Обзор программных комплексов по расчету надежности сложных технических систем // Компоненты и технологии. 2007. № 5. С. 183–190.

15. ГОСТ 27.310–95. Надежность в технике. Анализ видов, последствий и критичности отказов. Доступ из справочно-правовой системы «Техэксперт». URL: <http://docs.cntd.ru/document/gost-27-310-95> (дата обращения 28.03.2023).

16. ГОСТ Р 51901.5–2005. Руководство по применению методов анализа надежности. Доступ из справочно-правовой системы «Техэксперт». URL: <http://docs.cntd.ru/document/1200041156>. (дата обращения 28.03.2023).

Analysis of the Efficiency of the Simulation Model of Cloud Computing in Case of Multiple Server Failures

I. M. Gostev, Dr. Sci, Leader Researcher, igostev@gmail.com,
Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Moscow, 127051, Russian Federation

P. E. Golosov, PhD, Dean, xasel@mail.ru, Russian Academy of National Economy and Public Administration under the Russian President (RANEPA), Moscow, 119571, Russian Federation

Corresponding author:

Ivan M. Gostev, Dr. Sci, Leader Researcher, Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Moscow, 127051, Russian Federation
E-mail: igostev@gmail.com

Received on March 23, 2023

Accepted on April 11, 2023

Currently, the use of distributed (cloud) resources is becoming more and more widespread. Despite the high reliability of equipment used in call centers, failures do occur. And if long-term storage systems have high reliability due to multiple duplication, then the failure of the server on which the task is performed in specialized computing systems (SCS) can correspond to the failure of the entire system. In order to eliminate it, a simulation model of a specialized computing system built in the Simulink (SimEvent) environment, classified in the queuing theory as $G/G/n/\infty$, is investigated. It is characterized by many input streams with an infinite queue, has two feedbacks, reflecting reprocessing situations in case of a server failure or no solution at the first service attempt. The system architecture is focused on the parallel execution of a certain class of tasks, divided into data-independent sub-tasks. The main criterion for the functioning of such a system is the completion of all received tasks in a time not exceeding the deadline for their implementation. The purpose of the work, the results of which are presented in the article, is to study the reliability of the functioning of such a system for various values of the probability of server failure. Since the performance and efficiency of such a system has already been studied in previous works, the main direction of research will be focused on the impact of server failures on overall productivity and efficiency under various laws of input flows.

Keywords: simulation, cloud and distributed systems, scheduling, execution efficiency, deadline date, server failures

For citation:

Gostev I. M., Golosov P. E. Analysis of the Efficiency of the Simulation Model of Cloud Computing in Case of Multiple Server Failures, *Programmnaya Ingeneria*, 2023, vol. 14, no. 6, pp. 278–284. DOI: 10.17587/prin.14.278-284.

References

1. **Malashenko Y. E., Nazarova I. A.** Control model for heterogeneous computational tasks based on guaranteed estimates of execution times, *J. Comput. Syst. Sci. Int.*, 2012, vol. 51, pp. 526–534. DOI: 10.1134/S1064230712040090 (in Russian).
2. **Kupalov-Jaropolk I. K., Malashenko Ju. E., Nazarova I. A., Ronzhin A. F.** Metody ocenki jeffektivnosti i direktivnyh srokov vypolnenija resursoemkih vychislitel'nyh zadaniy, *Informatika i ego primeneniye*, 2013, vol. 7, no. 2, pp. 17–25 (in Russian).
3. **Baranov A. V., Kiseljev A. V., Korneev B. V.** et al. Software complex “Pyramid” organization of parallel computing with data parallelization, *Proc. Int. Supercomputer conf. and conf. young scientists “Scientific service on the Internet: Supercomputer centers and tasks”*, Moscow, MSU, 2010, pp. 299–302 (in Russian).
4. **Zabrodin A. V., Levin V. K., Korneev B. B.** Massively parallel systems MBC-100 and MBC-1000, *Proc. Scientific Session MEPhI – 2000*, Moscow, MEPhI, 2000, vol. 2, pp. 194–195 (in Russian).
5. **Brucker P.** *Scheduling Algorithms*. 5th ed. Springer-Verlag Berlin and Heidelberg, 2007. 383 p.
6. **Golosov P. E., Gostev I. M.** About any simulation models of Schedulers Operational, *Telecommunication*, 2021, vol. 6, pp. 10–21 (in Russian).
7. **Golosov P. E., Gostev I. M.** On simulation modeling of operating system functioning with preemptive scheduling, *Telecommunication*, 2021, vol. 8, pp. 2–22 (in Russian).
8. **Golosov P. E., Gostev I. M.** Simulation of servers with Interrupt in Large Multiprocessors System, *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie*, 2021, vol. 64, no. 11, pp. 879–886. DOI: 10.17586/0021-3454-2021-64-11-879-886 (in Russian).
9. **Golosov P. E., Gostev I. M.** About one cloud computing simulation model, *Systems of Signals Generating and Processing in the Field of on Board Communications, Conference Proceedings*, 2021, pp. 9416100. DOI: 10.1109/IEEECONF51389.2021.9416100.
10. **Golosov P. E., Gostev I. M.** Cloud computing simulation model with a sporadic mechanism of parallel problem solving control, *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2022, vol. 22, no. 2, pp. 269–278 (in Russian).
11. **Reliability** and efficiency in engineering: a Handbook. In 10 v. / On Ed. Avduvski V. C. etc. Moscow, Mashinostroenie, 1988 (in Russian).
12. **Pohabov J. P.** Problems of reliability and ways to solve them when creating unique highly responsible systems, *Reliability*, 2019, no. 1, pp. 10–17 (in Russian).
13. **Rjabinin I. A.** *Reliability and safety of structurally complex systems*, SpB., Politehnika, 2000, 248 p. (in Russian).
14. **Strogonov A., Jdanov V.** Overview of software systems for calculating the reliability of complex technical systems, *Components and Technology*, 2007, no. 5, pp. 183–190 (in Russian).
15. **GOST 27.310–95.** Reliability in Technology. Analysis of failure modes, consequences and criticality. Access from the legal reference system “Techexpert”, available at: <http://docs.cntd.ru/document/gost-27-310-95> (in Russian) (date of access 28.03.2023).
16. **GOST P 51901.5–2005.** Guidance on the application of reliability analysis methods. Access from the legal reference system “Techexpert”, available at <http://docs.cntd.ru/document/1200041156> (in Russian) (date of access 28.03.2023).

И. Ю. Терёхина, асс. каф. информационной безопасности, iteryokhina@cs.msu.ru,
Московский государственный университет имени М. В. Ломоносова

Выявление аномалий с использованием построенной модели процессов в виде ациклического ориентированного графа

*Поступила в редакцию 17.04.2023
Принята к публикации 26.04.2023*

Рассмотрены подходы к решению задачи поиска аномалий при одновременном функционировании нескольких информационных технологий (процессов), типичное функционирование каждой из которых может быть описано некоторой формальной моделью. В качестве формальной модели рассмотрен ациклический ориентированный граф. Проанализированы как случай, когда при построении модели процесса в трассах лога событий может содержаться след только одного процесса, так и случай, когда в одной трассе содержатся следы нескольких процессов. Доказаны оценки сложности ответа на вопрос, является ли поступающая трасса аномальной, если формальные модели процессов уже были построены.

Ключевые слова: поиск аномалий, ациклический ориентированный граф, обнаружение процесса, модель процесса

Для цитирования:

Терёхина И. Ю. Выявление аномалий с использованием построенной модели процессов в виде ациклического ориентированного графа // Программная инженерия. 2023. Том 14, № 6. С. 285—291. DOI: 10.17587/prin.14.285-291.

Введение

Задача построения или восстановления модели для некоторой информационной технологии в литературе обозначается как задача обнаружения процесса. Данная задача не является новой, однако она все еще остается актуальной. В контексте рабочих процессов данная задача впервые изучалась в работе [1], после чего появлялось большое число различных подходов к ее решению [2—6]. В контексте обеспечения информационной безопасности задача построения модели процесса часто рассматривается совместно с задачей поиска аномалий.

Традиционно для задачи обнаружения процесса вводится понятие *одного исполнения процесса* — наблюдение за изменением атрибутов процесса в ограниченных временных рамках. Исполнению процесса ставится в соответствие формальное понятие *трассы* в некотором конечном алфавите возможных *действий* (инструкций) процесса как

наблюдение за изменением конечного подмножества атрибутов, упорядоченных по времени. После этого появляется возможность ввести определение понятия *лога* (лога событий) — конечного упорядоченного множества трасс, достаточных для адекватного описания функционирования процесса и дальнейшего восстановления математически строгого описания модели процесса.

Для задачи построения модели процесса предполагается, что лог событий эталонного процесса задан. В качестве формальной модели для описания процесса могут рассматриваться конечный автомат [2], нейронные сети [3], сеть Петри [4], ориентированный граф [5], множество ассоциативных правил [6] и др. У каждой из моделей есть свои достоинства и недостатки. Так, например, было показано, что модели в виде сетей Петри обладают большой выразительной мощностью, однако построение корректной сети Петри для лога может быть трудоемкой задачей [7]. В задаче поиска аномалий обычно происходит сопоставление новых

событий с имеющейся моделью процесса. Если событие не может быть представлено в построенной модели, то оно называется *аномальным*.

Данная работа является продолжением исследования, описанного в работе [8]. Рассматривается подход представления процессов в множестве ориентированных ациклических графов (*Directed Acyclic Graph*, DAG). Достоинствами данного подхода являются хорошая интерпретируемость полученной модели и линейная сложность построения модели относительно размера лога. Недостатком этого метода является ограничение, накладываемое на модель процесса, заключающееся в отсутствии ориентированных циклов. Это обстоятельство сужает класс логов, на которых данный метод может быть применен.

Определения

Пусть задан конечный алфавит V возможных *действий* для процесса P . *Трассой* будет называться конечное слово в алфавите V , *логом* — неупорядоченное конечное множество трасс.

Пусть задан лог событий L одного процесса P . Данный лог задает частичный предпорядок на множестве действий V : $A \lesssim_L B$ — действие B *следует* за действием A в логе L . Например, для лога $L = \{ABA, DE\}$ справедливо: $A \lesssim_L B$, $B \lesssim_L A$, $A \lesssim_L A$, $D \lesssim_L E$.

Будем считать, что действие B *зависит* от действия A в логе L ($A \rightarrow_L B$), если в логе L действие B следует за A , а A не следует за B . Если в логе L действие A следует за B и B следует за A , либо A не следует за B и B не следует за A , то будем говорить, что A и B *независимы* в логе L . Так, для лога $L = \{ABA, DE\}$ справедливо: $D \rightarrow_L E$.

Пусть задано множество действий V и лог L некоторого процесса, ориентированный граф G называется *графом зависимостей лога L* , если путь из A в B в графе G существует тогда и только тогда, когда B зависит от A в логе L .

Для заданного лога может существовать несколько графов зависимостей. В частности, графы, которые имеют одинаковые транзитивные замыкания, будут представлять собой одинаковые графы зависимостей, т. е. будут являться графами зависимостей одного и того же лога L . Для практического применения оптимальным является рассмотрение таких графов, которые с одной стороны отражали бы все зависимости лога, с другой стороны не порождали бы при этом новых зависимостей, которых в логе не было. Данные графы зависимостей называются *конформными логу* и именно такие графы зависимостей будут считаться кор-

ректными моделями для процессов по аналогии с работой [8].

Одна трасса t также задает отношение следования, данное отношение будет обозначаться $\lesssim_t$. Например, для трассы $t = AB$ будет справедливо $A \lesssim_t B$.

Можно ввести определение *подграфа G' , порождаемого трассой t* , в графе зависимостей $G = (V, E)$: $G' = (V', E')$, где $V' = \{A \in t\} \subseteq V$ и $E' = \{(B, A) \in E \mid B \lesssim_t A\}$, где $\lesssim_t$ — отношение частичного предпорядка, задаваемое трассой t .

Пусть заданы граф зависимостей процесса $PG = (V, E)$ и трасса t . Так же, как и в работе [1] будет предполагаться, что все возможные исполнения процесса P начинаются и заканчиваются одним (не обязательно одинаковым) действием. Будем говорить, что *трасса t и граф G согласованы*, если:

- подграф $G' = (V', E')$, порождаемый трассой t , связан, т. е. между каждой парой вершин $u, v \in V'$ существует путь в G' ;
- первое и последнее действия трассы t — действие начинающее и действие завершающее процесс P ;
- все вершины в V' достижимы из начального действия.

Граф зависимостей G называется *конформным логу L* , если выполнены следующие условия:

- для каждой зависимости в L существует путь в G ;
- между независимыми действиями в L не существует пути в G ;
- каждая трасса лога L согласована с G .

Среди конформных графов зависимостей для практического применения оптимальным выбором является конформный граф с минимальным числом ребер, такой конформный граф называется *минимальным*.

Было показано [1], что для любого лога существует конформный граф зависимостей, а для некоторых видов лога возможно построить и минимальный конформный логу граф. Данные алгоритмы будут рассмотрены в следующем разделе. Также авторами работы [1] было показано, что могут существовать несколько минимальных конформных логу графов и на практике выбор наилучшего из них, как модели процесса, остается открытой задачей.

Методы и цели исследования

В работе [1] были предложены три алгоритма и рассчитана временная сложность построения модели в виде ациклического ориентированного

графа для одного процесса. Пусть $m = |L|$ — число трасс в лог L ; $n = |V|$ — число возможных действий процесса P . Были рассмотрены следующие свойства лог L .

1. Лог содержит только такие трассы, в которых все возможные действия процесса встречаются ровно по одному разу. Было показано, что в таком случае каждый граф зависимостей G лог L будет являться конформным. Более того, предложенный в работе алгоритм строит минимальный конформный лог L граф G для данного случая. Сложность его построения составляет $O(mn^2)$.

2. В предположении, что конформный лог L граф G не содержит циклов, конформный граф может быть построен с временной сложностью $O(mn^3)$. В данном случае не гарантируется, что построенный конформный граф будет минимальным.

3. В общем случае поиск конформного графа G для лог L может быть осуществлен за $O(m(kn^3))$, где k — максимальное число повторов действия в лог L .

Основным предположением о входных параметрах данных алгоритмов является предположение, что число трасс m в лог L значительно превышает число действий n . Под элементарными операциями для вычисления оценок сложности данных алгоритмов понимаются простейшие операции редактирования над графами (например, добавление/удаление вершины, добавление/удаление ребра, слияние/расщепление вершин).

Цель данной работы — решение задачи поиска аномалий при наличии уже построенных моделей процессов, если известна дополнительная информация о свойствах этих процессов и/или о свойствах проверяемой на аномальность трассы.

Суть исследования и результаты

В рамках данного исследования рассматривается лог событий, в котором встречаются трассы, соответствующие нескольким процессам. Сначала рассматривается случай, когда известно, что одна трасса лог может содержать данные ровно одного процесса. Далее рассматривается случай, когда в рамках одной трассы лог могут встречаться данные нескольких процессов. Для всех рассмотренных видов логов приводятся временные оценки сложности решения задачи поиска аномалий, которые являются улучшением аналогичных оценок сложности, описанных в работе [8]. Последним рассматривается случай, когда и проверяемая на аномальность трасса содержит в себе действия нескольких процессов, приводится оценка слож-

ности ответа на вопрос об ее аномальности относительно уже построенных моделей процессов.

Поиск аномалий с использованием модели процесса, построенной по лог, который содержит данные нескольких процессов

Рассматриваются s процессов $P_1, \dots, P_s$ с непустыми множествами действий $V_1, \dots, V_s$ и графами $G_1 = (V_1, E_1), \dots, G_s = (V_s, E_s)$ соответственно.

Задача, решаемая в данном подразделе: пусть задан некоторый лог L , который может содержать в себе трассы (следы) нескольких процессов, необходимо построить конформные графы (модели) этих процессов и оценить трудоемкость данного построения.

Пусть $M = |L|$, $n = \max_i |V_i|$, $i = 1, \dots, s$; k — максимальное число повторов действия в лог L . Предполагается, что $M \gg n$, $M \rightarrow \infty$, размер лог может быть сколь угодно велик, а суммарное число возможных действий n для s процессов конечно.

Рассматривается случай, когда s процессов имеют различные множества действий. Простейшим примером такого лог событий при $s = 2$ является лог $L = \{AB, DE\}$.

Теорема 1 (процессы с различными множествами действий) [8]. Пусть задан лог L для s процессов. Пусть s процессов имеют попарно различные множества действий, $V_i \cap V_j = \emptyset$, при $i \neq j$, $i, j = 1, \dots, s$. Тогда сложность построения s конформных графов составляет $O(Mn^2)$, $O(Mn^3)$, $O(M(kn)^3)$ соответственно, в зависимости от свойств лог L .

Если построено s конформных графов зависимостей для s процессов и известно, что в поступающей трассе t может быть след только одного процесса, то задача наличия аномалий в t сводится к задаче определения согласованности этой трассы с s графами зависимостей.

Пусть $n = \max_i |V_i|$, $e = \max_i |E_i|$, $i = 1, \dots, s$.

Теорема 2 (поиск аномалий в процессах с различными множествами действий). Пусть s процессов имеют попарно различные множества действий, $V_i \cap V_j = \emptyset$, при $i \neq j$, $i, j = 1, \dots, s$. Пусть для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в некоторой трассе t может быть выполнен с временной сложностью $O(s + |t| + n^2 + e)$.

Доказательство. Пусть $G' = (V', E')$ — граф, порождаемый трассой t (алгоритм и сложность его построения будут описаны далее на шаге 4). Тогда проверка согласованности трассы t построенным конформным графам состоит из перечисленных далее шагов.

Шаг 1. Определение соответствующего трассе t конформного графа: так как действия между

процессами для теоремы 1 не пересекаются, то достаточно $O(s)$ операций, чтобы сравнить, например, начальные действия построенных графов и начальное действие проверяемой трассы. Пусть нужный конформный граф для проверки соответствия трассы t — граф $G = (V, E)$.

Шаг 2. Проверка $V' \subseteq V$. При хранении в памяти вершин V проверку можно осуществить за $O(|t|)$ операций.

Шаг 3. Проверка достижимости вершин из V' из начальной вершины графа G . На этом шаге уже известно, что число различных действий в трассе t не превышает n , поэтому проверку можно осуществить с помощью запуска алгоритма поиска в ширину [9], сложность $O(n + e)$.

Шаг 4. Построение графа G' для трассы t . Построение графа для одной трассы сводится к построению сначала отношения $\lesssim_t$, а потом построению отношения $\rightarrow_t$, которое соответствует множеству дуг E' . За один проход по трассе t для каждого из возможных n действий возможно определить первую и последнюю позиции, на которых встречалось это действие, временная сложность составит $|t|$, сложность по дополнительно используемой памяти составит $2n$.

Пусть хранение отношения $\lesssim_t$ организовано с помощью хеш-таблицы, множество пар вида «ключ: значение», где ключи — строковое значение, уже встреченные в трассе t пары действий, значения — целое число, встречалась ли пара в обратном порядке. Например, если задана трасса $t = AB$, то соответствующая ей хеш-таблица будет иметь вид: $h = \{AB : 0\}$; если же задана трасса $t = ABA$, то соответствующая ей хеш-таблица будет иметь вид: $h = \{AB : 1\}$, так как в трассе есть подтрасса BA .

Используя информацию о первом и последнем вхождении каждого из действий за $O(n^2)$ сравнений, возможно построить отношение $\lesssim_t$. При этом вставка нового ключа в хеш-таблицу происходит только в том случае, если текущая пара действий не является парой в обратном порядке для некоторого ключа, который в хеш-таблице уже присутствует (в таком случае необходимо увеличить на единицу значение по этому ключу, см. пример выше для трассы $t = ABA$).

Тогда построение отношения $\rightarrow_t$ возможно осуществить за $O(|t| + n^2)$ операций и $O(n^2)$ дополнительной памяти для хранения пар действий.

Шаг 5. Проверка $E' \subseteq E$. Так как на этом шаге уже известно, что $|V'| < n$, то проверка займет не более $O(n^2)$ операций.

Шаг 6. Проверка связности графа G' . На этом шаге уже известно, что количество ребер не превышает e , поэтому проверку связности возможно

осуществить с помощью одного запуска алгоритма поиска в ширину, сложность $O(n + e)$.

Итоговая сложность проверки согласованности трассы по перечисленным выше шагам составляет $O(s + |t| + (n + e) + |t| + n^2 + n^2 + (n + e))$, т. е. $O(s + |t| + n^2 + e)$.

Если для задачи поиска аномалий построены модели процессов в виде ациклических ориентированных графов, то параметры s, n, e будут константами и справедливо следующее следствие.

Следствие 1 (поиск аномалий в процессах с различными множествами действий). Пусть s процессов имеют попарно различные множества действий, $V_i \cap V_j = \emptyset$, при $i \neq j$, $i, j = 1, \dots, s$. Пусть для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в некоторой трассе t может быть выполнен с временной сложностью $O(|t|)$.

Пусть $m = \max |L_i|$, L_i — часть лога L , соответствующая i -му процессу, $i = 1, \dots, s$. Пусть $n = \max |V_i|$, $i = 1, \dots, s$. Пусть известно, что процесс однозначно можно идентифицировать его начальным действием. Простейшим примером лога событий для двух процессов может быть лог $L = \{AB, CD\}$. При условии s различных начальных действий справедлива следующая теорема.

Теорема 3 (процессы, отличимые по начальному действию) [8]. Пусть задан лог L для s процессов. Пусть s процессов имеют попарно различные начальные действия. Тогда сложность построения s конформных графов составляет $O(smn^2)$, $O(smn^3)$, $O(sm(kn)^3)$ соответственно, в зависимости от свойств лога L .

Пусть $n = \max |V_i|$, $e = \max |E_i|$, $i = 1, \dots, s$.

Теорема 4 (поиск аномалий в процессах, отличимых по начальному действию). Пусть s процессов имеют попарно различные начальные действия. Пусть для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в некоторой трассе t может быть выполнен с временной сложностью $O(s + |t| + n^2 + e)$.

Доказательство данной теоремы аналогично доказательству теоремы 2.

Если для задачи поиска аномалий построены модели процессов в виде ациклических ориентированных графов, то параметры s, n, e будут константами и справедливо следующее следствие.

Следствие 2 (поиск аномалий в процессах, отличимых по начальному действию). Пусть s процессов имеют попарно различные начальные действия. Пусть для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в некоторой трассе t может быть выполнен с временной сложностью $O(|t|)$.

Рассматривается случай, когда не накладывается условие на попарно различные начальные действия для каждого из s процессов, однако есть возможность разделить лог L на части по действиям, которые отличают каждый процесс от всех остальных процессов. Следует отметить, что таким образом обеспечивается необходимое и достаточное условие существования системы различных представителей для теоремы Ф. Холла [10]. Простейшим примером лога для этого случая для двух процессов будет лог $L = \{ABA, AAC\}$, где действие B может встречаться только в процессе 1, действие C может встречаться только в процессе 2.

Теорема 5 (отличимые процессы по некоторому действию) [8]. Пусть задан лог L для s процессов. Пусть каждый из s процессов имеет по одному известному действию A_i , $i = 1, \dots, s$ такому, что оно не содержится в остальных процессах и содержится в каждой трассе, соответствующей i -му процессу. Тогда сложность построения s конформных графов составляет $O(smn^2)$, $O(smn^3)$, $O(sm(kn)^3)$ соответственно, в зависимости от свойств лога L .

Пусть $n = \max_i |V_i|$, $e = \max_i |E_i|$, $i = 1, \dots, s$.

Теорема 6 (поиск аномалий в процессах, отличимых по некоторому действию). Пусть каждый из s процессов имеет по одному известному действию A_i , $i = 1, \dots, s$ такому, что оно не содержится в остальных процессах и содержится в каждой трассе, соответствующей i -му процессу. Пусть для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в некоторой трассе t может быть выполнен с временной сложностью $O(|t| + n^2 + e)$.

Доказательство. Доказательство данной теоремы аналогично доказательству теоремы 2, за исключением шага 1: так как нам известны уникальные действия для процессов A_i , $i = 1, \dots, s$, то достаточно одного прохода по трассе t , чтобы найти одно из действий A_i . Сложность выбора нужного i -го конформного графа составляет $O(|t|)$, что не превышает сложности шагов 2–6 теоремы 2. Таким образом, временная сложность поиска аномалий в некоторой трассе t составляет $O(|t| + n^2 + e)$.

Если для задачи поиска аномалий построены модели процессов в виде ациклических ориентированных графов, то параметры s , n , e будут константами и справедливо следующее.

Следствие 3 (поиск аномалий в процессах, отличимых по некоторому действию). Пусть каждый из s процессов имеет по одному известному действию A_i , $i = 1, \dots, s$ такому, что оно не содержится в остальных процессах и содержится в каждой трассе, соответствующей i -му процессу. Пусть

для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в некоторой трассе t может быть выполнен с временной сложностью $O(|t|)$.

Теорема 3 рассматривает случай уникальных для процессов подстрок длины 1, которые начинаются с первой буквы трасс лога L . Пусть $m = \max_i |L_i|$, $n = \max_i |V_i|$, $i = 1, \dots, s$. Справедливо обобщение этой теоремы.

Теорема 7 (отличимые процессы по конечной подстроке) [8]. Пусть задан лог L для s процессов. Пусть известно, что каждая из трасс лога, начиная с некоторого фиксированного номера i , содержит подстроки фиксированной длины r , каждая из которых может принадлежать только одному из s процессов. Тогда сложность построения s конформных графов составляет $O(smn^2)$, $O(smn^3)$, $O(sm(km)^3)$ соответственно, в зависимости от свойств лога L .

Пусть $n = \max_i |V_i|$, $e = \max_i |E_i|$, $i = 1, \dots, s$.

Теорема 8 (поиск аномалий в процессах, отличимых по конечной подстроке). Пусть известно, что трасса t , начиная с некоторого номера i , содержит подстроку длины r , которая может принадлежать только одному из s процессов. Пусть для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в трассе t может быть выполнен с временной сложностью $O(|t| + n^2 + e)$.

Доказательство. Доказательство данной теоремы аналогично доказательству теоремы 2, за исключением шага 1: для определения нужного конформного графа необходимо использовать ту же идеальную хеш-функцию h , которая использовалась при построении моделей s процессов. Сложность выполнения шага 1 составит $O(1)$, что не превышает сложности шагов 2–6 теоремы 2. Таким образом, временная сложность поиска аномалий в некоторой трассе t составляет $O(|t| + n^2 + e)$.

Если для задачи поиска аномалий построены модели процессов в виде ациклических ориентированных графов, то параметры s , n , e будут константами и справедливо следующее следствие.

Следствие 4 (поиск аномалий в процессах, отличимых по конечной подстроке). Пусть известно, что трасса t , начиная с некоторого номера i содержит подстроку длины r , которая может принадлежать только одному из s процессов. Пусть для s процессов построено s конформных графов зависимостей, тогда поиск аномалий в трассе t может быть выполнен с временной сложностью $O(|t|)$.

Поиск аномалий с использованием модели процесса, построенной по логу, который содержит данные нескольких процессов внутри каждой из трасс

В данном подразделе решается задача поиска аномалий с использованием модели процесса, которая была построена по логу L такому, что в логе L в одной трассе могут содержаться данные нескольких процессов.

Следует заметить, что каждый из алгоритмов работы [1] после того, как были добавлены все зависимости из лога, удаляет те дуги, которые идут в обе стороны для пар действий. Это означает, если согласно логу L существуют зависимости $A \lesssim B$ и зависимость $B \lesssim A$, то дуги между вершинами, соответствующими действиям A и B , будут удалены. Если при этом действие A принадлежит процессу 1, а действие B принадлежит процессу 2, то это означает, что найдено условие, позволяющее отличать различные одновременно функционирующие процессы. Однако это условие накладывает ограничение на «полноту» лога, чтобы каждое из действий одного процесса было независимо от каждого из действий другого процесса.

Пусть $M = |L|$, $n = \max_i |V_i|$, $i = 1, \dots, s$.

Теорема 9 (множество процессов в трассе без повторов действий) [8]. Пусть задан лог L , который не содержит повторов действий в рамках одной трассы. Пусть в каждой из трасс лога L могут присутствовать действия от 1 до s процессов. Пусть множества действий для процессов не пересекаются, т. е. $V_i \cap V_j = \emptyset$, при $i \neq j$, $i, j = 1, \dots, s$. Если для всех действий $A \in V_i$, $B \in V_j$, $i \neq j$, $i = 1, \dots, s$ в логе L выполнено $A \lesssim B$ и $B \lesssim A$, то сложность построения s конформных графов составляет $O(Mn^2)$.

Если построенные s конформных графов по теореме 9 обладают свойствами из теорем 1, 3, 5, 7 и некоторая трасса t содержит в себе след только одного процесса, то для ответа на вопрос, является ли трасса t аномальной, могут быть применены следствия 1, 2, 3, 4 соответственно.

Следующая теорема отвечает на вопрос о сложности поиска аномалий, когда и поступающая трасса t содержит в себе следы нескольких процессов.

Теорема 10 (поиск аномалий для трассы, содержащей следы множества процессов, имеющих различные множества действий). Пусть множества действий для s процессов не пересекаются, т. е. $V_i \cap V_j = \emptyset$, при $i \neq j$, $i, j = 1, \dots, s$. Пусть для s процессов построено s конформных графов зависимостей. Пусть в некоторой трассе t могут содержаться следы множества процессов. Тогда поиск аномалий в трассе t может быть выполнен с временной сложностью $O(s^2 + s|t| + sn^2 + se)$.

Доказательство. Для того чтобы было возможно применить теорему 2, необходимо, чтобы трасса t была бы разбита на подтрассы, каждая из которых содержала бы в себе след только одного процесса.

Так как уже построено s конформных графов процессов, значит, известны множества V_i , $i, j = 1, \dots, s$. Так как множества действий для s процессов не пересекаются, т. е. $V_i \cap V_j = \emptyset$, при $i \neq j$, $i, j = 1, \dots, s$, значит за один проход по трассе t возможно однозначно определить, какому из процессов принадлежит каждое из действий в этой трассе, сложность шага составляет t . Пусть трасса t содержит в себе действия из $s' \leq s$ процессов. То есть трасса t может быть разбита на s' подтрасс t_i , $i = 1, \dots, s'$. Пусть максимальная длина подтрассы $p = \max_i |t_i|$, $i = 1, \dots, s'$.

Теперь, когда построены трассы t_i , $i = 1, \dots, s'$, становится возможным применение теоремы 2. Трасса t считается аномальной, если хотя бы одна из трасс t_i является аномальной. Таким образом, сложность в худшем случае составляет $|t|$ для разбиения трассы t на подтрассы t_i , $i = 1, \dots, s'$; s' вызовов алгоритма из теоремы 2 и составляет $O(|t| + s'(s + p + n^2 + e))$. Так как $s' \leq s$, $p \leq |t|$, то временная сложность составит $O(s^2 + s|t| + sn^2 + se)$.

Если для задачи поиска аномалий построены модели процессов в виде ациклических ориентированных графов, то параметры s , n , e будут константами и справедливо следующее следствие.

Следствие 5 (поиск аномалий для трассы, содержащей следы множества процессов, имеющих различные множества действий). Пусть множества действий для s процессов не пересекаются, т. е. $V_i \cap V_j = \emptyset$, при $i \neq j$, $i, j = 1, \dots, s$. Пусть для s процессов построено s конформных графов зависимостей. Пусть в некоторой трассе t могут содержаться следы множества процессов. Тогда поиск аномалий в трассе t может быть выполнен с временной сложностью $O(|t|)$.

Заключение

Рассмотрены задача построения формальной модели множества процессов в виде ациклического ориентированного графа и задача поиска аномалий с помощью построенных эталонных моделей процесса, а также случаи наличия следов нескольких процессов как между трассами одного лога событий, так и внутри трасс одного лога событий.

Показано, что решение задачи поиска аномалий имеет линейную сложность относительно длины трассы, для которой требуется дать ответ о наличии в ней аномалий, и квадратичную сложность относительно максимально возможного количества

действий и зависимостей в одном моделируемом процессе. Показано, что с помощью системы различных представителей возможно эффективно выделять траектории множества различных процессов.

Список литературы

1. Agrawal R., Gunopulos D., Leymann F. Mining process models from workflow logs // *Advances in Database Technology—EDBT'98: 6th International Conference on Extending Database Technology Valencia, Spain, March 23–27, 1998 Proceedings* 6. Springer Berlin Heidelberg, 1998. P. 467–483. DOI: 10.1007/bfb0101003.
2. Kunze M., Weske M. Behavioural models: From modelling finite automata to analysing business processes // Springer International Publishing, 2016. P. 163–185. DOI: 10.1007/978-3-319-44960-9.
3. Sommers D., Menkovski V., Fahland D. Supervised learning of process discovery techniques using graph neural networks // *Information Systems*. 2023. P. 102209. DOI: 10.1016/j.is.2023.102209.
4. Liu C., Cheng L., Zeng Q., Wen L. Formal Modeling and Discovery of Hierarchical Business Processes: A Petri Net-Based

Approach // *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2022. DOI: 10.1109/TSMC.2022.3195869.

5. Hübscher G., Geist V., Auer D. et al. Graph-based managing and mining of processes and data in the domain of intellectual property // *Information Systems*. 2022. No. 106. P. 101844. DOI: 10.1016/j.is.2021.101844.

6. Huang Y., Fan J., Yan Z. et al. Research on early warning for gas risks at a working face based on association rule mining // *Energies*. 2021. Vol. 14, No. 21. P. 6889. DOI: 10.3390/en14216889.

7. Терёхина И. Ю., Грушо А. А., Тимонина Е. Е., Шоргин С. Я. Построение моделей процесса с помощью простых сетей Петри // *Системы и средства информатики*. 2020. Том 30, № 4. С. 61–75. DOI: 10.14357/08696527200406.

8. Teryokhina I. Anomaly detection in several running processes // *International Journal of Open Information Technologies*. 2021. Vol. 10, No. 1. P. 21–27.

9. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to algorithms. MIT press, 2022. 1312 p.

10. Грушо А. А., Тимонина Е. Е., Грушо Н. А., Терёхина И. Ю. Выявление аномалий с помощью метаданных // *Информатика и ее применения*. 2020. Том 14, № 3. С. 76–80. DOI: 10.14357/19922264200311.

Anomaly Detection Using the DAG Process Models

I. Yu. Teryokhina, Assistant, iteryokhina@cs.msu.ru,
Lomonosov Moscow State University, Moscow, 119991, Russian Federation

Corresponding author:

Irina Yu. Teryokhina, Assistant, Lomonosov Moscow State University, Moscow, 119991, Russian Federation
E-mail: iteryokhina@cs.msu.ru

Received on April 17, 2023

Accepted on April 26, 2023

The problem of anomaly detection when several information technologies (processes) are functioning simultaneously is considered. It is assumed that the normal or typical functioning of the process can be described by some formal model. In this paper a directed acyclic graph (DAG) is considered as such formal model. It is considered that all necessary information about the processes possible executions can be received from an event log, which is considered given.

In this paper the event log traces containing the data from one and several processes were examined. The complexity estimates for solving anomaly detection problem with the use of the constructed earlier formal models are determined.

It has been shown that the solution of the anomaly detection problem has linear complexity with respect to the length of the trace, for which it is required to give an answer whether such trace contains an anomaly. Also, it has been proven that the solution of the anomaly detection problem has quadratic complexity with respect to the maximum possible number of actions and dependencies in one simulated process. Moreover, it has been demonstrated how the system of different representatives may help to effectively identify the trajectories of many different processes.

Keywords: anomaly detection, directed acyclic graph, process mining, process model construction.

For citation:

Teryokhina I. Yu. Anomaly Detection Using the DAG Process models, *Programnaya Ingeneria*, 2023, vol. 14, no. 6, pp. 285–291. DOI: 10.17587/prin.14.285-291.

References

1. Agrawal R., Gunopulos D., Leymann F. Mining process models from workflow logs, *Advances in Database Technology—EDBT'98: 6th International Conference on Extending Database Technology Valencia, Spain, March 23–27, 1998 Proceedings* 6, Springer Berlin Heidelberg, 1998, pp. 467–483. DOI: 10.1007/bfb0101003.

2. Kunze M., Weske M. Behavioural models: From modelling finite automata to analysing business processes, *Springer International Publishing*, 2016, pp. 163–185. DOI: 10.1007/978-3-319-44960-9.

3. Sommers D., Menkovski V., Fahland D. Supervised learning of process discovery techniques using graph neural networks, *Information Systems*, 2023, p. 102209. DOI: 10.1016/j.is.2023.102209.

4. Liu C., Cheng L., Zeng Q., Wen L. Formal Modeling and Discovery of Hierarchical Business Processes: A Petri Net-Based Approach, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022. DOI: 10.1109/TSMC.2022.3195869.

5. Hübscher G., Geist V., Auer D. et al. Graph-based managing and mining of processes and data in the domain of intellectual property, *Information Systems*, 2022, no. 106, p. 101844. DOI: 10.1016/j.is.2021.101844.

6. Huang Y., Fan J., Yan Z. et al. Research on early warning for gas risks at a working face based on association rule mining, *Energies*, 2021, vol. 14, no. 21, p. 6889. DOI: 10.3390/en14216889.

7. Teryokhina I. Yu., Grusho A. A., Timonina E. E., Shorgin S. Ya. Building process models with simple Petri nets, *Systems and Means of Informatics*, 2020, vol. 30, no. 4, pp. 61–75. DOI: 10.14357/08696527200406 (in Russian).

8. Teryokhina I. Anomaly detection in several running processes, *International Journal of Open Information Technologies*, 2021, vol. 10, no. 1, pp. 21–27.

9. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to algorithms, MIT press, 2022, 1312 p.

10. Grusho A. A., Timonina E. E., Grusho N. A., Teryokhina I. Yu. Anomaly detection with metadata, *Informatics and Applications*, 2020, vol. 14, no. 3, pp. 76–80. DOI: 10.14357/19922264200311 (in Russian).

К. А. Кулаков, канд. физ.-мат. наук, доц., доц. кафедры, kulakov@cs.petrstu.ru,
Н. Д. Москин, канд. техн. наук, доц., доц. кафедры, moskin@petrsu.ru,
А. А. Рогов, д-р техн. наук, проф., зав. кафедрой, rogov@petrsu.ru,
Р. В. Воронов, д-р техн. наук, доц., проф. кафедры, rvoronov@petrsu.ru,
Петрозаводский государственный университет

Поступила в редакцию 27.03.2023

Принята к публикации 25.04.2023

Программный комплекс СМАЛТ как инструментальный для исследования теоретико-графовых моделей текстов

Программный комплекс СМАЛТ (статистические методы анализа литературных текстов) предназначен для проведения исследований в области атрибуции литературных текстов. В статье рассмотрены новые инструментальные средства хранения, визуализации, сравнения и поиска данных, реализованные в системе. Они предназначены для анализа текстов с помощью теоретико-графовых моделей. Приведены примеры филологических исследований, выполненных с применением этих средств.

Ключевые слова: программный комплекс СМАЛТ, атрибуция текстов, теоретико-графовая модель, визуализация, сравнение, хранение

Для цитирования:

Кулаков К. А., Москин Н. Д., Рогов А. А., Воронов Р. В. Программный комплекс СМАЛТ как инструментальный для исследования теоретико-графовых моделей текстов // Программная инженерия. 2023. Том 14, № 6. С. 292—300. DOI: 10.17587/prin.14.292-300.

Введение

В настоящее время многочисленные задачи, связанные с обработкой текстовой информации (например, машинный перевод, ведение чатов, реферирование, выявление плагиата, атрибуция, выделение неоднородных фрагментов в тексте, классификация текстов и т. д.), решают с помощью количественных методов. В отечественной лингвистике подобного рода исследования, проводимые на стыке филологии и математики, начались в начале XX века с трудов А. А. Маркова [1] и Н. А. Морозова [2]. В наши дни для подобных исследований активно используют компьютерные технологии [3—7].

Среди большого числа алгоритмов, позволяющих решать отмеченные выше задачи, наиболее популярными являются нейросетевые методы. Появилось новое направление анализа текстов — автоматическая обработка текстов (АОТ). Анализируя большое количество информации, эти методы позволяют выявить статистические зако-

номерности и на их основе решить поставленные задачи. Однако существует ряд задач (например, возникающих при установлении авторства, поиске плагиата и т. д.), где нет большого количества информации, а значит, нейросетевые методы не дают нужного результата. Кроме того, в русскоязычных текстах со свободным порядком слов при анализе и понимании текста иногда нужно учитывать возникающие подчинительные связи между теми словами, которые расположены друг от друга на расстоянии в несколько лексем. Современные нейросетевые методы, как правило, основываются на векторной модели смысла слова, т. е. на анализе близкостоящих слов, а значит, учитываются только синтагматические связи. Они не берут в расчет связи между предложениями. В связи с этим обстоятельством приобретают большое значение графовые модели текстов, когда указываются взаимосвязи между конструкциями текста (словами и предложениями). Использование таких моделей позволяет учитывать больше информации, заложенной в тексте. Это является важным

при решении ряда задач, например, атрибуции (в частности, установления авторства).

Рассмотрим, как реализован анализ графов в языках программирования R и Python. Во-первых, в обоих языках можно воспользоваться базовыми структурами данных (массивами, списками) для представления графов с помощью матриц смежности или инцидентности, списков смежности или ребер, а также более сложными конструкциями (таблицы в R или классы в Python) для организации связей и зависимостей между объектами графа. Во-вторых, эти языки содержат различные специализированные библиотеки со встроенными функциями для удобной работы с графами.

Одной из наиболее популярных библиотек языка программирования Python для работы с графами является NetworkX. Эта библиотека содержит структуры данных для хранения ориентированных и неориентированных графов и мультиграфов. В ней реализованы многочисленные алгоритмы над графами, она имеет встроенные средства визуализации графов. Библиотека достаточно проста в использовании. Для определения графа необходимо задать список вершин и список ребер, состоящий из кортежей — пар смежных вершин. Эти списки добавляются в создаваемый объект графа, после чего можно вызывать те или иные алгоритмы обработки графа или визуализировать его.

Более мощное средство для визуализации графов в Python — это библиотека PyVis. С ее помощью можно представлять достаточно большие графы, вершины которых помечены дополнительной информацией и раскрашены в различные цвета. Функции библиотеки позволяют интерактивно взаимодействовать с визуализацией, в том числе изменять масштаб, перемещать элементы графа на экране и пр. Сам граф задается достаточно просто — списками вершин, списками меток и цветов вершин, а также списком ребер (пар вершин).

Для работы с графами на языке R можно воспользоваться пакетом igraph. Граф задается с помощью массивов начальных вершин и конечных вершин ребер. Методы пакета позволяют решать широкий класс задач анализа графов, вычисления различных характеристик графа, модификации графа, выделения подграфов и пр. Кроме того, имеются возможности для визуализации графов, представления различной статистической информации.

Однако все приведенные инструментальные средства предназначены для обработки специалистом по анализу данных. При проведении эксперимента требуется выполнить следующие шаги:

- собрать данные;

- представить их в соответствующем формате;
- провести анализ с помощью соответствующего пакета;
- визуализировать результат и показать его специалисту;
- в случае изменения данных для эксперимента все требуется повторить заново.

В системе СМАЛТ (статистические методы анализа литературных текстов) база данных (БД) привязана к методам обработки и выполнения данных шагов значительно упрощается, проводить эксперименты может филолог. В настоящей статье приведены способы хранения и методы анализа графовых моделей текстов в информационной системе СМАЛТ.

Обновленная архитектура программного комплекса СМАЛТ

Программный комплекс СМАЛТ предназначен для сбора корпуса текстов в дореволюционной графике и помощи в проведении научных исследований с корпусом. Зародившись как небольшая программа для разметки текстов на ПК Macintosh [8] программный комплекс развился до полноценного веб-приложения, с помощью которого филологи выполняют загрузку и разметку текстов в дореволюционной графике, а также различные расчеты для проведения научных исследований [9–13].

Программный комплекс СМАЛТ представляет собой хранилище исходных текстов в текстовом формате, формате Microsoft Word или HTML, БД с размеченными текстами и результатами исследований и веб-приложение для обработки запросов к программному комплексу. Ключевыми моментами работы веб-приложения являются обработка разметки текстов, где каждому слову ставится в соответствие набор атрибутов (в частности, часть речи), а также построение и обработка текстов из исходных слов и разметки. Таким образом, важным в работе программного комплекса СМАЛТ являются скорость работы и универсальность структур данных.

Веб-приложение построено на базе архитектурного шаблона *Model-Controller* (MC). Для каждой задачи имеется входной скрипт, в котором определяются запрашиваемое действие и операция над данными. Например, для задачи исследования теоретико-графовых моделей текстов имеются шесть действий:

- работа со списком графов;
- загрузка нового графа;
- просмотр отдельного графа;
- обновление данных графа;
- сравнение графов;
- визуализация графа.

Операции над данными позволяют скоординировать действия. Например, для действия «сравнение графов» с помощью операции задается алгоритм сравнения.

Контроллер представляет собой реализацию бизнес-логики приложения в части решаемой задачи. В базовом контроллере прописаны методы генерации элементов интерфейса пользователя, например, шапка и подвал страницы, отображение текста с разметкой. Модель реализует интерфейс работы с данными из БД. В модели выполняются проверка доступности запрашиваемой операции, формирование запросов к БД, получение и обработка результатов.

Доступ к функциональным возможностям программного комплекса регулируется с помощью модели пользователей системы с обязательным управлением доступом (*Mandatory Access Control*, MAC). Существуют четыре уровня доступа: уровень обычного пользователя, уровень редактора, уровень модератора и уровень администратора. Отдельно указывается роль исследователя для выполнения различных запросов к системе.

Формат хранения теоретико-графовых моделей текстов

Рассмотрим используемые структуры данных для работы с теоретико-графовыми моделями текстов. В качестве примера подобной модели можно привести «граф сильных связей» [3]. Для его построения анализируется матрица частот парной встречаемости грамматических классов слов текста (биграмм). Для получения такой матрицы необходимо:

- выбрать систему грамматических классов (например, существительное, прилагательное, числительное, местоимение, глагол, причастие, деепричастие, наречие и т. п.); каждому грамматическому классу ставится в соответствие вершина графа;

- перекодировать последовательность слов анализируемого текста в последовательность соответствующих обозначений грамматических классов;

- вычислить частоты парной встречаемости a_{ij} для каждой пары классов с учетом направления развертывания текста (слева направо); если значение a_{ij} ниже некоторого заранее заданного порога α , то связь отбрасывается (она считается «слабой»); остальные «сильные связи» формируют теоретико-графовую модель.

На рис. 1 показана модель, построенная по тексту Ф. М. Достоевского «Молодое перо» (порог

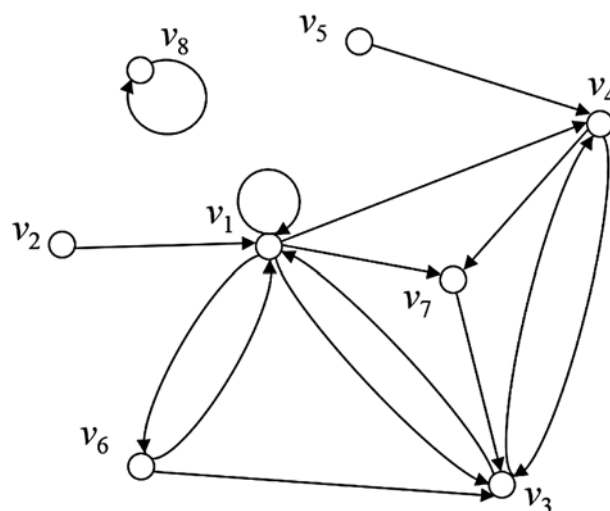


Рис. 1. Теоретико-графовая модель «сильных связей» грамматических классов

$\alpha = 0,02$). Здесь значимыми оказались восемь классов: существительное, прилагательное, местоимение, глагол, частица, предлог, союз и цитата (это соответственно вершины $v_1, v_2, \dots, v_8$). Подробнее со спецификой разбора текстов в дореволюционной орфографии и другими примерами моделей можно познакомиться в монографии [8].

При импорте данных теоретико-графовой модели из работы [14] на вход подается файл в специальном формате, содержащий:

- описание графа;
- перечень вершин с указанием частей речи вершины;
- перечень ребер графа.

Результат разбора файла размещается в следующих таблицах (рис. 2):

- `graph_description` — описание теоретико-графовой модели;
- `graph_vertices` — перечень узлов теоретико-графовой модели;
- `graph_edges` — перечень ребер теоретико-графовой модели;
- `graph_texts` — связь теоретико-графовой модели с текстом;
- `graph_favorite` — отметки добавления теоретико-графовой модели в список избранного для пользователей.

Разбор графа связывается с помощью внешних ключей с таблицей пользователей (`sys_users`) и таблицей текстов (`text`).

Каждая теоретико-графовая модель принадлежит пользователю-создателю и доступна только ему по умолчанию. Если пользователь обладает правами исследователя, то он может добавить модель в перечень общедоступных. Удаление графа реализовано с помощью отметки об удалении.

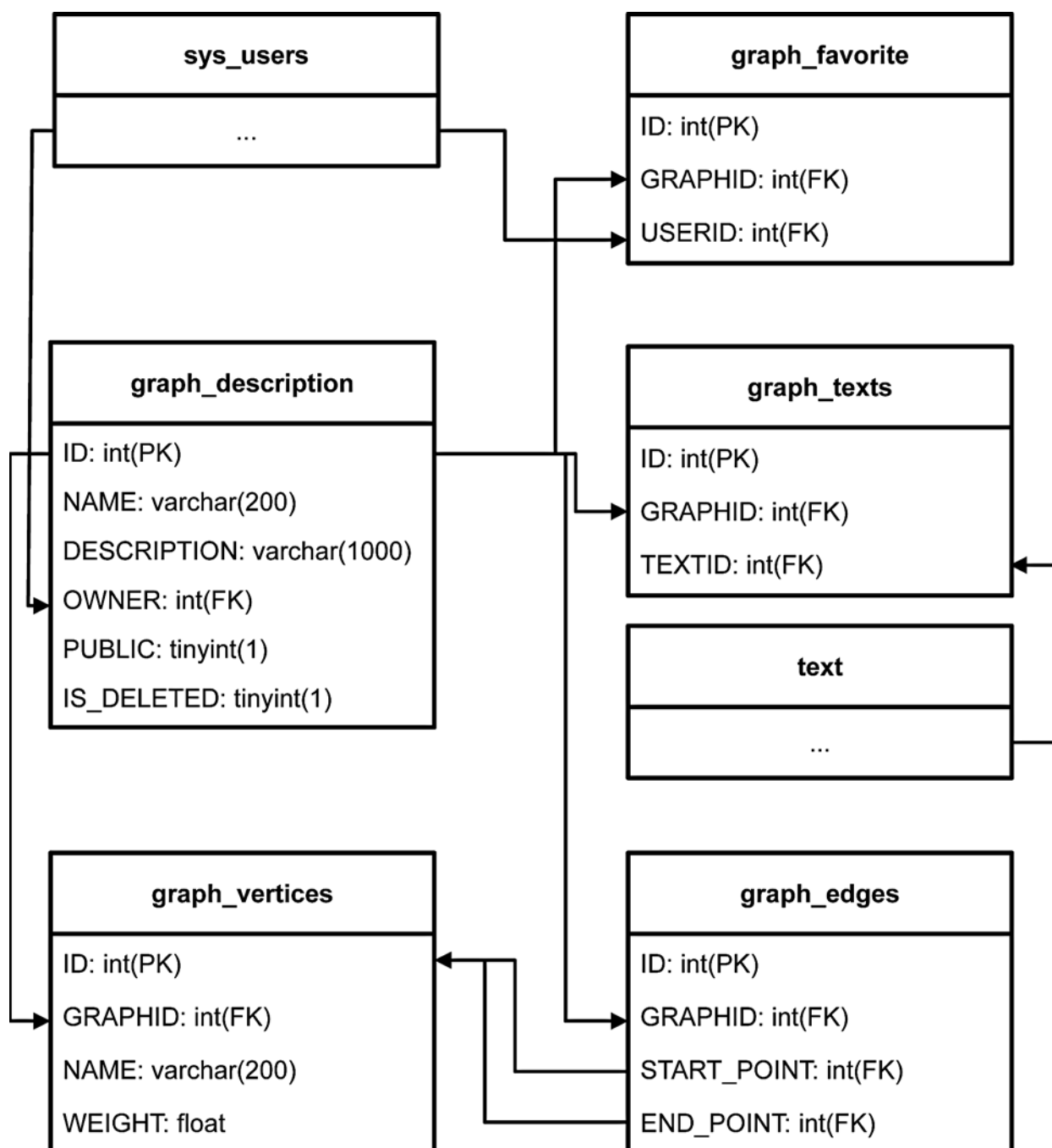


Рис. 2. Структура базы данных для хранения теоретико-графовых моделей

В дальнейшем администратор системы может выполнить очистку БД и окончательно удалить отмеченные графы.

Сравнительный анализ графов

Для проведения сравнительного анализа текстов на основе их теоретико-графовых моделей в программном комплексе СМАЛТ используются подграфовые метрики [15]. Одним из способов

оценки возможных ошибок сравнения является мера на основе операций редактирования (операций вставки, удаления и замены вершин и ребер в графе). Исторически это расстояние возникло на основе расстояний между строками. При этом на практике (особенно при анализе текстов) часто одни операции являются более значимыми по сравнению с другими, поэтому каждой операции ставится в соответствие ее стоимость (вес). Алгоритм ищет минимальную суммарную стоимость

операций редактирования, с помощью которых можно преобразовать граф G_1 в граф G_2 .

Вычисление расстояния на основе операций редактирования часто осуществляется с помощью алгоритма поиска на дереве. Если выполняются несколько специальных условий при определении стоимостей операций редактирования, то достаточно рассматривать только ограниченное число последовательностей операций редактирования для нахождения одной с минимальной стоимостью.

Один из наиболее распространенных методов поиска расстояния базируется на A^* -алгоритме [15]. Идея алгоритма заключается в том, чтобы использовать дерево поиска для представления рассматриваемой оптимизационной задачи так, чтобы корневая вершина дерева соответствовала точке начала поиска, внутренние вершины соответствовали частным решениям, а вершины-листья — полным решениям. Дерево поиска строится динамически во время выполнения алгоритма путем многократного создания вершин-потомков, связанных ребрами с текущей рассматриваемой вершиной.

Более эффективные алгоритмы можно построить, если учитывать особенности теоретико-графовых моделей. Отметим, что достаточно часто при моделировании языковой структуры текстов возникают деревья, т. е. связные (ориентированные или неориентированные) графы, не содержащие циклов. В качестве примера можно привести деревья зависимостей или деревья составляющих, которые отражают синтаксическую структуру текстов. Интересные структуры получаются как результат применения некоторых методов интеллектуального анализа данных. Речь идет о деревьях решений, которые строятся для решения задачи атрибуции текстов (например, при исследовании дореволюционных статей из журналов «Время», «Эпоха» и еженедельника «Гражданин» [8]).

С учетом изложенного выше для сравнения деревьев T_1 и T_2 на основе операций редактирования в программном комплексе СМАЛТ был реализован алгоритм, рассмотренный в работе [15]. Сложность данного алгоритма равна $O(|T_1| \times |T_2| \times \min(\text{depth}(T_1), \text{leaves}(T_1)) \times \min(\text{depth}(T_2), \text{leaves}(T_2)))$, где $|T_i|$ — число вершин в дереве T_i ; $\text{depth}(T_i)$ — максимальное значение $\text{depth}(v_j)$, а $\text{depth}(v_j)$ — это число ребер на пути от корня T_i до его вершины v_j ; $\text{leaves}(T_i)$ — число листьев дерева T_i . С помощью него было проведено исследование на предмет повышения устойчивости результатов классификации, полученных с использованием деревьев решений [16].

Кроме того, в программном комплексе СМАЛТ были реализованы расстояния для сравнения графов с упорядоченными вершинами. Приведем меры сравнения «графов сильных связей» (подробнее об их построении и интерпретации в работах [3, 8]):

- метрика городских кварталов ρ_1 ;
- евклидова метрика ρ_2 ;
- мера Хэмминга ρ_3 ;
- коэффициент Роджерса—Танимото ρ_4 .

Например, коэффициент Роджерса—Танимото можно найти по формуле

$$\rho_4 = \frac{m_{1,2}}{m_1 + m_2 - m_{1,2}},$$

где m_1 и m_2 — число вершин в графах 1 и 2, $m_{1,2}$ — число общих вершин.

Значение коэффициента ρ_4 заключено в отрезке от 0 до 1. Чем больше доля общих вершин, тем ближе значение ρ_4 к единице. Данная мера также используется при атрибуции анонимных текстов.

Визуализация теоретико-графовых моделей и результатов анализа для филологов

Частный случай теоретико-графовой модели в виде дерева решений представлен на рис. 3. Пользователь может увидеть описание и статистики моделей, перечень вершин и ребер, а также визуализацию теоретико-графовой модели. Для сравнения теоретико-графовых моделей реализована специальная форма (рис. 4). Пользователь может выбрать две теоретико-графовых модели, их глубину, стоимости операций добавления вершины, удаления вершины, изменения близкой вершины по весу, уровня близости вершин по весу, изменения дальней вершины по весу, изменения вершины с частичным совпадением частей речи, изменения вершины с несовпадением частей речи.

После выполнения сравнения пользователю доступен как итоговый результат, так и полученная таблица весов (рис. 5). Также отображается затраченное на сравнение время сервера.

Визуализация теоретико-графовой модели выполняется с помощью библиотеки GraphViz¹. Для подключения библиотеки в программном комплексе реализован класс GraphVizModel, который преобразует вершины и ребра теоретико-графовой модели в вершины и ребра графа с указанием названий вершин, их веса и цветовым обозначением веса.

¹ <https://graphviz.org/>

Граф дерева решений "Достоевский - остальные 19.01"

- Владелец: Kulakov Kirill
- Доступность: Публичный список

Назад к списку

Название: Достоевский - остальные 19.01
Описание: Дерево решений-0 экспериментальный граф Дерево (биграмма, шаг 500 слов, объем 1000 слов)
Метрики:

- Число вершин: 55;
- Число ребер: 54;
- Глубина графа: 5.

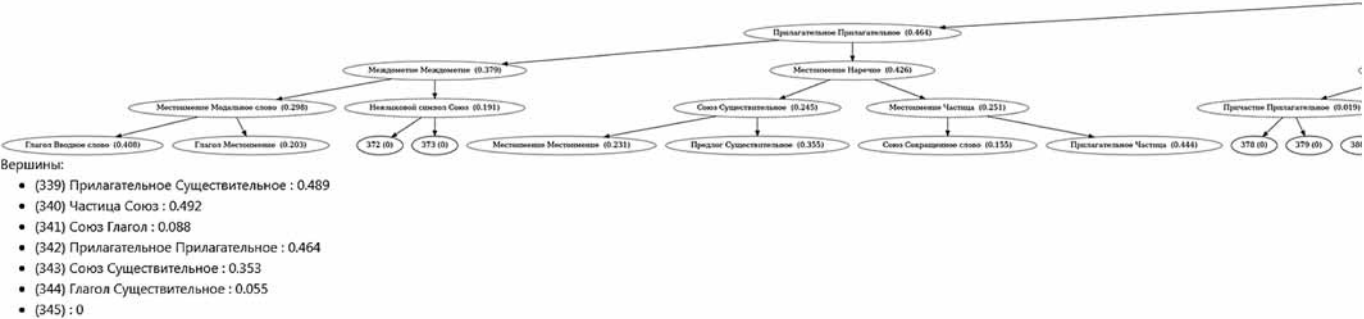


Рис. 3. Пример описания теоретико-графовой модели

SMALT ShowerСтатьиДерево атрибутовИсследованияСписки текстовСправка

Вход / Регистрация

Сравнение графов деревьев решений

Назад к списку

Сравнение на базе весов вершин

Граф 1: Достоевский - остальныеГлубина графа 1 (0 - весь граф): 0Граф 2: Достоевский - остальныеГлубина графа 2 (0 - весь граф): 0

Стоимость добавления вершины	Стоимость удаления вершины	Стоимость изменения близкой вершины	Уровень близости вершин	Стоимость изменения дальней вершины
1	1	1	0.25	2

СравнитьСбросить

Сравнение на базе имен и весов вершин

Граф 1: Достоевский - остальныеГлубина графа 1 (0 - весь граф): 0Граф 2: Достоевский - остальныеГлубина графа 2 (0 - весь граф): 0

Стоимость добавления вершины	Стоимость удаления вершины	Стоимость изменения близкой вершины по весу	Уровень близости вершин по весу	Стоимость изменения дальней вершины по весу	Стоимость изменения вершины с частичным совпадением частей речи	Стоимость изменения вершины с не совпадением частей речи
1	1	1	0.25	2	1	2

СравнитьСбросить

Сервис разработан при финансовой поддержке РФФИ (Отделение гуманитарных и общественных наук), проект «Проблема атрибуции анонимных и псевдонимных статей в журналах "Время", "Эпоха" и еженедельнике "Гражданин"» (No 18-012-90026).

Замечания по работе сервиса:Страница создана за 0.00351 сек.

Рис. 4. Форма сравнения теоретико-графовых моделей

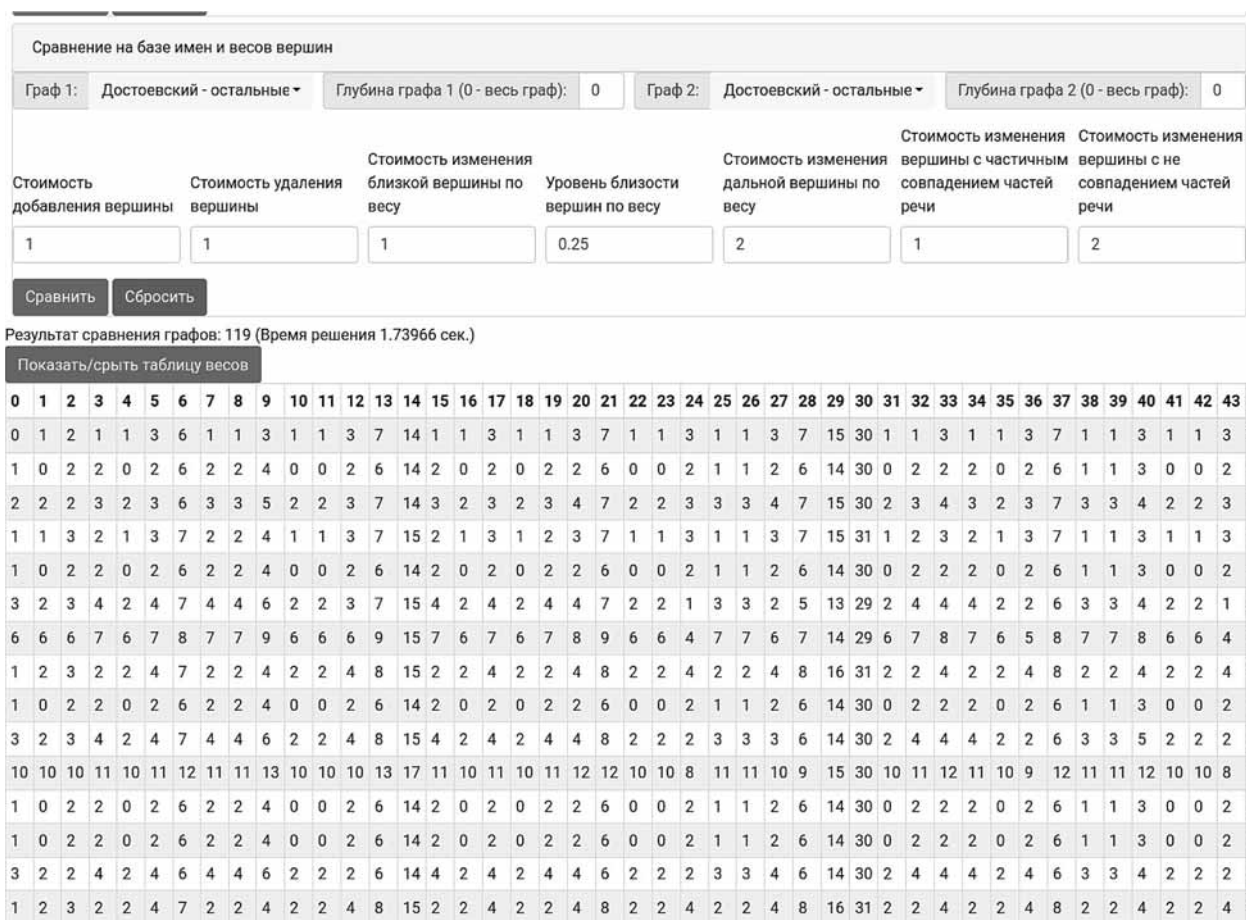


Рис. 5. Результат сравнения теоретико-графовых моделей с фрагментом таблицы весов

Результат визуализации продемонстрирован на рис. 6. На данном рисунке отображен фрагмент теоретико-графовой модели, которая была построена по дореволюционным текстам из журналов «Время», «Эпоха» и еженедельника «Гражданин» на основе дерева решений. При участии специалистов Института филологии Петрозаводского государственного университета была сформирована выборка текстов Ф. М. Достоевского и других авторов. При этом для увеличения объема выборки тексты были поделены на фрагменты объемом 1000 слов с шагом скользящего окна в 500 слов (подробнее в работе [8]).

Далее для каждого фрагмента была вычислена частота встречаемости биграмм, т. е. заданных последовательностей двух частей речи (например, частица + союз). Помимо биграммы на рис. 6 в скобках указан индекс Джини, методика подсчета которого изложена в работе [16]. В случае, когда индекс Джини равен нулю, условие для дальнейшей классификации отсутствует (для идентификации этих случаев была введена сквозная нумерация вершин по всему множеству исследуемых деревьев). Отметим, что подобных теоретико-графовых моделей недостаточно, чтобы сделать окончательные выводы об атрибуции текста. Эти выводы могут

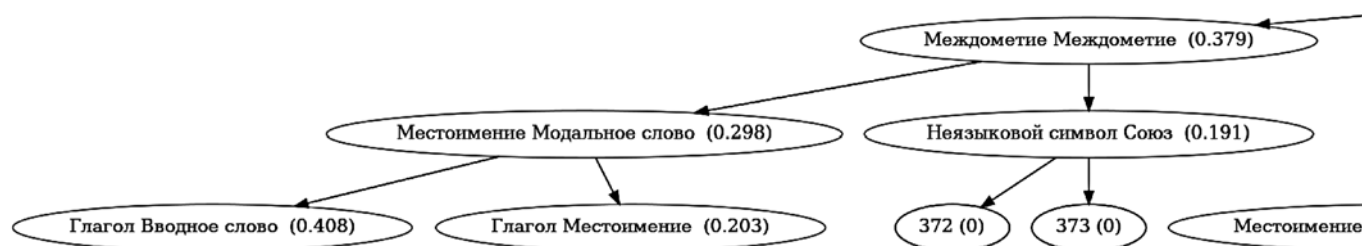


Рис. 6. Визуализация фрагмента теоретико-графовой модели

сделать лишь эксперты в области литературоведения, а модели нужны чтобы подтвердить или опровергнуть ту или иную филологическую гипотезу.

Приведем проведенные эксперименты над теоретико-графовыми моделями с помощью системы СМАЛТ:

- определение параметров для построения деревьев и лесов решений, пригодных для решения задачи атрибуции [17];
- проверка методики «сильный граф» для атрибуции текстов Ф. М. Достоевского и других авторов [8];
- апробация методики проверки устойчивости результатов методов машинного обучения с помощью подграфовой метрики на основе операций редактирования [16];
- описание и анализ текстов с помощью лексических спектров, которые позволили использовать два новых параметра как признаки при решении задачи классификации на деревьях решений [18];
- применение графовых моделей в виде деревьев решений, которые позволили выявить характерные особенности стилей писателей [8];
- решение задач атрибуции с использованием графовых моделей введены в научный оборот и активно используются специалистами-филологами в их публикациях [9–13].

Заключение

Теоретико-графовые модели находят широкое применение при анализе текстовых данных. Существующие системы анализа графовых моделей (в языках программирования R и Python) ориентированы на специалистов по анализу данных и используются в отрыве от исследуемых текстов.

Программный комплекс СМАЛТ позволяет загружать, просматривать и анализировать теоретико-графовые модели с построением связей с исследуемыми текстами и может использоваться специалистами в области филологии.

Список литературы

1. **Марков А. А.** Примѣръ статистическаго изслѣдованія надъ текстомъ «Евгенія Онегина», иллюстрирующій связь испытаній въ цѣпь // *Извѣстія Императорской Академіи Наукъ*. 1913. Том 7, № 3. С. 153–162.
2. **Морозов Н. А.** Лингвистические спектры: средство для отличия плагиатов от истинных произведений того или иного известного автора. Стилеметрический этюд // *Известия отд. русского языка и словесности Имп. Акад. наук*. Петрозаводск: тип. Имп. Акад. наук, 1915. Т. XX, кн. 4. С. 93–134.
3. **Милов Л. В., Бородин Л. И., Иванова Т. В.** и др. *От Нестора до Фонвизина: новые методы определения авторства* / Под ред. Л. В. Милова. М.: Прогресс, 1994. 445 с.

4. **Шевелев О. Г., Петраков А. В.** Классификация текстов с помощью деревьев решений и нейронных сетей прямого распространения // *Вестник Томского государственного университета*. 2006. № 290. С. 300–307.

5. **Великанова Н. П., Орехов Б. В.** Цифровая текстология: атрибуция текста на примере романа М. А. Шолохова «Тихий Дон» // *Мир Шолохова*. 2019. Том 1, № 11. С. 70–82.

6. **Масаева О. С.** Метод Дельта Берроуза // *Научно-исследовательская работа обучающихся и молодых ученых: материалы 74-й Всероссийской (с международным участием) научной конференции обучающихся и молодых ученых*. Петрозаводск, 2022. С. 262–266.

7. **Марусенко М. А.** Атрибуция анонимных и псевдонимных текстов как типичная задача распознавания образов // *Историография и источниковедение отечественной истории*. Санкт-Петербург, 2003. Вып. 3. С. 116–135.

8. **Рогов А. А., Абрамов Р. В., Бучнева Д. Д.** и др. Проблема атрибуции в журналах «Время», «Эпоха» и еженедельнике «Гражданин». Петрозаводск: Острова, 2021. 400 с.

9. **Бучнева Д. Д.** Проблемы атрибуции в «Гражданине» Достоевского: дебаты и аргументы // *Неизвестный Достоевский*. 2022. Том 9, № 3. С. 25–53.

10. **Бучнева Д. Д.** Помогают ли современные статистические методы при атрибуции текстов? // *Матер. IV Междунар. науч.-практ. конф. «Формирование профессиональной компетентности филолога в поликультурной образовательной среде»*. Симферополь, 2021. С. 47–53.

11. **Захаров В. Н.** Задачи атрибуции анонимных статей в изданиях Достоевского // *Неизвестный Достоевский*. 2022. Том 9, № 3. С. 5–24.

12. **Захарова О. В.** Атрибуция в зеркале статистики: анонимные статьи в журналах братьев Достоевских «Время» и «Эпоха» // *Неизвестный Достоевский*. 2021. Том 8, № 2. С. 81–106. DOI: 10.15393/j10.art.2021.5481.

13. **Лебедев А. А.** Анализ последовательностей частей речи и категория идиостиля // *Ученые записки Петрозаводского государственного университета*. 2021. Том 43, № 5. С. 29–31. DOI: 10.15393/uchz.art.2021.633.

14. **Рогов А. А., Кулаков К. А., Москин Н. Д.** Программная поддержка в решении задачи атрибуции текстов // *Программная инженерия*. 2019. Том 10, № 5. С. 234–240. DOI: 10.17587/prin.10.234-240.

15. **Москин Н. Д.** Алгоритмы сравнения графов и теоретико-графовых моделей. Петрозаводск: Изд-во ПетрГУ, 2009. 84 с.

16. **Moskin N. D., Kulakov K. A., Rogov A. A., Abramov R. V.** Research the Stability of Decision Trees Using Distances on Graphs // *Труды института системного анализа РАН*. 2023. Том 73, № 1. С. 94–100. DOI: 10.14357/20790279230111.

17. **Abramov R. V., Kulakov K. A., Lebedev A. A., Moskin N. D., Rogov A. A.** Research of features of Dostoevsky's publicistic style by using n-grams based on the materials of the «Time» and «Epoch» magazines // *Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления*. Санкт-Петербург, 2021. Том 17, Вып. 4. С. 389–396. DOI: 10.21638/11701/spbu10.2021.407.

18. **Moskin N. D., Kulakov K. A., Rogov A. A., Abramov R. V.** Characteristics of Lexical Spectra of Texts in the Problem of Establishing Authorship // *Stability and Control Processes (SCP 2020): Proceedings of the 4th International Conference Dedicated to the Memory of Professor Vladimir Zubov, series Lecture Notes in Control and Information Sciences*. Springer, 2022. P. 761–768. DOI: 10.1007/978-3-030-87966-2_87.

SMALT Software Package as a Tool for the Study of Graph-Theoretic Models of Texts

K. A. Kulakov, Associate Professor, kulakov@cs.petrstu.ru,

N. D. Moskin, Associate Professor, moskin@petrstu.ru,

A. A. Rogov, Head of Department, rogov@petrstu.ru,

R. V. Voronov, Professor, rvoronov@petrstu.ru

Petrozavodsk State University, Petrozavodsk, 185910, Russian Federation

Corresponding author:

Nikolai D. Moskin, Associate Professor,

Petrozavodsk State University, Petrozavodsk, 185910, Russian Federation

E-mail: moskin@petrstu.ru

Received on March 27, 2023

Accepted on April 25, 2023

The SMALT software package (Statistical methods of analysis of literary texts) is implemented to conduct research in the field of attribution of literary texts. The article discusses new tools for storing, visualizing, comparing and searching data implemented in the system. They are implemented for text analysis using graph-theoretic models. Examples of philological studies performed using these tools are given.

Keywords: SMALT software package, text attribution, graph-theoretic model, visualization, comparison, storage

For citation:

Kulakov K. A., Moskin N. D., Rogov A. A., Voronov R. V. SMALT Software Package as a Tool for the Study of Graph-Theoretic Models of Texts, *Programmnaya ingeneria*, 2023, vol. 14, no. 6, pp. 292–300. DOI: 10.17587/prin.14.292-300.

References

1. **Markov A. A.** An example of a statistical study on the text “Eugene Onegin”, illustrating the connection of trials in a chain, *Izvestiya Imperatorskoj Akademii Nauk*, 1913, vol. 7, no. 3, pp. 153–162 (in Russian).
2. **Morozov N. A.** Linguistic spectra: a means to distinguish plagiarisms from the true works of a given famous author. Stylometric study, *Izvestiya otd. russkogo yazyka i slovestnosti Imp. Akad. nauk. Petrograd: tip. Imp. Akad. nauk*, 1915, vol. XX, no. 4, pp. 93–134 (in Russian).
3. **Milov L. V., Borodkin L. I., Ivanova T. V.** et al. *Ot Nestora do Fonvizina: novye metody opredeleniya avtorstva* / Ed. L. V. Milov. Moscow, Progress, 1994, 445 p. (in Russian).
4. **Shevelev O. G., Petrakov A. V.** Text classification with decision trees and feed-forward neural networks, *Vestnik Tomskogo gosudarstvennogo universiteta*, 2006, no. 290, pp. 300–307 (in Russian).
5. **Velikanova N. P., Orekhov B. V.** Digital textology: text attribution on the example of the novel by M. A. Sholokhov «Quiet Flows the Don», *Mir Sholohova*, 2019, vol. 1, no. 11, pp. 70–82 (in Russian).
6. **Masaeva O. S.** Burrow’s Delta method, *Research work of students and young scientists: materials of the 74th All-Russian (with international participation) scientific conference of students and young scientists*, Petrozavodsk, 2022, pp. 262–266 (in Russian).
7. **Marusenko M. A.** Attribution of anonymous and pseudonymous texts as a typical task of pattern recognition, *Istoriografiya i istochnikovedenie otechestvennoj istorii*, 2003, no. 3, pp. 116–135 (in Russian).
8. **Rogov A. A., Abramov R. V., Buchneva D. D.** et al. *The problem of attribution in the magazines “Time”, “Epoch” and the weekly “Citizen”*, Petrozavodsk, Islands Publ., 2021, 400 p. (in Russian).
9. **Buchneva D. D.** Problems of attribution in Dostoyevsky’s “Citizen”: debate and arguments, *Neizvestnyj Dostoevskij*, 2022, vol. 9, no. 3, pp. 25–53 (in Russian).
10. **Buchneva D. D.** Do modern statistical methods help in the attribution of texts? *Materials of the IV International Scientific and Practical Conference “Formation of the professional competence of a philologist in a multicultural educational environment”*, Simferopol, 2021, pp. 47–53 (in Russian).
11. **Zaharov V. N.** Problems of attribution of anonymous articles in Dostoevsky’s publications, *Neizvestnyj Dostoevskij*, 2022, vol. 9, no. 3, pp. 5–24 (in Russian).
12. **Zaharova O. V.** Attribution in the mirror of statistics: anonymous articles in the magazines of the Dostoevsky brothers “Time” and “Epoch”, *Neizvestnyj Dostoevskij*, 2021, vol. 8, no. 2, pp. 81–106 (in Russian).
13. **Lebedev A. A.** Analysis of sequences of parts of speech and the category of idiostyle, *Uchenye zapiski Petrozavodskogo gosudarstvennogo universiteta*, 2021, vol. 43, no. 5, pp. 29–31. DOI: 10.15393/uchz.art.2021.633 (in Russian).
14. **Rogov A. A., Kulakov K. A., Moskin N. D.** Software support in solving the problem of text attribution, *Programmnaya ingeneria*, 2019, vol. 10, no. 5, pp. 234–240. DOI: 10.17587/prin.10.234-240 (in Russian).
15. **Moskin N. D.** *Algorithms for comparing graphs and graph-theoretic models*, Petrozavodsk, PetrSU Publ., 2009, 84 p. (in Russian).
16. **Moskin N. D., Kulakov K. A., Rogov A. A., Abramov R. V.** Research the Stability of Decision Trees Using Distances on Graphs, *Trudy instituta sistemnogo analiza RAN*, 2023, vol. 73, no. 1, pp. 94–100. DOI: 10.14357/20790279230111.
17. **Abramov R. V., Kulakov K. A., Lebedev A. A., Moskin N. D., Rogov A. A.** Research of features of Dostoevsky’s publicistic style by using n-grams based on the materials of the “Time” and “Epoch” magazines, *Vestnik Sankt-Peterburgskogo universiteta. Prikladnaya matematika. Informatika. Processy upravleniya*, 2021, vol. 17, no. 4, pp. 389–396. DOI: 10.21638/11701/spbu10.2021.407.
18. **Moskin N. D., Kulakov K. A., Rogov A. A., Abramov R. V.** Characteristics of Lexical Spectra of Texts in the Problem of Establishing Authorship, *Stability and Control Processes (SCP 2020): Proceedings of the 4th International Conference Dedicated to the Memory of Professor Vladimir Zubov, series Lecture Notes in Control and Information Sciences*, Springer, 2022, pp. 761–768. DOI: 10.1007/978-3-030-87966-2_87.

Method for Improving the Accuracy of Predictive Values of Time Series Based on the Imputation of Historical Data

E. V. Palchevsky, Lecturer, teelxp@inbox.ru,

Faculty of Information Technology and Big Data Analysis, Financial University under the Government of the Russian Federation, Moscow, 109456, Russian Federation

Corresponding author:

Evgeniy V. Palchevsky, Lecturer, Financial University under the Government of the Russian Federation, Moscow, 109456, Russian Federation

E-mail: teelxp@inbox.ru

Received on April 09, 2023

Accepted on April 27, 2023

The developed method for imputing retrospective data to improve the accuracy of long-term neural network forecasting is considered. The peculiarity of the method is to increase the accuracy of the predictive values of the time series obtained at the output of the recurrent neural network by increasing the amount of historical data provided by the production department of the Kumertau Electric Networks.

Using the example of forecasting the values of electricity consumption, an experiment was carried out on data imputation, which proves the effectiveness of using this method to improve accuracy in forecasting time series.

Keywords: data imputation, electricity consumption forecasting, neural networks, intelligent forecasting system.

For citation:

Palchevsky E. V. Method for Improving the Accuracy of Predictive Values of Time Series Based on the Imputation of Historical Data, *Programmnaya Ingeneria*, 2023, vol. 14, no. 6, pp. 301–306. DOI: 10.17587/prin.14.301-306

УДК 004.032.26

Е. В. Пальчевский, преподаватель, teelxp@inbox.ru, Факультет информационных технологий и анализа больших данных, Финансовый университет при Правительстве Российской Федерации, Москва, 109456, Российская Федерация

Метод повышения точности прогнозных значений временных рядов на основе импутации ретроспективных данных

Поступила в редакцию 09.04.2023

Принята к публикации 27.04.2023

Рассмотрен разработанный метод импутирования ретроспективных данных для повышения точности долгосрочного нейросетевого прогнозирования. Особенность метода заключается в повышении точности прогнозных значений временного ряда, получаемых на выходе рекуррентной нейронной сети, за счет увеличения количества исторических данных, предоставленных производственным отделением «Кумертауские электрические сети».

На примере прогнозирования значений потребления электроэнергии проведен эксперимент по импутации данных, доказывающий эффективность применения данного метода для повышения точности при прогнозировании временных рядов.

Ключевые слова: импутация данных, прогнозирование потребления электроэнергии, нейронные сети, интеллектуальная система прогнозирования

Для цитирования:

Palchevsky E. V. Method for Improving the Accuracy of Predictive Values of Time Series Based on the Imputation of Historical Data // Программная инженерия. 2023. Том 14, № 6. С. 301—306. DOI: 10.17587/prin.14.301-306.

Introduction

Due to the peculiarities of power system management in the Russian Federation, forecasting of electricity consumption is attracting more and more attention. An accurate model for forecasting electricity consumption allows management companies to constantly adjust not only the behavior of consumers when the energy sector of the market changes, but also take into account the need to reserve additional electric power capacities [1]. At the same time, for the sustainability of the energy industrial business, an accurate model for predicting electricity consumption is an important component, as it can be used to adjust the models of production and consumption of electricity. Moreover, an accurate electricity consumption forecasting model can be crucial for making decisions on energy management policy, for example, obligations on load capacities and operational safety of power plants, power systems [2]. This not only closes the gap between electricity supply and demand in the electricity market, but also significantly improves the operational efficiency of electricity system management.

But when forming predictive models, especially for long-term forecasting, the problem of their accuracy arises. There are two main reasons: the poor adaptability of the mathematical apparatus and, in the case of forecasting based on artificial neural networks, the lack of the necessary historical data. Research in the field of data imputation was carried out by both domestic (A. V. Losev [3], D. I. Kurapeev [4], etc.) and foreign scientists (S. Boseong [5], C. Young [6], W. Daoming [7], etc.). For example, in [8], a new data recovery method “EvoImputer» was proposed and compared with the classical approaches: mean, median, multiple imputation, expectation maximization, and K-nearest neighbors. In [9], effective combinations of data imputation approach, machine learning and data interpretation are introduced. At the same time, in all the above works and other analyzed sources, recovery by objects is not taken into account, i. e., data imputation can occur not only due to the “direct” restoration of all data, but also due to the restoration of data for individual objects (in our case, the objects are sensors for monitoring the state of power facilities).

Thus, the purpose of this work is to improve the accuracy of the predicted values of electricity consumption based on the imputation of retrospective

data obtained from sensors for monitoring the state of power facilities when predicting by a second-generation recurrent neural network.

Description of the data set and the method of their imputation

The data set (table 1) used in this study includes 25903 values from 01.01.2018 to 30.06.2022: the amount of electricity consumed and the air temperature from one sensor (meter).

Data about sensors (coordinate and attribute data), as well as all components of their data, are presented in the form of a logically unified database with the subsequent establishment of a connection between them — geocoding.

Thus, we have a set of points (sensors, denoted by Z), which are the main source of information space formation. Any point has a number of properties (denoted by y^i , where i is the number of the object) that can be used to predict (together with the data of the main sources) the required controlled indicators. In particular, these indicators include consumed electricity.

Accordingly, we represent the subject area under study as a set of interacting objects and a set of relations between these objects. We get the ability to visualize a

Table 1
Fragment of the database for the formation of the artificial neural network dataset

Date	Meter number	Consumed electricity, kWh	Air temperature, °C
Jan 01, 2018	1	7156.5	−8.3
Jan 02, 2018		7567.1	−9.4
Jan 03, 2018		8123.2	−14.4
Jan 04, 2018		8119.3	−11.4
Jan 05, 2018		8912.0	−12.2
Jan 06, 2018		7118.4	−11.1
Jan 07, 2018		7945.2	−11.9
Jan 08, 2018		7981.5	−8.1
Jan 09, 2018		8589.0	−13.2
Jan 10, 2018		8611.7	−13.1

set of such objects in the form of a geometric coverage of the entire subject area by some areas — clusters.

We get that each main object (sensor) together with some territory (for example, a substation) is a limited area. A set of such areas form the domain space. These areas change dynamically over time, i. e., when changing some attributes, the boundaries of the substation change. When two objects enter the same area, they interact through relationships with subsequent changes in certain parameters. Thus, a new information unit arises that uniquely corresponds to the interaction process, in which there are new characteristics of this formed object — a virtual process (for example, an increased load of energy consumption). Considering this process at different levels, ranging from physical interaction to abstract components, we obtain the interaction of objects determined by functor relations. Functor relations do not affect the external relations under consideration, but new information objects appear that form multidimensional polycubes. As a result, by performing clustering on some significant for calculating energy consumption, we get the opportunity to provide a forecast of the necessary information even on the missing data from some sensors.

Selecting objects of the same class, we determine the correspondence of a certain set of properties, for example:

- location of sensors in one substation;
- belonging of the sensors to the same electrical network;
- the possibility of calculating distances between points of one cluster (no more than the cluster size), etc.

All this makes it possible to predict some time parameters of electricity consumption, its levels and dynamics.

Under the procedure for forming classes, we will understand the ordering of objects according to their similarity by the clustering method. By clustering we will understand the representation of objects as a vector of numbers that determine the geometric location of the object in space, the coordinates of which are the attributes of the object. A group of objects that form a region that is compact in some sense in space will be called a cluster. With incomplete or fuzzy information about the object, clusters may have blurred boundaries and overlaps with each other.

Thus, the vector of object parameters is the basis for dividing objects into clusters, and we will use the set of domain objects as a set of clustering objects.

This set is determined by the set of attributes, which are the characteristics of the objects of the analyzed subject area. Through the use of fuzzy logic, we assert that for each object in a certain quantitative scale all attribute values are measured, that is, the vector is

correlated with coordinates from the values of the specified scale. Proceeding from this, we introduce the notation: $Z = \{z_1, \dots, z_n\}$ is the set of objects of the subject area (monitoring sensors), where n is the number of objects; $y^i = \{y_1^i, \dots, y_{p_i}^i\}$, $i = 1, \dots, n$ is a set of properties (in the form of a parameter vector of the i -th object) that characterize clustering objects (i. e., they are the basis for combining objects into clusters); p^i is the number of attributes characterizing the i -th object. Currently, one of the most common fuzzy clustering algorithms is the fuzzy c-means algorithm. At the same time, the main difference between fuzzy clustering methods and clear clustering methods is the introduction of membership functions for different clusters for each object.

Let's introduce the notation: F is the set of fuzzy clusters of the subject area under consideration. Each cluster will include one or more objects of the subject area, then $F = \{F_1, \dots, F_k\}$ is a set of fuzzy clusters of the subject area, where k is their number. Let's introduce the notation: $\mu^F = \{(\mu_1^1, \dots, \mu_k^1), \dots, (\mu_1^n, \dots, \mu_k^n)\}$ is the set of membership functions of objects in fuzzy clusters.

We also introduce a constraint $\sum_{j=1}^k \mu_j^i = 1$, which means

that the sum of the values of membership functions in all clusters for each object i is equal to 1. Since we are interested in the coverage of the entire set of objects by clusters, we can introduce an additional constraint

$\sum_{i=1}^n \mu_j^i \geq 1$, i. e., fulfillment of this condition for each

cluster $j = 1, \dots, k$. Then the assessment of the i -th object by attribute y_j^i , $j = 1, \dots, k$, can be represented by the membership function $\mu_j^n(y^i)$.

Now we can calculate the coordinates of the centers of clusters, denote v^j by formula:

$$v^j = \frac{\sum_{i=1}^n \left(\mu_j^i(y^i)^m \sum_{l=1}^{p_i} y_l^i \right)}{\sum_{i=1}^n \mu_j^i(y^i)^m},$$

where m is a hyperparameter that determines the fuzziness of the cluster: the larger the value of m , the fuzzier the cluster $\sum_{l=1}^{p_i} y_l^i$ will be ($l = 1, \dots, p_i$ for each

clustering object, i calculates the sum of the properties of the parameter vector).

Denoting a moving point in the power grid as a virtual object, we come to the conclusion that there is a certain control for it, consisting of:

- 1) sets of possibilities (set of routes)
 $Sv = Fv \{Fv^1, \dots, Fv^k\}$;

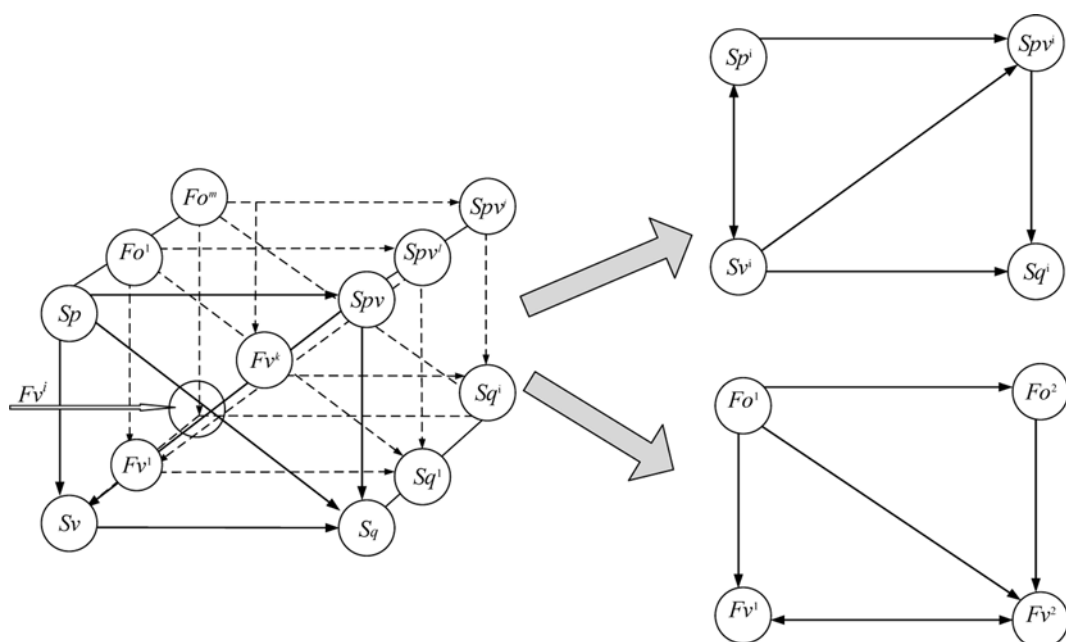


Figure 1. Multidimensional scheme of interaction of point parameters (in accordance with the Descartes square rule)

2) sets of restrictions (a set of restrictions on the parameters of the main objects and parameters of points: $Sp = Fo\{Fo^1, \dots, Fo^m\}$;

3) incident management systems (the occurrence of emergency situations, etc.) in the form of a set of $Spv = Sv \circ Sp$ and, respectively, $Spv^j = Sv^j \circ Sp^j$ ($j = 1, \dots, l$), where j is the number of the incident, and l is the number of incidents;

4) case management systems in the form of a set $Sq = Sv \circ Spv$ (a set of exceptional situations, etc.) and, respectively, $Sq^j = Sv^j \circ Spv^j$ ($j = 1, \dots, r$) where j is the number of the precedent, and r is the number of precedents.

Obviously, $F = Fo \cup Fv$. Thus, we arrive at a multilevel forecasting system with the possibility of imputing data and predicting incidents, translating them into precedents, and building chains of possible system states (Figure 1) based on the membership function. The specific type of membership functions is determined only taking into account the specifics of the existing uncertainty, the real situation on the object and the number of degrees of freedom in the functional dependence, which determines the calculation of the function itself in artificial neural network based on the processing of a temporal data set.

Approbation of the data imputation method on the example of time series forecasting

It is proposed to test and conduct a comparative analysis of the data imputation method using the

example of time series forecasting [10, 11]. The data (Table 1) necessary for the experiment were obtained at the Kumertau Electric Networks Production Association of Bashkirenergo LLC. The experiment itself includes the predicted values [12] of the amount of electricity consumed before data imputation (Table 2) with their subsequent comparison with the predicted values after data imputation (Table 3). For convenience, the values are given as average errors by day.

Also, within the framework of this study, a comparative analysis was carried out (Table 3, Figure 2) with the following data imputation methods (implemented in the Python programming language using the scikit-learn library, Table 4).

Thus, due to data imputation, the accuracy of the predictive values calculated by the artificial neural network [12] increased and the improvement was from 3 kWh (mean/median method) to 15 kWh (author's method).

Table 2

Average errors of predictive values of the amount of energy consumed (on the example of the Kumertau Electric Networks enterprise), obtained based on the calculation by an artificial neural network without data imputation

Total average forecast error, kWh	Average error in the forecast for 5 days, kWh	Average error in the forecast for 4 days, kWh	Average error in 3-day forecast, kWh	Average error in the forecast for 2 days, kWh	Average error in the forecast for 1 day, kWh
308.39	448.85	369.55	300.90	205.20	217.45

**Comparative analysis of average errors
of data imputation methods**

Author of the article, kWh	Average/median, kWh	One-dimensional recovery, kWh	k-nearest neighbors, kWh
293.40	305.18	298.04	299.65

Table 3

to improve their accuracy, which is what this study is aimed at. A data imputation method for neural network forecasting of the amount of electricity consumed was developed and tested on the example of the Kumertau Electric Networks enterprise.

The analysis of the effectiveness of the proposed method showed that the use of the proposed method of data imputation, implemented in the Python programming language, when predicting electricity consumption is appropriate and effective: the accuracy of the forecast increases (by 15 kWh), including with respect to other methods.

Thus, the proposed method for predicting electricity consumption can be useful for companies implementing specialized energy-saving measures, which is especially important in today's world economic realities.

Increasing the accuracy of forecast values helps to reduce the costs of energy enterprises for reserving electric power capacities to provide consumers with electricity.

Conclusion

Given the widespread use of artificial intelligence technologies, including in forecasting, there is a need

Table 4

Average errors of predictive values of the amount of energy consumed (on the example of the enterprise Kumertau Electrical Networks), obtained on the basis of the calculation by an artificial neural network, taking into account data imputation

Data imputation methods	Total average forecast error, kWh	Average error in the forecast for 5 days, kWh	Average error in the forecast for 4 days, kWh	Average error in 3-day forecast, kWh	Average error in the forecast for 2 days, kWh	Average error in the forecast for 1 day, kWh
Mean/median	305.18	441.25	365.37	299.68	201.64	218.00
Method of the author of the article	293.40	420.00	370.10	294.07	187.98	194.87
One-dimensional data recovery	298.04	440.14	362.17	297.00	200.15	190,76
K-nearest neighbors	299.65	442.72	364.83	298.43	199.23	193.02

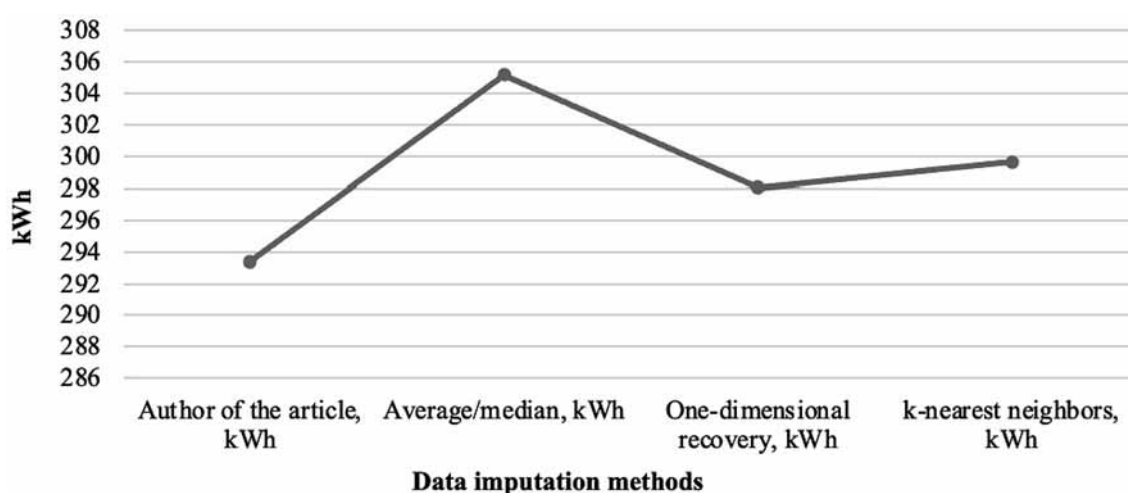


Figure 2. Comparative analysis of data imputation methods in time series forecasting

References

1. Guo-Feng F., Xiao W., Ya-Ting L., Wei-Chiang H. Forecasting electricity consumption using a novel hybrid model. *Sustainable Cities and Society*, 2020, vol. 61, pp. 102320. DOI: 10.1016/j.scs.2020.102320.
2. Wei-Chiang H. *Hybrid Intelligent Technologies in Energy Demand Forecasting*, Berlin, Springer, 2020, 192 p.
3. Losev A. S., Massel A. G. Prediction of solar radiation and data imputation for the digital twin of a solar power plant, *Information and Mathematical Technologies in Science and Management*, 2022, no. 1 (25), pp. 91—101. DOI: 10.38028/ESI.2022.25.1.008 (in Russian).
4. Kurapeev D. I., Lushnov M. S., Man T., Zhukova N. A. Imputation and system modeling of acid-base state parameters for different groups of patients, *Scientific and technical journal of information technologies, mechanics and optics*, 2022, vol. 22, no. 1, pp. 155—166. DOI: 10.17586/2226-1494-2022-22-1-155-166.
5. Boseong S., Jaekyung S., Taejin K., Byeng D. Y. Missing data imputation using an iterative denoising autoencoder (IDAE) for dissolved gas analysis, *Electric Power Systems Research*, 2022, vol. 212, pp. 108642. DOI: 10.1016/j.epsr.2022.108642.
6. Yong C., Qixun C. A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction, *Transportation Research Part C: Emerging Technologies*, 2022, vol. 143, pp. 103820. DOI:10.1016/j.trc.2022.103820.
7. Daoming W., Roozbeh R. F., Mehrdad S., Niloofar M. COLI: Collaborative clustering missing data imputation, *Pattern Recognition Letters*, 2021, vol. 152, pp. 420—427. DOI:10.1016/j.patrec.2021.11.011.
8. Shatha A., Hossam F., Hazem H. EvoImputer: An evolutionary approach for Missing Data Imputation and feature selection in the context of supervised learning, *Knowledge-Based Systems*, 2022, vol. 236, pp. 107734. DOI: 10.1016/j.knosys.2021.107734.
9. Gideon A. L., Mohd Z., Anoop Krishman N. M., Suman-ta D. Prediction of concrete strengths enabled by missing data imputation and interpretable machine learning, *Cement and Concrete Composites*, 2022, vol. 128, pp. 104414. DOI: 10.1016/j.cemconcomp.2022.104414.
10. Palchevsky E. V., Antonov V. V. Decision Support System based on Application of the Second Generation Neural Network, *Programmnaya Ingeneria*, 2022, vol. 13, no. 6, pp. 301—308. DOI: 10.17587/prin.13.301-308.
11. Palchevsky E. V., Antonov V. V., Enikeev R. R., Breikin T. A system based on an artificial neural network of the second generation for decision support in especially significant situations, *Journal of Hydrology*, 2023, vol. 616, pp. 128844. DOI: 10.1016/j.jhydrol.2022.128844.
12. Palchevsky E. V., Antonov V. V., Kromina L. A., Rodionova L. E., Fakhrullina A. R. Artificial Neural Network for Forecasting Electricity Consumption in Energy Enterprises, *Programmnaya Ingeneria*, 2023, vol. 14, no. 1, pp. 34—41. DOI: 10.17587/prin.14.34-41.

ИНФОРМАЦИЯ

В рамках Международного конгресса «Суперкомпьютерные дни в России» **Нижегородский государственный университет им. Н. И. Лобачевского** проводит международную конференцию и молодежную школу «Математическое моделирование и суперкомпьютерные технологии». Проведение конференции (13—14 ноября) и молодежной школы (15—16 ноября) будет содействовать дальнейшему повышению синергетического эффекта от взаимовыгодного взаимодействия двух приоритетных направлений развития науки и техники — математического моделирования и суперкомпьютерных технологий.

Тематика конференции и молодежной школы охватывает широкий круг задач математического моделирования сложных процессов и численных методов их исследования и проблемы разработки методов суперкомпьютерных вычислений для решения актуальных задач в различных областях науки, промышленности, бизнеса и образования.

Конференция и молодежная школа проводятся под эгидой **Суперкомпьютерного консорциума университетов России** при партнерстве с Национальным центром НТИ «Технологии машинного обучения и когнитивные технологии» (СПбГУ ИТМО) и Национальным центром НТИ «Технологии хранения и анализа больших данных» (МГУ).

Организаторы: **Научно-образовательный математический центр «Математика технологий будущего»** и **Приволжский научно-образовательный центр суперкомпьютерных технологий** ННГУ им. Н. И. Лобачевского.

Рабочие языки: русский и английский.

Подробности: <http://agora.guru.ru/hpc2023>

XVI-я Всероссийская Мультиконференция по проблемам управления (МКПУ-2023)



**11 сентября – 15 сентября 2023 г.
г. Волгоград, Россия**

**ЗАРЕГИСТРИРОВАТЬСЯ
НА САЙТЕ <https://lomonosov-msu.ru/>**

**ПОСЛЕ РЕГИСТРАЦИИ ПЕРЕЙТИ ПО ССЫЛКЕ
<https://lomonosov-msu.ru/rus/event/8035/>**

ОРГАНИЗАТОРЫ И СПОНСОРЫ

- Российская академия наук
- Министерство науки и высшего образования РФ
- Научный совет РАН по мехатронике и робототехнике
- Научный совет РАН по теории и процессам управления
- Научный совет РАН по машиностроению
- Научный совет РАН по проблемам транспорта
- Волгоградский государственный технический университет
- ГНЦ РФ ОАО «Концерн «ЦНИИ «Электроприбор»
- Южный федеральный университет
- Казанский (Приволжский) федеральный университет
- Национальный центр физики и математики
- НИИ многопроцессорных вычислительных систем ЮФУ
- НИЦ суперЭВМ и нейрокомпьютеров
- Институт проблем управления им. В.А. Трапезникова РАН
- Институт проблем механики им. А.Ю. Ишлинского РАН
- Национальный исследовательский центр им. проф. Н.Е. Жуковского»
- Московский автомобильно-дорожный государственный технический университет
- ПАО «КАМАЗ»
- АО ФНПУ «Титан-Баррикады»
- НПО Специальных материалов

ЦЕЛЬ МУЛЬТИКОНФЕРЕНЦИИ

Обсуждение результатов фундаментальных и прикладных исследований в области процессов управления и их практического применения в различных сферах человеческой деятельности.

СОСТАВ МУЛЬТИКОНФЕРЕНЦИИ

Мультиконференция включает **четыре локальные научно-технические конференции:**

- **Робототехника и мехатроника (РиМ-2023)**, председатель Программного комитета – академик РАН Ф.Л. Черноусько, сопредседатель – академик РАН И.А. Каляев
- **Управление в распределенных и сетевых системах (УРСС-2023)**, председатель Программного комитета – академик РАН И.А. Каляев, сопредседатели – академик РАН Д.А. Новиков, член-корр. РАН Р.Л. Смелянский
- **Управление аэрокосмическими системами (УАКС-2023)**, председатель Программного комитета – академик РАН С.Ю. Желтов, сопредседатель – член-корр. РАН К.И. Сыпало
- **Управление в перспективных наземных транспортных системах (УПНТС-2023)**, председатель Программного комитета – член-корр. РАН В.М. Приходько, сопредседатель – д.т.н. Г.О. Котиев

Проведение Мультиконференции МКПУ-2023 планируется в **очном формате**.

ПРЕЗИДИУМ МУЛЬТИКОНФЕРЕНЦИИ

Председатель Президиума:

В.Г. Пешехонов академик РАН (ГНЦ РФ АО «Концерн «ЦНИИ «Электроприбор», Санкт-Петербург)

Члены Президиума:

И.В. Бычков	академик РАН (ИНЦ СО РАН, Иркутск)	Р.Л. Смелянский	член-корр. РАН (МГУ, Москва)
С.Н. Васильев	академик РАН (ИПУ РАН, Москва)	К.И. Сыпало	член-корр. РАН (ЦАГИ, Жуковский)
С.Ю. Желтов	академик РАН (ГосНИИАС, Москва)	Е.А. Федосов	академик РАН (ГосНИИАС, Москва)
И.А. Каляев	академик РАН (НИИ МВС ЮФУ, Таганрог)	Ф.Л. Черноусько	академик РАН (ИПМех РАН, Москва)
Д.А. Новиков	академик РАН (ИПУ РАН, Москва)	С.Л. Чернышев	академик РАН (ЦАГИ, Жуковский)
В.И. Лысак	академик РАН (ВолГТУ, Волгоград)	В.А. Шурыгин	член-корр. РАН (АО ФНПУ «Титан-Баррикады»)
Г.А. Полов	академик РАН (НИИ ПМЭ МАИ, Москва)	Е.И. Якушенко	академик РАН (НПО Специальных материалов, Санкт-Петербург)
В.М. Приходько	член-корр. РАН (МАДИ, Москва)		
М.В. Сильников	член-корр. РАН (НПО Специальных материалов, Санкт-Петербург)		

МЕЖДУНАРОДНЫЙ КОНГРЕСС Суперкомпьютерные дни в России

23 СЕНТЯБРЯ – 30 СЕНТЯБРЯ 2023 г.

Конгресс объединяет научную конференцию, научные школы суперкомпьютерной академии, серию специализированных научных семинаров, экскурсии в ведущие суперкомпьютерные центры и множество других событий, проводимых на различных площадках Москвы и России.

Международная научная конференция

Суперкомпьютерные дни в России

25.09 – 26.09.2023

Конференция проводится в течение двух дней и включает множество параллельных мероприятий: доклады мировых лидеров HPC-индустрии, научные секции, тренинги и семинары, постерную секцию, конференцию молодых ученых, выставку суперкомпьютерных технологий.

Научные школы

Суперкомпьютерная Академия

23.09 – 30.09.2023

Научные школы Суперкомпьютерной Академии – специализированные мероприятия по актуальным направлениям развития науки и технологий.

- ❖ Технологии параллельного программирования MPI и OpenMP.
- ❖ Высокопроизводительные вычисления на кластерах с использованием графических ускорителей NVIDIA.
- ❖ Применение платформы Python для высокопроизводительных вычислений.
- ❖ Квантовая информатика.
- ❖ Реализация глубоких нейросетей на высокопроизводительных кластерах.

Семинары и мастер-классы

23.09 – 30.09.2023

Темой семинара может быть любая тема, связанная с высокопроизводительными вычислениями, усиливающая или дополняющая тематику конгресса.

Форма семинара определяется его руководителем. Семинары могут проводиться на площадке научной конференции, на одной из площадок конгресса или же в формате телеконференции.

Предложения по проведению семинаров принимаются на конкурсной основе.

Выставка суперкомпьютерных технологий

25.09 – 26.09.2023

На выставке демонстрируются новейшие российские и зарубежные программные и аппаратные решения и технологии для высокопроизводительных вычислений от лидеров рынка HPC.

Подробности: <https://congress.russianscdays.org/>

ООО "Издательство "Новые технологии". 107076, Москва, ул. Матросская Тишина, д. 23, стр. 2
Технический редактор *Е. В. Конова.* Корректор *А. В. Чугунова.*

Сдано в набор 27.04.2023 г. Подписано в печать 31.05.2023 г. Формат 60×88 1/8. Заказ PI623
Цена свободная.

Оригинал-макет ООО "Авансес солюшнз". Отпечатано в ООО "Авансес солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru



ВСЕРОССИЙСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ **НАУЧНЫЙ СЕРВИС В СЕТИ ИНТЕРНЕТ**

Институт прикладной математики
имени М.В. Келдыша РАН
проводит с 18 по 22 сентября 2023 г.

XXV ВСЕРОССИЙСКУЮ КОНФЕРЕНЦИЮ **НАУЧНЫЙ СЕРВИС В СЕТИ ИНТЕРНЕТ**



www.keldysh.ru

Конференция посвящена направлениям и тенденциям использования интернет-технологий в современных научных исследованиях. Основная цель конференции – предоставить возможность для обсуждения, апробации и обмена мнениями о наиболее значимых результатах, полученных ведущими российскими учеными за последнее время в данной области деятельности.

Подробности на
agora.guru.ru/abrau2023/

ТЕМАТИКА КОНФЕРЕНЦИИ

- Научные исследования и Интернет, интернет-представительство научных организаций и проектов.
- Решение задач и обработка данных на суперкомпьютерах центров коллективного пользования.
- Интернет-проекты в области параллельных вычислений, математическое моделирование, вычислительные сервисы.
- Интернет-проекты для биомедицины.
- Модели и методы построения поисковых систем и систем навигации в интернете, технологии и системы распределенного хранения и обработки данных.
- Цифровые библиотеки и библиографические базы, семантический веб, наукометрия в интернете.
- Технологии и опыт построения информационных систем и баз данных, документации и результатов эксперимента на основе интернет-технологий.
- Онлайновая научная публикация, открытая наука, живая публикация, онлайн-рецензирование, мультимедийные иллюстрации.
- Популярный научный интернет, онлайн-энциклопедии, история науки в интернете.
- Интернет-активность ученого, персональная страница, профили ученого в библиографических базах, аттестация в интернете.
- Системное и инструментальное программное обеспечение, языки и модели программирования, формальные методы для интернет-технологий.

Издательство «НОВЫЕ ТЕХНОЛОГИИ» выпускает научно-технические журналы



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



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

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

Ежемесячный
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

Адрес редакции журналов для авторов и подписчиков:

107076, Москва, ул. Матросская Тишина, д. 23, стр. 2, оф. 45. Издательство "НОВЫЕ ТЕХНОЛОГИИ".

Тел.: (499) 270-16-52. E-mail: antonov@novtex.ru