ПРОБЛЕМЫ ТЕОРЕТИЧЕСКОЙ ИНФОРМАТИКИ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

УДК 004.9

ОПТИМИЗАЦИЯ РАБОТЫ С ДАННЫМИ В МЕТОДОЛОГИИ VB-MAPP

С. Ю. Балашева, И. А. Демидова, Д. В. Кучеренко

Воронежский государственный университет

Аннотация. В данной статье на примере работы компании «АутМама» рассматривается VB-MAPP — метод тестирования детей с расстройствами аутистического спектра. Особое внимание уделяется оптимизации процесса записи, хранения и поиска результатов тестирования, что позволяет повысить скорость работы пользователей с данными и их анализа. В статье также предлагается внедрение в структуру базы данных дополнительных сущностей, например, участников тестирования и индивидуальных медицинских особенностей, которые могут улучшить качество диагностики и обеспечить более полное понимание возможностей и потребностей каждого ребенка.
Ключевые слова: VB-MAPP, методологии тестирования, аутизм, РАС, ABA, оценка навыков, изучение особенностей, оптимизация структуры.

Введение

Тестирование детей с расстройствами аутистического спектра (РАС) представляет собой важную область исследования и практики, направленную на выявление и понимание особенностей их развития и учебных потребностей. По данным ВОЗ, ежегодно число детей с аутизмом увеличивается на 13 % в мире [1] и на 12–18 % (по данным Министерства Просвещения РФ) в России [2]. Увеличение числа таких детей указывает на необходимость разработки и внедрения более качественных и точных инструментов для их диагностики и сопровождения.

Одним из эффективных инструментов для оценки навыков и поведения детей с РАС является метод тестирования Вербально-Бихевиорального Программирования VB-MAPP (Verbal Behavior Milestones Assessment and Placement Program). Этот метод разработан для анализа вербальных и социальных навыков у детей, позволяя специалистам получить детальное представление о текущем уровне развития и межличностных способностях.

VB-MAPP основан на методах прикладного анализа поведения и учитывает аспекты, которые особенно важны для успешной интеграции детей с РАС в общество. Статья направлена на изучение особенностей применения этого метода в процессе тестирования, на анализ его эффективности и практического применения в различных образовательных и терапевтических целях, а также на совершенствование данной системы для более результативного и структурированного использования.

1. Описание методологии VB-MAPP

VB-MAPP — это программа оценки навыков речи и социального взаимодействия для детей с аутизмом и другими нарушениями, а также это неотъемлемый инструмент в прикладном анализе поведения, который помогает как оценивать, так и разрабатывать поведенческие программы вмешательства, является частью образовательной программы ABA (Applied Behavior Analysis).

Программа была разработана в 2008 году Марком Сандбергом, и она основана на анализе речевого поведения Б.Ф. Скиннера, этапах развития ребенка и исследованиях в области анализа поведения [3].

Если проводить оценку VB-MAPP по всем правилам, тщательно тестируя каждый навык, то процесс проведения тестирования будет очень трудоемким и продолжительным. Проведе-

ние оценки уровня навыков 1-го уровня занимает от 2 до 3-х часов, тщательное тестирование навыков второго уровня — от 4-х до 6 часов, а навыков третьего уровня — 10–12 часов. Следовательно, если ребенок, которому специалист проводит оценку навыков, не говорит на данный момент, и демонстрирует, в общем, невысокий уровень навыков — проведение оценки навыков для такого ребенка может проводиться в течение одного занятия (2–3 часа). Но, если у ребенка уже сформированы некоторые речевые навыки, он неплохо понимает речь, то на проведение оценки навыков для такого ребенка потребуется 2–3 занятия, иногда больше. Если же у ребенка хорошо развиты речевые навыки, ребенок много и часто коммуницирует, то проведение оценки навыков для такого ребенка может занять довольно много занятий (6–10 занятий) [4].

Следовательно, обращаясь к специалисту, необходимо обратить внимание, насколько качественно проведена оценка навыков. К примеру, если специалист отчитывается о проведенной оценке навыков и отдает родителям разработанную программу после первого же занятия, то это может навести на подозрение о недобросовестных действиях специалиста.

По данным сайта «ФРЦ Аутизм» программа тестирования VB-MAPP включает в себя такие разделы:

1. «Вехи развития», содержащий 170 вех (навыков) развития трех возрастных уровней типично развивающихся детей (1 уровень — 0–18 месяцев, 2 уровень — 18–30 месяцев, 3 уровень — 30–48 месяцев). Предназначен для определения имеющихся у ребенка вербальных и социальных навыков.

2. «Барьеры», где специалист определяет преграды, мешающие обучению ребенка и темы, в которых у ребенка могут возникать проблемы. Под барьерами понимаются проблемы с поведением, контролем, зависимостью, самостимуляцией, неспособностью к обобщению, слабой мотивацией, сенсорной чувствительностью, дефектами в вербальных реакциях, социальных навыках, зрительном восприятии, имитации и артикуляции.

3. «Самопомощь», где прописаны 18 областей развития и их оценки, показывающие готовность ребенка к обучению в естественной среде.

4. Раздел «Анализ заданий и мониторинг приобретения навыков», дополняющий раздел «Вехи развития». Представляет собой перечень дробных преднавыков более раннего уровня, необходимых для достижения цели.

5. Также есть еще один компонент, содержащий в себе индивидуальные планы и цели обучения. Эта часть помогает сбалансировать программу вмешательства и обеспечить включение всех необходимых для конкретного ребенка задач.

2. Построение моделей

Как было сказано ранее, тестирование по методике VB-MAPP состоит из четырех основных частей, на основе которых формируется пятая — индивидуальный план развития ребенка. Несмотря на то, что мы не являемся педагогами и не обладаем квалификацией для непосредственного проведения программы VB-MAPP, в нашей работе мы опираемся на опыт и практические наработки компании «АутМама». Используя их подход как основу, мы стремимся адаптировать методологию VB-MAPP к специфическим потребностям сотрудников данной организации, чтобы сделать процесс тестирования более понятным и структурированным.

В ходе исследования существующей системы было выявлено, что ее текущая структура неудобна для практического использования. Система содержит множество разрозненных таблиц, затрудняющих хранение информации и поиск конкретных тестов для ребенка.

Предполагается разработка информационной базы на платформе «1С:Предприятие 8», благодаря ее широким возможностям для автоматизации бизнес-процессов и гибкой настройке под конкретные потребности.

Более детальное описание объектов, а также полное раскрытие теоретических и практических аспектов внедрения системы будут представлены в соответствующих дипломных работах. Это позволит углубленно рассмотреть методологию и реализацию, а также продемонстрировать преимущества использования информационной базы на платформе «1С:Предприятие 8» для оптимизации процесса диагностики и сопровождения детей с РАС.

Для удобства понимания основные и некоторые дополнительные сущности рассмотрены на конкретных примерах.

Для начала была разработана таблица вех (табл. 1). В этой таблице каждая веха представляет собой определенную группу навыков, например, «Социальные навыки». Для каждой вехи предусмотрены уровни развития, которые отражают прогресс ребенка, начиная с минимального уровня и заканчивая более высоким.

Каждый уровень характеризуется балльной оценкой, позволяющей количественно измерить достижения ребенка, а также описанием, которое детализирует поведенческие особенности и навыки, проявляемые на данном уровне. Например, в рамках социальных навыков минимальный уровень (1) соответствует начальным проявлениям общения, тогда как более высокий уровень предполагает активное взаимодействие, например, частые ответы на просьбы.

Такой формат позволяет специалистам легко отслеживать динамику развития ребенка и адаптировать индивидуальный план на основе прогресса, что делает систему более структурированной и удобной для использования.

Таблица 1

Пример заполнения вех развития

Веха	Уровень	Баллы	Описание
Социальные навыки	1	0,5	Минимально общается
		1	Иногда инициирует общение
		1,5	Часто отвечает на просьбы
		2	Инициирует общение при необходимости, редко реагирует на просьбы
	2	2,5	Относительно свободно отвечает на просьбы
		3	Общение может инициировать по просьбе, спонтанно или при необходимости
…	…	…	…

Кроме вех развития, были выделены барьеры (табл. 2), которые могут препятствовать достижению определенных навыков. Для каждой вехи предусмотрены уровни барьера, описывающие степень сложности, с которой ребенок может сталкиваться, например, от «слабо выражен» до «крайне трудна». Такая структура позволяет учесть индивидуальные трудности ребенка и скорректировать подход к обучению с учетом его потребностей.

Навыки самопомощи представлены в табл. 3. Они состоят из основных групп, такие как «Одежда» и «Прием пищи», а каждая группа содержит перечень конкретных навыков, необходимых для выполнения базовых действий. В столбце «Наличие» отмечено, освоен ли данный навык, что позволяет быстро оценить уровень самостоятельности ребенка в повседневной жизни и определить области, требующие дополнительной поддержки.

Анализ заданий служит дополнением к основной системе вех, позволяя более детально оценить навыки ребенка (табл. 4). Если ребенок пока не соответствует требованиям определенного уровня, например, второго, специалисты могут отметить, какие конкретные дополнительные задания из предыдущих этапов ему уже удалось освоить.

В таблице для каждой вехи представлены уровни и подуровни, обозначенные буквами, например, 1а, 1б и т. д. Для уровня 1 в категории «Социальные навыки» определены навыки, та-

Таблица 2

Пример заполнения барьеров

Веха	Уровень барьера	Описание
Социальные навыки	1	Слабо выражен
	2	Имеет трудности
	3	Имеет значительные трудности
	4	Эта тема крайне трудна для ребенка
ТАКТ	1	Слабо выражен
	2	Имеет трудности
…	…	…

Таблица 3

Пример заполнения тестирования «Самопомощь»

Группа	Навык	Наличие
Одевание	Снимает одежду	Да
	Обувается	Нет
	Надевает одежду	Да
Прием пищи	Ест ложкой без помощи	Нет
…	…	…

Таблица 4

Пример заполнения «Анализа заданий»

Веха	Уровень	Баллы	Описание
Социальные навыки	1	1а	Смотрит, когда к нему обращаются
		1б	Может спонтанно ответить
		1в	Иногда спонтанно инициирует общение
		1г	Иногда спонтанно отвечает и говорит
	2	2а	Редко отвечает, когда спрашивают
		2б	Отвечает, когда спрашивают
…	…	…	…

кие как способность смотреть на собеседника, реагировать на обращение и иногда инициировать общение. Эта структура позволяет более гибко оценивать навыки, фиксируя достижения ребенка, даже если он еще не достиг следующего уровня, и при необходимости адаптировать учебный план под его текущие возможности.

В рамках данной системы также были введены дополнительные сущности для учета информации о сотрудниках, детях и родителях. Эти данные позволяют более эффективно организовать процесс взаимодействия всех участников.

Также предлагается возможность дополнить систему функцией составления расписания занятий, что упростит кураторам ориентирование по занятиям и обеспечит планирование работы с детьми.

Расписание составляется вручную путем записи ребенка на занятия в доступное на данный момент время. Предполагается, что сформированное расписание может просматриваться в виде таблиц, аналогичных табл. 5.

Все описанные выше схемы и объекты имеют удобную структуру, могут быть созданы в конфигурации, разрабатываемой на платформе «1С:Предприятие 8».

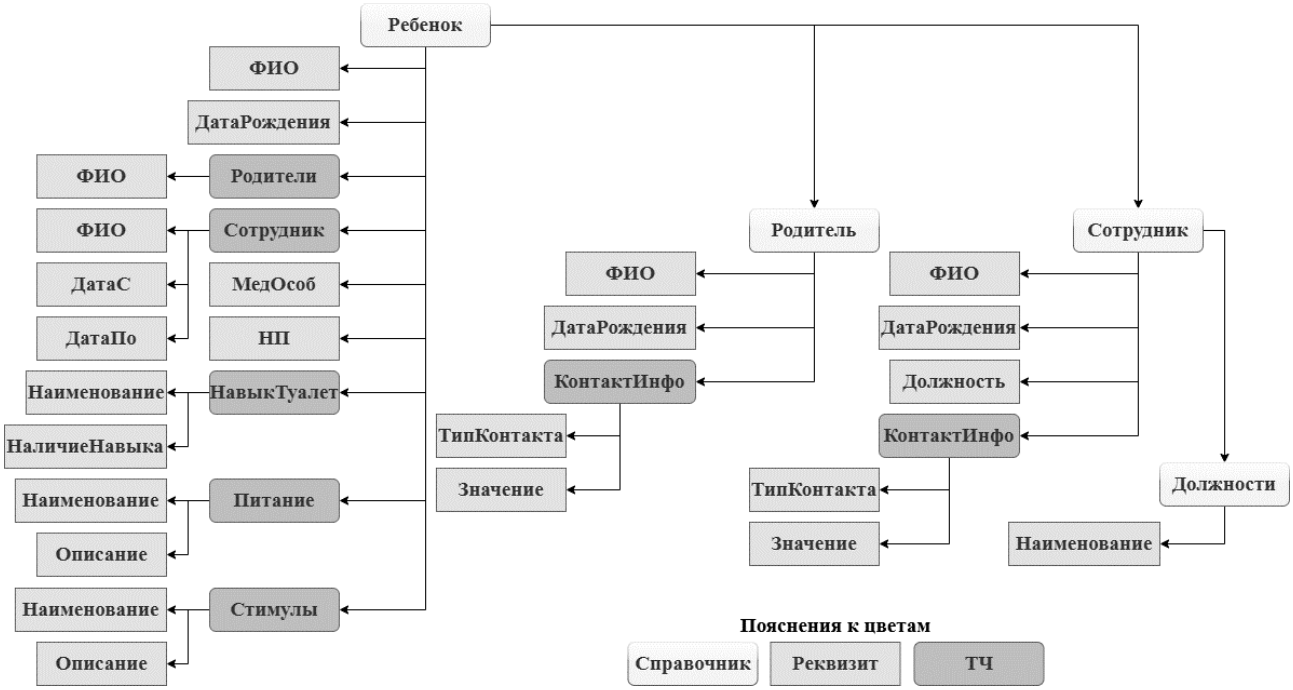


Рис. 1. Схема взаимоотношений дополнительных сущностей

Таблица 5

Шаблон для расписания

Дата в формате «День недели + ДД.ММ.ГГГГ»		
Время в формате «ЧЧ:ММ»	ФИО Ребенка 1	ФИО Сотрудника 1
Время в формате «ЧЧ:ММ»	ФИО Ребенка 2	ФИО Сотрудника 1
…	…	…

Такой подход обеспечивает централизованное хранение данных и упрощает доступ к информации, что делает процесс тестирования и сопровождения детей с РАС более эффективным и удобным для практического применения.

Заключение

Проведенное исследование подтвердило важность структурированного подхода к оценке навыков и развития детей с РАС, что позволяет более точно выявлять их сильные стороны и зоны, требующие поддержки. Методология VB-MAPP, адаптированная с учетом опыта компании «АутМама», показала свою эффективность в упрощении процесса тестирования и создании индивидуального плана для каждого ребенка.

Описанные схемы вех, барьеров, анализа заданий и навыков самопомощи обеспечивают комплексный подход к диагностике, позволяя фиксировать динамику развития ребенка и учитывать его индивидуальные особенности. Структурированное представление информации в таблицах также облегчает специалистам работу с данными, а разработка данной системы в виде информационной базы на платформе «1С Предприятие» открывает возможности для автоматизации процессов записи, хранения и анализа данных.

Литература

1. Всемирный день распространения информации о проблеме аутизма // Риа Новости: [сайт]. – 2022. – URL: https://ria.ru/20240402/autizm-1937186744.html (дата обращения: 28.10.2024).

2. Аналитическая справка о состоянии образования обучающихся с расстройствами аутистического спектра в субъектах Российской Федерации в 2022 году // ФРЦ Аутизм: [сайт]. – 2022. – URL: https://autism-frc.ru/ckeditor_assets/attachments/4263/analiticheskaya_spravka_monitoring_ras_2022_29_12_2022.pdf (дата обращения: 26.10.2024).

3. Программа оценки навыков речи и социального взаимодействия для детей с аутизмом и другими нарушениями (VB-MAPP) // ФРЦ Аутизм: [сайт]. – 2024. – URL: https://autism-frc.ru/early-help/metody/1420 (дата обращения: 25.10.2024).

4. *Эрц Ю. М.* VB-MAPP на русском языке — 5 наиболее распространенных «недочетов» при использовании / Ю. М. Эрц // Аутизм | ABA-терапия: [сайт]. – 2024. – URL: https://autism-aba.blogspot.com/2017/02/5-false-actions-in-using-vb-mapp.html (дата обращения: 28.10.2024).

УДК 004.415

АНАЛИТИЧЕСКИЙ ОБЗОР НЕКОТОРЫХ ТЕХНОЛОГИЙ ФУНКЦИОНАЛЬНОГО ТЕСТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ

Д. М. Башлыков

Воронежский государственный университет

Аннотация. В статье рассматриваются технологии мануального и автоматизированного тестирования одного из компонентов программного продукта. Целью данной статьи является раскрытие основных возможностей функционального тестирования, демонстрация примера проведения тестирования программного модуля ручными и автоматизированными средствами.

Ключевые слова: функциональное тестирование, шаблон бизнес-объектов, автоматизация тестирования, мануальное тестирование, позитивное тестирование, тест-кейс, Selenium, TestNg, Allure.

Введение

Рассматривается проблема проверки функциональности одного из модулей программного продукта «Горно-геологическая информационная система», именуемого как Реестр шаблонов бизнес-объектов.

1. Основные принципы и понятия функционального тестирования

Функциональное тестирование — вид тестирования, включающий в себя процесс проверки функциональности системы, работы системных компонентов на соответствие заявленным требованиям и ожиданиям пользователей.

Тестируются компоненты, доступные пользователю, такие как вход в систему, основные навигации по программному продукту, важнейшие бизнес-требования, а также работа с базой данных, взаимодействие клиента с сервером и другие.

Принято выделять жизненно-важные принципы функционального тестирования:

– проверка ожидаемого поведения;
– полнота тестов;
– использование реальных данных;
– документирование тестов;
– автоматизация и ручное тестирование.

2. Описание пользовательского интерфейса реестра шаблонов бизнес-объектов

Реестр представляет собой многофункциональную форму, на которой пользователь может создавать, удалять, редактировать любой из шаблонов как через правую панель (выезжающее справа меню), так и через табличное представление бизнес-объектов (рис. 1).

Каждый шаблон наделен своим собственным списком атрибутов. Атрибуты могут быть системными и пользовательскими.

Системные атрибуты обязательны к заполнению и являются основными характеристиками шаблона. Пользовательские атрибуты необязательны, и их создание — опционально.

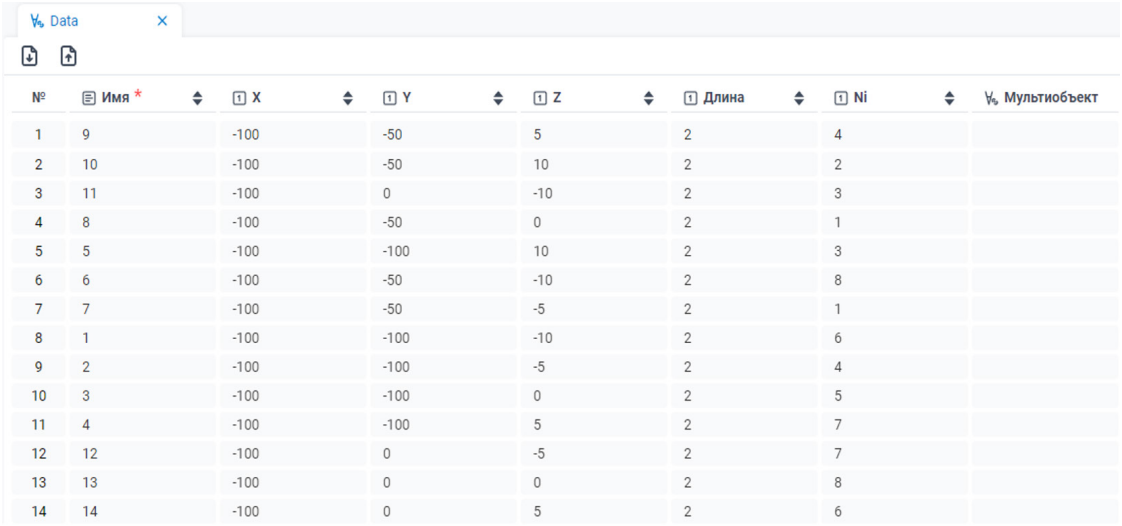


Рис. 1. Табличное представление шаблона

3. Мануальная проверка на создание шаблона бизнес-объектов

Также одним из важных критериев корректной работы табличного представления бизнес-объектов является создание нового шаблона прямо из вида в таблице (рис. 2).

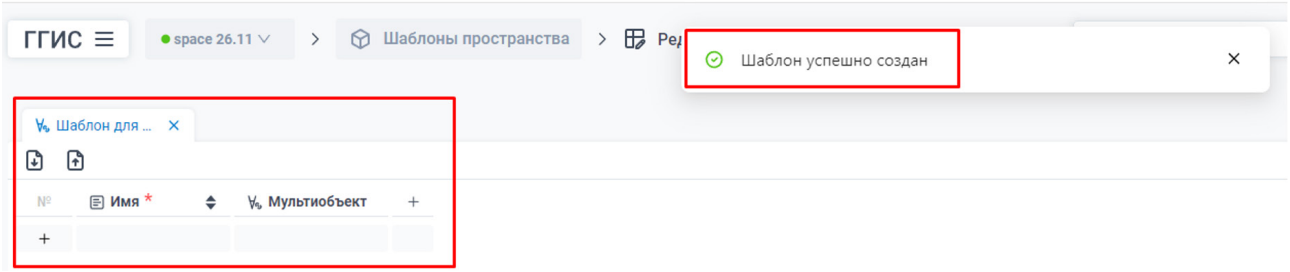


Рис. 2. Создание шаблона бизнес-объектов на стороне клиента

4. Проверка создания шаблона бизнес-объектов через API

При создании нового шаблона через табличное представление от клиента на сервер уходит Post-запрос на создание ресурса (рис. 3).

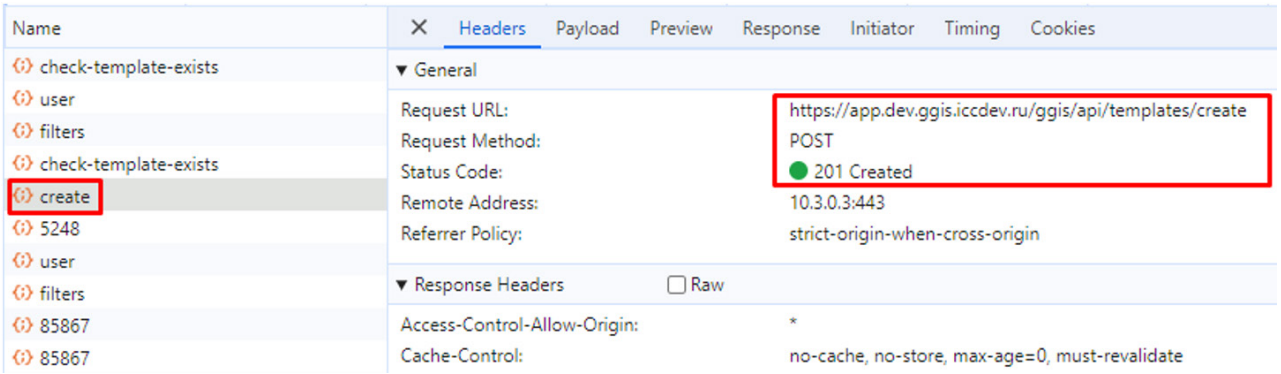


Рис. 3. Проверка корректности запроса на сервер о создание шаблона

Из запроса видно, что ресурс был успешно создан (получен статус ответа 201). Однако необходимо также проверить корректность ответа от сервера (response) со всеми его составляющими (статус-код, тело ответа в виде json-файла, время ответа и т. д.)

Для этого используется десктопное приложение Postman, которое служит помощником в тестирование прикладного программного интерфейса (API).

Сформировав запрос к тому же сервису, что и через пользовательский интерфейс приложения, заполнив все необходимые поля в теле запроса, передав токен аутентификации и отправив запрос, можно получить ответ от сервера (рис. 4).

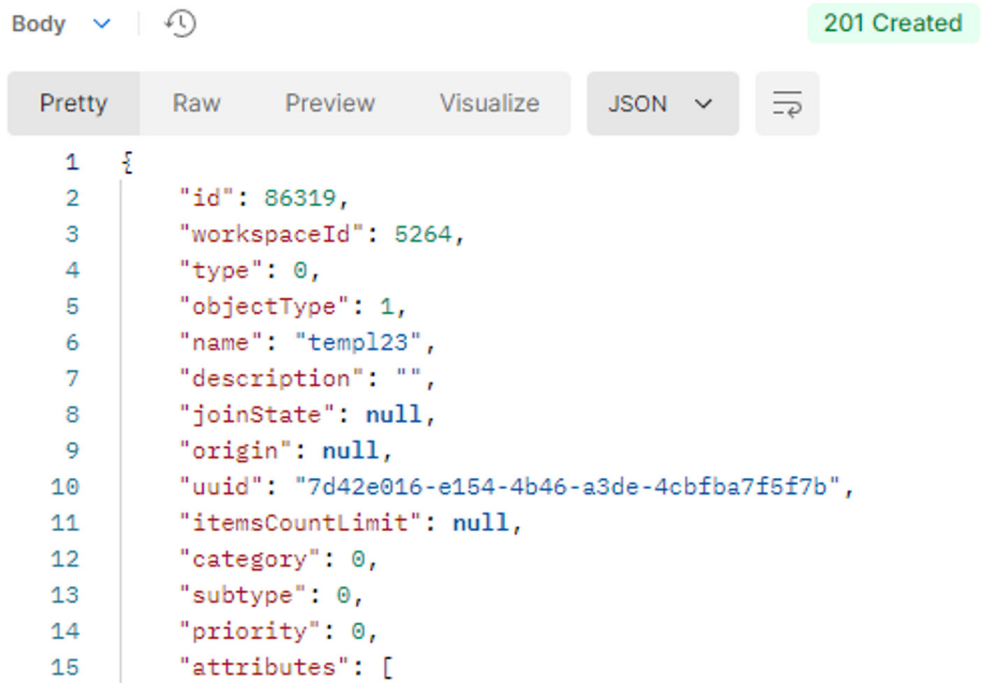


Рис. 4. Ответ от сервера в Postman

5. Проверка создания шаблона в базе данных

Для полноты проверки корректности работы данного компонента системы, а именно — создания шаблона бизнес объектов, необходимо проверить, что ресурс действительно появился на backend части приложения, в базе данных.

По исполнении следующего запроса SQL результирующая таблица будет состоять из шаблона бизнес объекта, созданного под учетной записью с логином ‘d.bashlykov’ в пространстве с id = 5248 и типом = 0 (т.е. пользовательский шаблон, а не шаблон по умолчанию) (рис. 5).

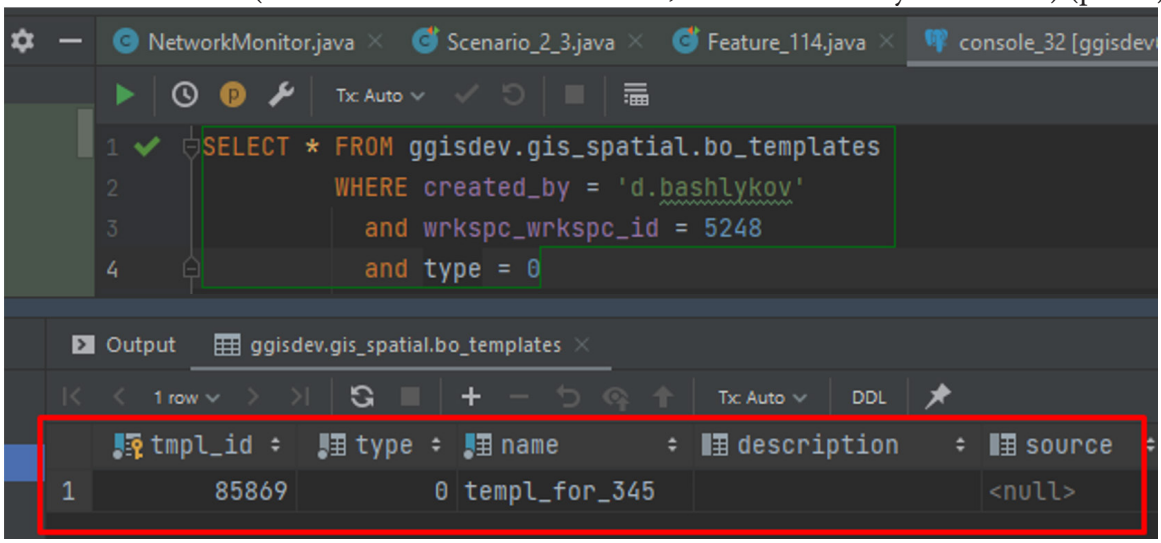


Рис. 5. Проверка существования шаблона в базе данных

6. Автоматизация тестирования на создание шаблона бизнес объектов

Проект по автоматизации тестирования как правило представляет собой слаженную, хорошо структурированную иерархию папок и пакетов с файлами. Чаще всего применяется объектная модель страницы для более гибкого подхода к автотестам.

Кейс на проверку создания нового шаблона бизнес-объектов подойдет для того, чтобы его автоматизировать. Для автоматизации тестирования в случае рассматриваемой задачи можно использовать библиотеку 'Selenium' вкупе с тестовым модулем 'TestNg'. Программный код реализован на языке программирования Java.

Результатом прогона данного сценария является отчет о результатах тестирования в среде Allure report (рис. 6).

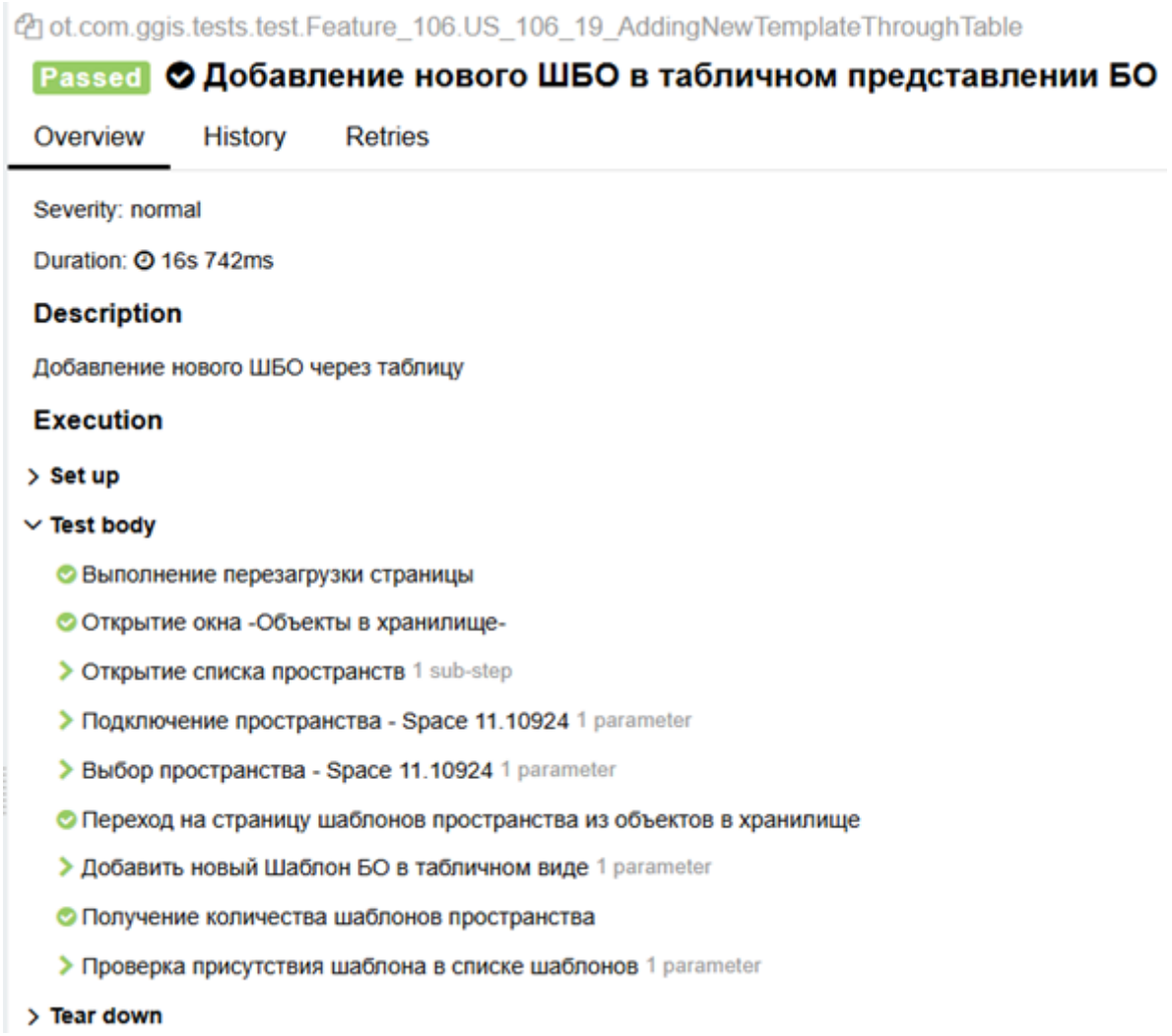


Рис. 6. Отчет в системе Allure

Заключение

Рассмотрена проблема тестирования одного из компонентов модуля системы и различные подходы к проверке части функциональности. Выявлены требования, проведены позитивные тестовые сценарии при применении мануального и автоматизированного способов тестирования. Анализ технологий тестирования показал, какими методами можно проводить проверку реализации каких-либо компонентов. Автоматизация тестирования способна покрывать

множество различных сценариев, что облегчает процесс проверки корректности функциональных возможностей программного продукта.

Литература

1. *Савин Р.* Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. – М. : Дело, 2007. – 312 с.

2. *Куликов С. С.* Тестирование программного обеспечения. Базовый курс. – 3-е изд. – Минск : Четыре четверти, 2020. – 312 с.

УДК 004.657

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕХАНИЗМОВ ИНДЕКСИРОВАНИЯ В ДВИЖКАХ MYISAM И INNODB В СУБД MYSQL

А. Э. Бурков

Воронежский государственный университет

Аннотация. В статье представлен анализ реализации B-tree индексов в MySQL на примере движков MyISAM и InnoDB. Рассматривается различие в структуре хранения данных: в MyISAM первичные и вторичные индексы идентичны, с хранением значений ключей и ссылок на строки данных в листьях, что позволяет таблицам существовать без первичного ключа. В InnoDB используется кластерный индекс, где строки данных размещаются непосредственно в листьях первичного индекса, а вторичные индексы содержат ссылки на первичный ключ. Такой подход ускоряет доступ к данным и обеспечивает их согласованность при обновлениях. Приведенные результаты показывают, что использование кластерного индекса в InnoDB значительно повышает эффективность работы с индексами по сравнению с MyISAM.

Ключевые слова: индексация данных, сбалансированное дерево поиска, производительность СУБД, таблица, движок хранения данных, база данных, кластеризация, ключ, строка.

Введение

Работа представляет собой сравнительный анализ устройства и возможностей использования индекса, основанного на сбалансированном дереве поиска (далее — B-tree).

В MySQL реализовано несколько типов индексов. Выбор типа индекса основывается на специфике задачи. Например, хэш индекс применяется для хранения данных в виде ключ-значение в оперативной памяти, FULLTEXT индекс используется для полнотекстового поиска данных, SPATIAL — для геоданных. Но в большинстве случаев применяется B-tree индекс, благодаря своей универсальности и эффективности. Данная структура обеспечивает логарифмическое время выполнения операций поиска.

MySQL предоставляет два основных движка хранения данных для таблиц: MyISAM и InnoDB. Первый использовался по умолчанию до версии 5.6, после чего основным стал InnoDB [1]. Оба движка предоставляют возможность создавать индексы для таблиц, но до версии 5.6.4 возможность полнотекстовой индексации была только у MyISAM, что делало его приоритетным, если такой тип индекса был необходим.

1. Организация индекса

1.1. Структура индекса в MyISAM

Схематично B-tree индекс для MyISAM представлен на рис. 1 [2].

В данной таблице индекс построен по числовому полю “ID”. В таком случае получается, что в узловых элементах находятся значения индекса и ссылки на более нижний узел или лист. В листовых элементах тоже находятся значения индекса, но они уже ссылаются на сами данные из таблицы.

Особенность MyISAM заключается в том, что для существования таблицы ей не нужен первичный ключ, для MyISAM нет разницы между первичным и вторичным индексами. Учитывая эту особенность, можно представить схематично данные индексы таким образом (рис. 2.):

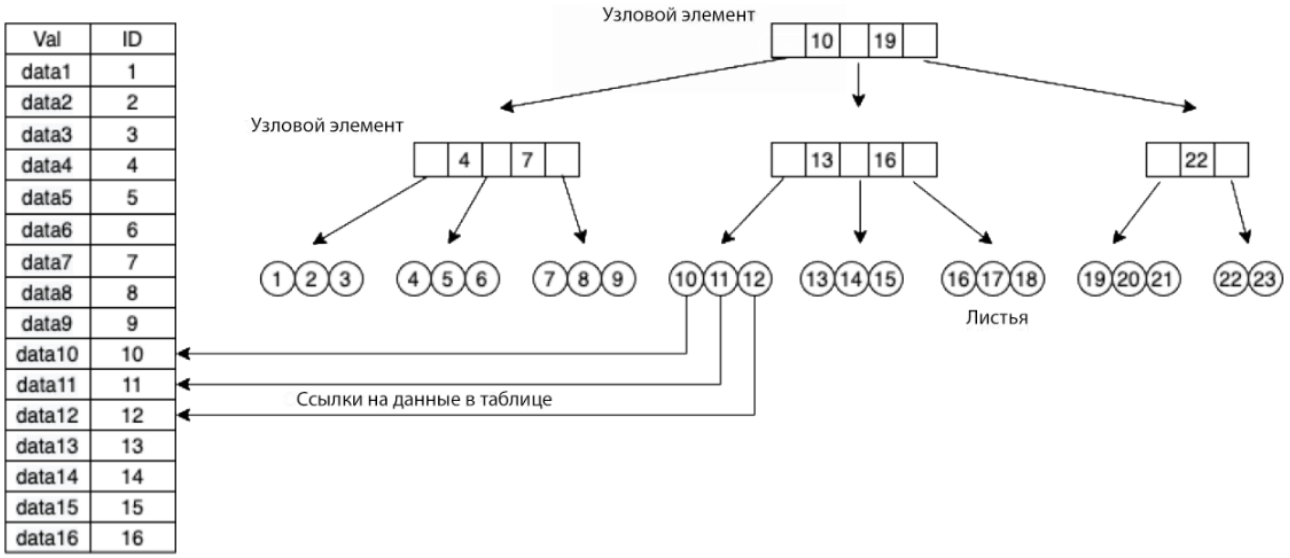


Рис. 1. Устройство B-tree индекса MyISAM

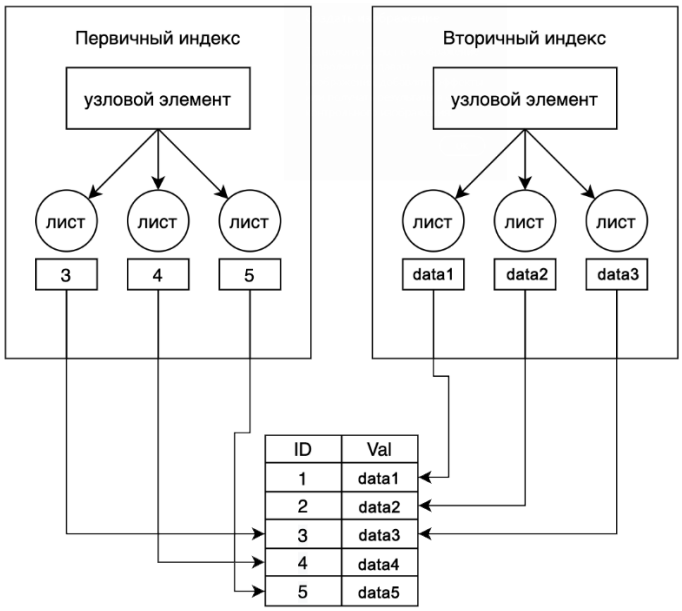


Рис. 2. Первичный и вторичный индексы MyISAM

На рисунке видно, что структура индексов одинаковая: первичный индекс построен по полю “ID”, а вторичный по полю “Val”. В обоих случаях в листьях расположены значения и ссылки на строки в таблице.

1.2. Структура индекса в InnoDB

InnoDB это более современный движок хранения данных, и в нем присутствуют различия в структуре между первичным и вторичным индексами. Таблица по-прежнему может существовать без первичного ключа, но в этом случае первичный ключ будет выбран автоматически по полю, по которому его можно создать. Это называется «суррогатный» первичный ключ. Если ни одно поле не подходит для создания первичного ключа, то InnoDB создаст новое числовое поле, которое будет скрыто.

В случае InnoDB используется концепция кластерного индекса [3]. Это означает, что индекс содержит не ссылки, а сами строки данных в листовых узлах. Схематично это можно представить на рис. 3.

Val2	Val	ID
v1	data1	1
v2	data2	2
v3	data3	3
v4	data4	4
v5	data5	5
v6	data6	6
v7	data7	7
v8	data8	8
v9	data9	9
v10	data10	10
v11	data11	11
v12	data12	12
v13	data13	13
v14	data14	14
v15	data15	15
v16	data16	16

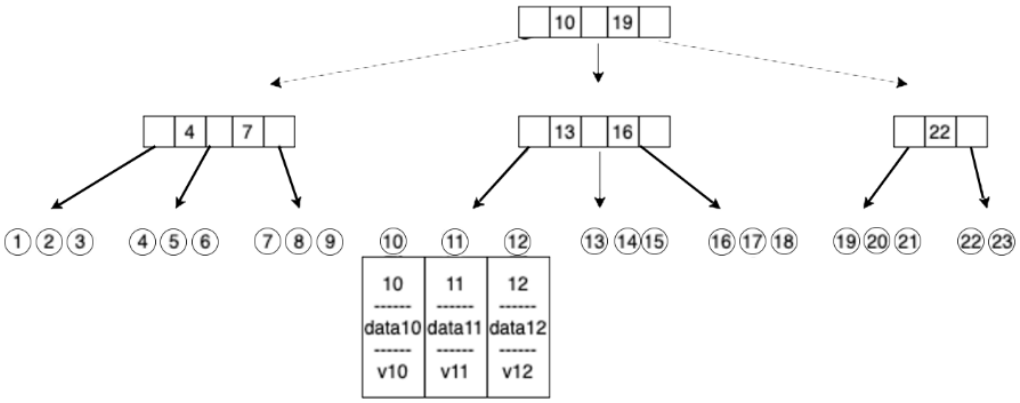
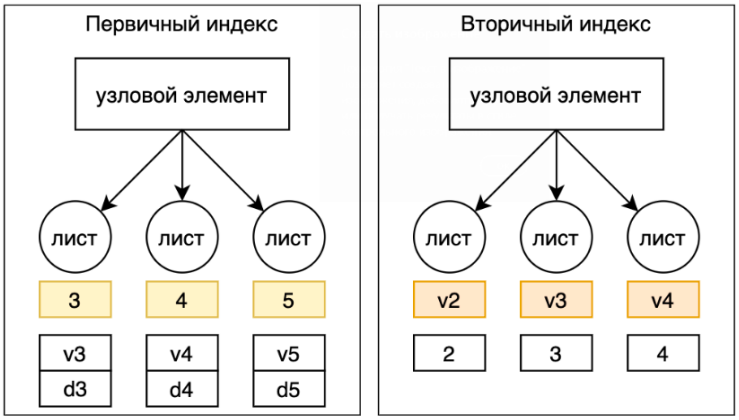


Рис. 3. Устройство B-tree индекса InnoDB

Как и в случае MyISAM индекс построен по числовому полю “ID”, но в данном устройстве индекса все данные строк хранятся в самом индексе. Это эффективно, так как позволяет избежать дополнительного чтения с диска при переходе от листа на данные в строке.

Так как у InnoDB первичный индекс отличается от вторичного, то их устройство также будут отличаться. На рис. 4 представлено схематичное устройство данных индексов.



ID	Val	Data
1	v1	d1
2	v2	d2
3	v3	d3
4	v4	d4
5	v5	d5

Рис. 4. Первичный и вторичный индексы InnoDB

Как и в случае MyISAM первичный индекс построен по полю “ID”, а вторичный по полю “Val”. По рисунку видно, что все данные хранятся в первичном индексе, во вторичном индексе, в листьях находятся значения индекса и первичный ключ. Это удобно для обеспечения согласованности данных. Если строка была перемещена или обновлена, связи между индексами остаются корректными, так как во вторичном индексе хранится первичный ключ.

2. Создание тестовой базы банных

Для демонстрации работы B-tree индекса на MyISAM и InnoDB подготовлена тестовая база данных авиаперевозок с данными, близкими к настоящим [4]. Размер полученной базы дан-

ных составляет 1.7 Гб для MyISAM и 2.4 Гб для InnoDB. Таблица, для которой будут использоваться индексы содержит более 30 миллионов записей.

Особенностью InnoDB при загрузке большого объема данных является то, что он не хранит точное количество строк в таблице. Вместо этого количество строк определяется приблизительно через статистику, собранную движком (рис. 5), в отличие от MyISAM, который поддерживает точное количество строк в таблице в метаданных, что позволяет быстро получить это число без необходимости полного сканирования таблицы (рис. 6).

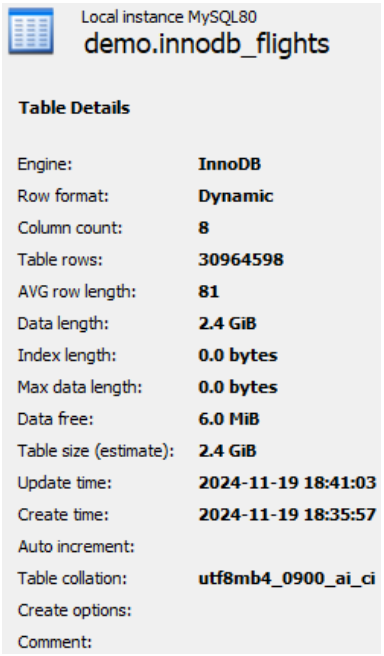


Рис. 5. Анализ таблицы на движке InnoDB

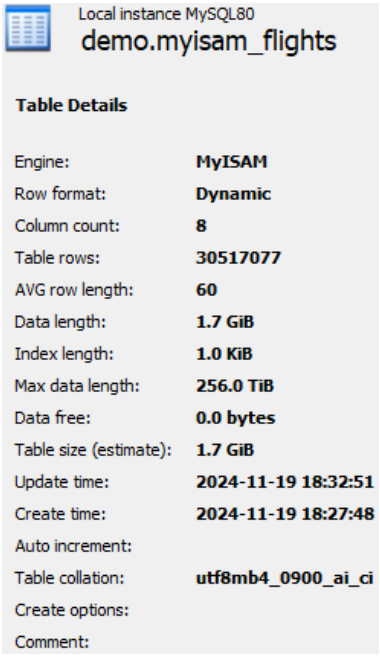


Рис. 6. Анализ таблицы на движке MyISAM

3. Сравнение эффективности B-tree индекса

Для сравнения эффективности скорости выполнения запроса построим индекс на одном и том же поле “passenger_name”. Создание индекса одинаково для обоих движков, например для MyISAM:

```
CREATE INDEX btree_index ON demo.myisam_flights (passenger_name) USING BTREE;
```

Проанализируем SELECT запрос по точному значению, поле которого покрыто индексом:

```
EXPLAIN ANALYZE
SELECT * FROM demo.innodb_flights
WHERE passenger_name = 'VITALIY MATVEEV';
```

Для сравнения эффективности были выбраны несколько показателей: время создания индекса, занимаемое место на диске, скорость выполнения. Результаты представлены в табл. 1.

Таблица 1

Результаты анализа запроса

Параметр	MyISAM	InnoDB
Время создания индекса	300 с	42 с
Занимаемое пространство	183 МБ	740 МБ
Время выполнения SELECT запроса	0.673 с	0.0291 с

InnoDB значительно быстрее MyISAM при создании индексов. Это указывает на более эффективную реализацию построения индексов в InnoDB [5]. MyISAM потребляет гораздо меньше места для хранения индекса, в то время как InnoDB занимает почти в 4 раза больше

памяти. В InnoDB используется кластеризация индексов, где дополнительно хранятся сами данные. InnoDB показывает значительно более быстрое время выполнения запросов, что объясняется оптимизированной обработкой данных благодаря кластерным индексам.

Для проверки скорости выполнения операций записи выполним UPDATE запрос, который затрагивает около 600000 записей по проиндексированному полю, чтобы сравнить скорость обновления:

```
UPDATE demo.myisam_flights
SET passenger_name = 'SERGEY TARASOV'
WHERE passenger_name like 'SERGEY%';
```

Также сравним скорость INSERT запроса. Для теста была создана хранимая процедура, которая в цикле генерирует 100000 записей. Общие результаты для операций записи представлены в табл. 2.

Таблица 2

Результаты анализа запросов операции записи

Параметр	MyISAM	InnoDB
Время выполнения UPDATE запроса	34 с	306 с
Время выполнения INSERT запроса	53 с	106 с

В случае InnoDB время выполнения обоих операций записи оказался больше, это связано с тем, что MyISAM использует более простую структуру индекса и не поддерживает транзакции, поэтому не тратит дополнительное время на запись данных в журнал изменений. Однако это может быть опасно, если операция записи прервется. Также MyISAM при записи блокирует всю таблицу, до тех пор, пока операция не завершится, в то время как InnoDB блокирует только изменяемую строку, позволяя другим транзакциям параллельно обрабатывать другие.

Когда новая запись вставляется в таблицу InnoDB, то движок старается оставить 1/16 страницы свободной для последующих вставок и обновлений индекса. Если записи в индексе вставляются в последовательном порядке, то результирующие страницы индекса заполняюстя примерно на 15/16. Если записи вставляются в случайном порядке, страницы заполняются от 1/2 до 15/16 [6]. Когда нужная страница обнаружена, то сначала происходит вставка записи в кластерный индекс, а затем для вторичных индексов InnoDB также ищет соответствующие узлы и обновляет их, ссылаясь на кластерный индекс.

Если происходит обновление строк с изменением значений в индексированных столбцах, то InnoDB находит строку для обновления через кластерный индекс, удаляет эту запись из индекса, а затем вставляет её аналогично процессу INSERT.

Заключение

По результатам анализа можно сделать вывод, что движок InnoDB является более подходящим для задач, требующих высокой скорости выполнения запросов чтения, благодаря использованию кластерных индексов и оптимизированной обработке данных. Однако это достигается за счет большего объема занимаемого пространства. Также InnoDB обеспечивает большую надежность благодаря поддержке транзакций и скорость операций записи при параллельной обработке, так как блокирует только изменяемые строки.

С другой стороны, MyISAM демонстрирует себя как более экономичный в плане объема хранимых данных, что делает его предпочтительным для сценариев, где скорость индексации не играет ключевой роли, а минимизация места хранения является приоритетом. В том числе обеспечивает более высокую скорость операций записи из-за отсутствия транзакций. Также это исключает возможность взаимоблокировок.

Таким образом, выбор движка следует осуществлять исходя из конкретных требований и ограничений системы.

Литература

1. Отличия движков InnoDB от MyISAM в MySQL. – URL: https://docs.rbsoft.ru/otlichiya-dvizhkov-innodb-ot-myisam-v-mysql. – (Дата обращения: 18.11.2024).

2. Различия индексов MySql. – URL: https://habr.com/ru/articles/556440. – (Дата обращения: 18.11.2024).

3. Документация MySQL. – URL: https://dev.mysql.com/doc. – (Дата обращения: 18.11.2024).

4. Демонстрационная база данных. – URL: https://postgrespro.ru/education/demodb. – (Дата обращения: 18.11.2024).

5. *Шварц Б.* MySQL по максимуму / Б. Шварц, В. Ткаченко, П. Зайцев. – Санкт-Петербург : Питер, 2020. – 864 с. – ISBN 978-5-4461-0696-7.

6. The Physical Structure of an InnoDB Index. – URL: https://dev.mysql.com/doc/refman/8.4/en/innodb-physical-structure.html. – (Дата обращения: 26.11.2024)

УДК 004.77

ИССЛЕДОВАНИЕ ОБОБЩЁННОЙ МОДЕЛИ ПЕРЕДАЧИ ИНФОРМАЦИИ В СЕТИ WI-FI

В. Е. Глушаков

Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина», г. Воронеж

Аннотация. Рассматривается математическая модель оценки времени передачи данных в сети Wi-Fi, позволяющая оценивать время доставки пакета в зависимости от разных параметров, и методика оценки загруженности сети на основе данной модели.
Ключевые слова: время доставки пакетов, беспроводная сеть, пропускная способность.

Введение

В связи с быстрым развитием беспроводных сетей актуальным является оценка и оптимизация времени доставки информации, которое во многом зависит от задержек и помех. В данной статье рассматривается математическая модель, позволяющая оценивать время доставки одного пакета одной станцией в зависимости от различных параметров: размера пакетов, коэффициента неудачной отправки, задержек и т. д., показана невозможность использования среднего времени доставки из-за очень большого разброса временных значений и необходимость разработки нового подхода для оценки загруженности сети, описана новая методика такой оценки.

Методы и принципы исследования

Для построения математической модели были использованы схемы передачи данных для стандарта 802.11, представленные на рис. 1, и описанные в [1, 2].

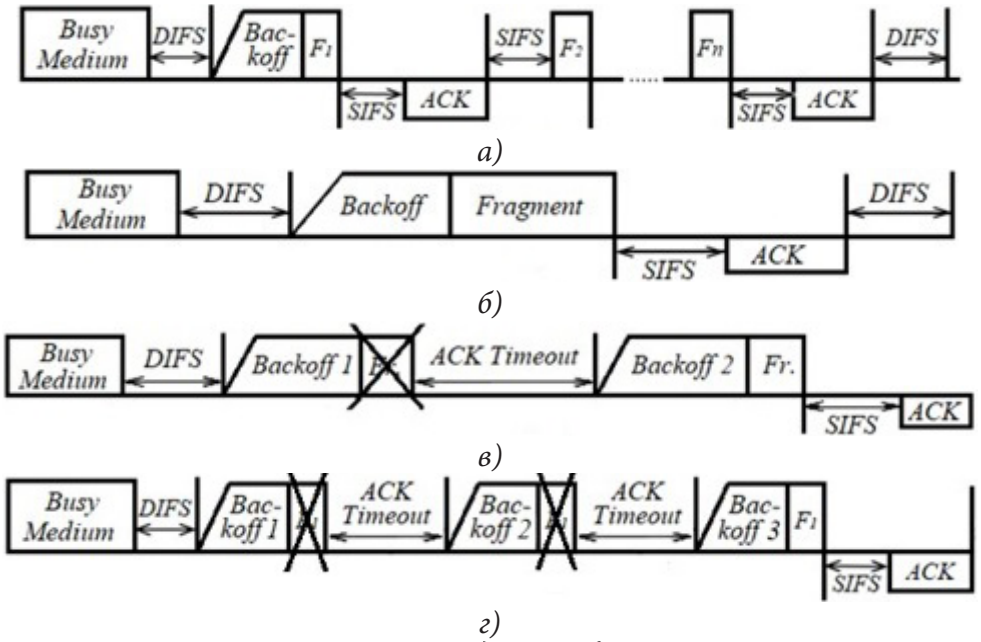


Рис. 1. Основные режимы передачи информации в сети Wi-Fi:
а) ускоренная передача n пакетов в режиме Bursting, б) успешная передача одного пакета, в) неудачная передача пакета с 1-й попытки, г) неудачная передача пакета со 2-й попытки

Математическое моделирование процесса передачи данных

Процесс обмена пакетами будем считать однородным марковским с дискретными состояниями и непрерывным временем. Для нахождения предельных вероятностей состояний системы и закона распределения времени передачи информации, с учётом схемы передачи данных на рис. 1, построим размеченный граф состояний, представленный на рис. 2.

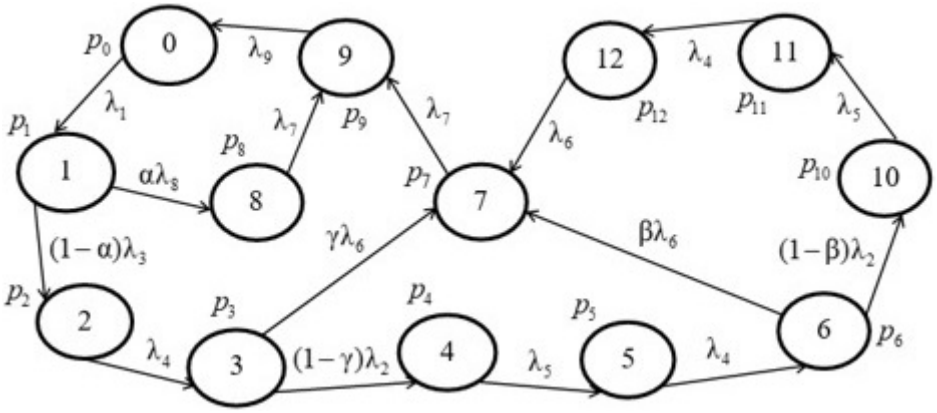


Рис. 2. Граф состояний для передачи информации

Опишем состояния, представленные на рис. 2.

- p_0 — первоначальное состояние (пакеты для отправки отсутствуют);
- p_1 — передающее устройство генерирует пакеты для отправки;
- p_2 — регламентная пауза *DIFS+ Backoff Time_1*;
- p_3, p_6, p_{12} — отправка пакета с задержкой с 1-го, 2-го и 3-го раза соответственно;
- p_4, p_{10} — передающее устройство осуществляет неудачную передачу части пакетов соответственно с 1-го и 2-го раза;
- p_5 — регламентная пауза *ACK_Timeout+ Backoff Time_2*;
- p_7 — регламентная пауза *SIFS*;
- p_8 — отправка пакета с задержкой в режиме Bursting, передающее устройство ждет время *SIFS*;
- p_9 — принимающее устройство передает пакет подтверждения ACK с задержкой (пакет доставлен);
- p_{11} — регламентная пауза *ACK_Timeout+Backoff Time_2*.

Здесь λ_i $(i=1,\ldots,9)$ — интенсивности передачи данных, зависящие от различных параметров функционирования сети; значения α, β и γ выбираются в зависимости от режимов функционирования сети, представленных на рис. 1.

Соответствующая система уравнений Колмогорова примет вид:

$$\begin{cases} \dfrac{dp_0}{dt} = \lambda_9 p_9 - \lambda_1 p_0 \\ \dfrac{dp_1}{dt} = \lambda_1 p_0 - \alpha\lambda_8 p_1 - (1-\alpha)\lambda_3 p_1 \\ \dfrac{dp_2}{dt} = (1-\alpha)\lambda_3 p_1 - \lambda_4 p_2 \\ \dfrac{dp_3}{dt} = \lambda_4 p_2 - \gamma\lambda_6 p_3 - (1-\gamma)\lambda_2 p_3 \\ \dfrac{dp_4}{dt} = (1-\gamma)\lambda_2 p_3 - \lambda_5 p_4 \end{cases}$$

$$\begin{cases} \dfrac{dp_5}{dt} = \lambda_5 p_4 - \lambda_4 p_5 \\ \dfrac{dp_6}{dt} = \lambda_4 p_5 - \beta\lambda_6 p_6 - (1-\beta)\lambda_2 p_6 \\ \dfrac{dp_7}{dt} = \beta\lambda_6 p_6 + \gamma\lambda_6 p_3 + \lambda_6 p_{12} - \lambda_7 p_7 \\ \dfrac{dp_8}{dt} = \alpha\lambda_8 p_1 - \lambda_7 p_8 \\ \dfrac{dp_9}{dt} = \lambda_7 p_7 + \lambda_7 p_8 - \lambda_9 p_9 \\ \dfrac{dp_{10}}{dt} = (1-\beta)\lambda_2 p_6 - \lambda_5 p_{10} \\ \dfrac{dp_{11}}{dt} = \lambda_5 p_{10} - \lambda_4 p_{11} \\ \dfrac{dp_{12}}{dt} = \lambda_4 p_{11} - \lambda_6 p_{12} \end{cases} \quad (1)$$

с начальными условиями

$$\begin{cases} p_0(0) = 1 \\ p_i(0) = 0 \quad (i = 1,\ldots,12). \end{cases} \quad (2)$$

С учётом схем на рис. 1 были рассмотрены следующие режимы работы сети Wi-Fi:

1) ускоренная передача нескольких пакетов $(\alpha = 1)$ — путь 0–1–8–9;

2) успешная передача пакета с 1-й попытки $(\alpha = 0,\ \gamma = 1)$ — путь 0–1–2–3–7–9;

3) успешная передача пакета со 2-й попытки $(\alpha = 0,\ \gamma = 0,\ \beta = 1)$ — путь 0–1–2–3–4–5–6–7–9;

4) успешная передача пакета с третьей попытки $(\alpha = 0,\ \gamma = 0,\ \beta = 0)$ — путь 1–2–3–4–5–6–10–11–12–7–9.

Для численного решения системы (1)–(2) были использованы значения параметров из стандарта IEEE 802.11b/g [3, 4]: $t_{SIFS} = 10$ *мкс*, $t_{SLOT\_TIMER} = 20$ *мкс*, $t_{DIFS} = t_{SIFS} + 2t_{SLOT\_TIMER} = 50$ *мкс*, $t_{ASK\_Timeout} = 10$ *мкс*, размер фрейма ACK — 14 байт. Система решалась приближённо неявным одношаговым методом Розенброка с использованием программного пакета Maple 17.

Анализ результатов

Полученные результаты сравнивались с результатами натурного эксперимента, проведённого в лаборатории факультета ПММ с использованием экспериментальной установки, представленной на рис. 3 и подробно описанной в [5, 6] (клиенты — ноутбуки, серверы — компьютеры).

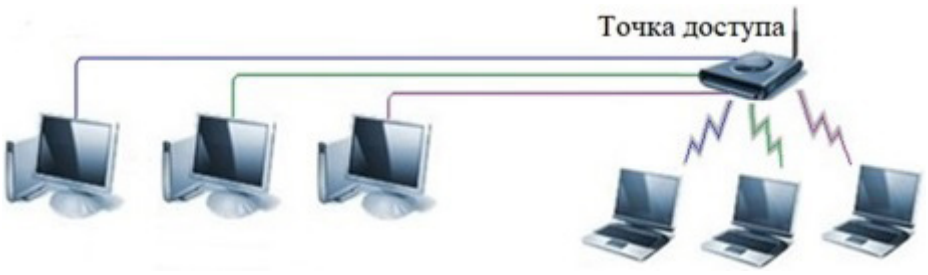


Рис. 3. Схема экспериментальных исследований

В результате проведения пяти измерений для пакетов разного размера с различной скоростью отправки были построены графики (рис. 4), показавшие несостоятельность использования среднего времени для оценки загрузки сети в связи с очень большим разбросом временных значений.

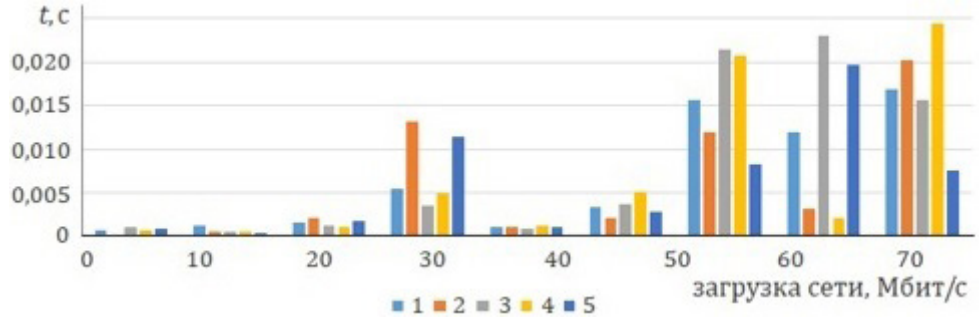


Рис.4. Среднее время доставки пакетов для различных серий экспериментов в зависимости от загруженности сети

В связи с этим в [6] была предложена следующая методика.

Для каждого из описанных выше режимов 1–4 была решена система (1)–(2), построены графики плотностей вероятности времени доставки пакетов, найдены точки их пересечения. Эти точки являются границами диапазонов, на которые разбивается всё время доставки пакета (рис. 5).

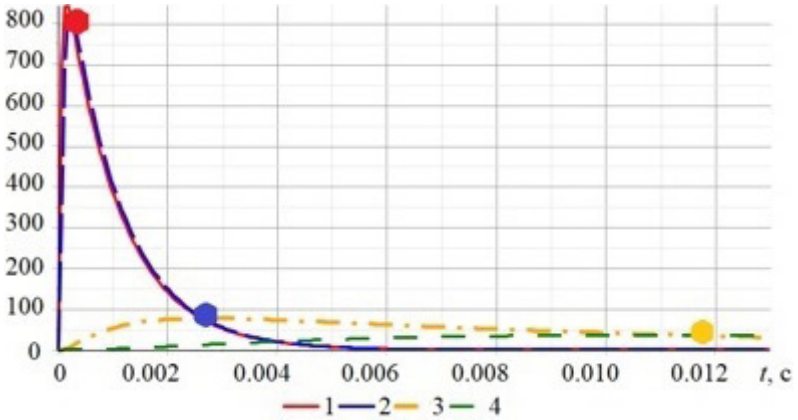


Рис. 5. Плотности вероятностей времени доставки фрагментов для выделенных режимов функционирования информационной системы

Для каждого диапазона найден процент пакетов, в него попадающий, построены линейные аппроксимации (рис. 6), посчитаны величина достоверности аппроксимации (R^2) и коэффициент линейной корреляции.

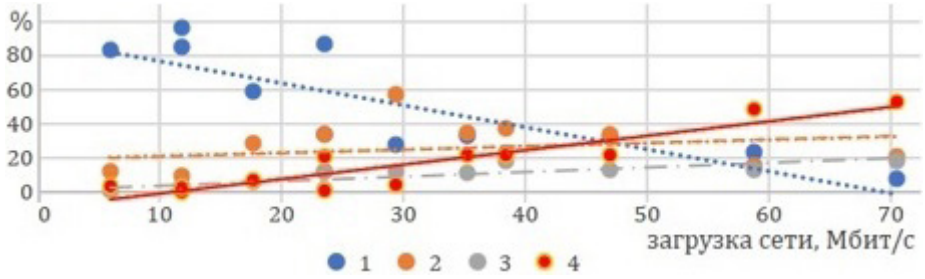


Рис. 6. Процент пакетов, время доставки которых попадает соответственно в каждый из четырех диапазонов, в зависимости от загрузки сети, и линейные аппроксимации данных зависимостей

Из рис. 6 видно, что наиболее предпочтительным является 4-й диапазон.

Для полученных значений натурного эксперимента были построены гистограммы. Экспериментальные данные, представленные на гистограмме, показали хорошее соответствие расчётам по модели (рис. 7).

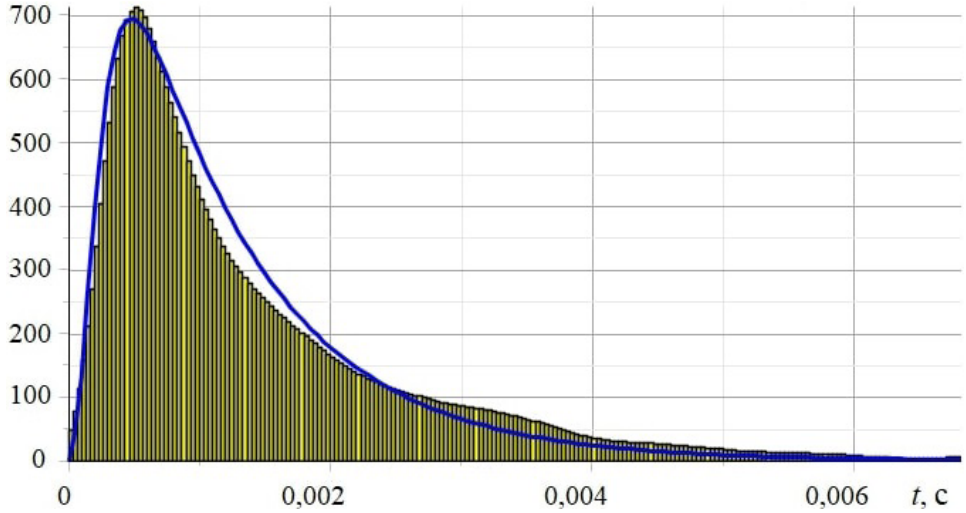


Рис. 7. Гистограмма и плотность вероятностей времени доставки для пакета размером 1400 байт, скорости отправки 100 Мбит/с, 0-й задержке и $\alpha = 0.09, \beta = 1, \gamma = 0.1$

Выводы

Данная математическая модель позволяет оценивать влияние различных параметров на время отправки пакета, использовать методику оценки загруженности канала.

Литература

1. Методы увеличения производительности в беспроводных сетях Wi-Fi, часть первая: Bursting, Compression, Fast Frames, Concatenation. – URL: https://www.ixbt.com/comm/tech-80211g-super_1.shtml (дата обращения: 20.04.2023).

2. Раздел 7. Локальные беспроводные сети WiFi. Лекции по стандартам. – URL: http://docplayer.ru/33818454-Razdel-7-lokalnye-besprovodnye-seti-wifi-lekcii-po-standartam.html (дата обращения 03.03.18г.). – Заглавие с экрана.

3. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band. IEEE Standard 802.11b-1999.

4. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band. IEEE Standard 802.11g-2003.

5. *Абрамов Г. В.* Анализ времени передачи данных в распределенных сетях с конкурирующим доступом / Г. В. Абрамов, А. Е. Емельянов, К. Ч. Колбая // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2016. – № 4. – С. 61–67.

6. *Абрамов Г. В.* Разработка системы мониторинга загруженности гибридных сетей / Г. В. Абрамов, В. Е. Глушаков, Р. В. Данилов // Вестник компьютерных и информационных технологий. – 2024. – № 7. – С. 29–35.

УДК 004.45

РАЗРАБОТКА УПРОЩЕННОЙ ВЕРСИИ ОС LINUX, ИСПОЛЬЗУЕМОЙ В ОБРАЗОВАТЕЛЬНОЙ СРЕДЕ

Р. С. Долгих, М. К. Чернышов

Воронежский государственный университет

Аннотация. В статье рассматриваются основные этапы и принципы создания упрощенной версии операционной системы Linux, предназначенной для использования в образовательных целях. Такого рода операционная система может быть ориентирована на начинающих пользователей, изучающих основы работы с операционными системами (ОС) на примере ОС семейства Linux. Основное внимание уделяется упрощению интерфейса, снижению системных требований и предоставлению обучающего контента, включая документацию, встроенные упражнения и инструменты для изучения базовых команд ОС. Описываются методология разработки, выбор программных компонентов и подходы к интеграции обучающих функций.
Ключевые слова: операционная система Linux, образовательные технологии, ядро операционной системы, интерфейс пользователя, легковесные системы.

Введение

Благодаря гибкости, надежности и открытой модели разработки, Linux является одной из наиболее популярных операционных систем в мире. Однако для начинающих пользователей и студентов, только начинающих изучать основы работы с операционными системами, Linux может казаться сложной и неподъемной операционной системой из-за большого количества настроек, команд и разнообразия дистрибутивов.

Целью задачи, рассматриваемой в данной работе, является создание упрощенного дистрибутива Linux, который будет служить образовательной платформой для обучения основам работы с данной операционной системой. Такой дистрибутив должен предоставлять доступный интерфейс, минимальный набор предустановленных программ, встроенные обучающие модули и руководство, что позволит учащимся сосредоточиться на изучении ключевых концепций.

Работа направлена на разработку инструмента, способного снизить барьер входа для пользователей, заинтересованных в освоении Linux, а также повысить интерес к этой операционной системе как к объекту исследования и практического применения.

1. Проектирование образовательной операционной системы

В процессе проектирования операционной системы необходимо учесть несколько ключевых аспектов:

• Упрощение интерфейса: интерфейс должен быть интуитивно понятным для начинающих пользователей. Это важно, чтобы студенты и школьники могли сосредоточиться на изучении принципов работы с ОС, а не тратить время на освоение сложных интерфейсов.

• Интерактивное обучение: система должна предлагать задания, которые помогают учащимся освоить основные команды Linux, такие как работа с файловой системой, процессами, сетью и правами доступа. Элементы геймификации, например, выполненные задания с автоматической проверкой, могут стимулировать обучение.

• Минимизация системных требований: операционная система должна быть достаточно легковесной, чтобы работать на старых и маломощных устройствах, которые могут быть до-

ступны в образовательных учреждениях. Это также позволяет использовать систему на виртуальных машинах, что облегчает массовое развертывание.

• Встроенные образовательные материалы: важным элементом является наличие встроенных инструкций и учебных пособий, доступных прямо из системы. Эти материалы могут включать как текстовые, так и видеоуроки, примеры заданий и пошаговые инструкции.

Целевая аудитория для этой системы — это в первую очередь новички в мире Linux, среди которых можно выделить несколько категорий:

• Школьники и студенты: ученики старших классов и студенты, которые начинают знакомиться с операционными системами и командной строкой. Часто они не имеют глубоких знаний в области ИТ и нуждаются в системе, которая будет не только простой, но и доступной для учебных целей.

• Преподаватели учебных заведенияй для учителей и преподавателей учебных заведений эта ОС должна стать инструментом, который легко разворачивается на компьютерах и предоставляет учащимся готовые для обучения материалы и задания. Преподаватели могут использовать систему для проведения лекций, лабораторных работ и семинаров.

• Самообучающиеся пользователи: те, кто хочет изучить Linux, но не имеет доступа к специализированным учебным курсам, могут использовать эту систему для самостоятельного изучения с пошаговыми инструкциями.

Выбор разработки образовательной ОС с нуля, начиная с ядра Linux, имеет несколько важных преимуществ:

• Полный контроль над системой: при создании ОС с нуля, разработчик имеет полный контроль над всеми компонентами, начиная от выбора пакетов и настройки ядра до оптимизации пользовательского интерфейса. Это позволяет исключить лишние и ненужные компоненты, минимизируя систему и делая её максимально легкой и удобной для обучающихся.

• Гибкость в обучении: ОС, разработанная с нуля, может быть адаптирована под специфические образовательные цели. Например, можно интегрировать в систему функции, ориентированные на определенную дисциплину, такие как командный интерфейс, скрипты и другие технологии, применимые в курсе Linux.

• Инновации и уникальность: в отличие от использования существующих дистрибутивов, создание с нуля позволяет внедрять новые методы обучения, такие как интерактивные модули и поддержка нестандартных форматов материалов. Это делает ОС уникальной и адаптированной под нужды конкретной аудитории.

• Масштабируемость и оптимизация: при разработке с нуля можно настраивать систему для работы даже на самых старых и слабых компьютерах, что идеально подходит для учебных заведений с ограниченными ресурсами. При этом можно исключить ненужные сервисы и приложения, обеспечив оптимальную производительность системы.

Основным результатом разработки ОС с нуля является создание системы, которая будет не только облегчать процесс обучения основам Linux, но и мотивировать пользователей к дальнейшему ее изучению. Ожидаемые результаты включают в себя:

• Увеличение вовлеченности учащихся: наличие интерактивных уроков и задач, которые активно вовлекают пользователей в процесс обучения, повышает уровень вовлеченности и интерес к Linux.

• Повышение доступности образования: возможность использовать систему на старых или менее мощных устройствах позволяет сделать обучение доступным для широкой аудитории, особенно в странах с ограниченными ресурсами.

• Рост навыков в области Linux: предоставление учащимся практических навыков работы с системой и командной строкой помогает им развивать важные IT-компетенции, которые могут быть полезны как в учебе, так и в дальнейшей карьере.

2. Сборка ядра и базовой системы

Сборка ядра и создание базовой системы — важный этап разработки образовательной операционной системы на базе Linux. На этом этапе формируется основа, на которой будет построена вся дальнейшая функциональность, включая пользовательский интерфейс, образовательные материалы и дополнительные инструменты. Основные задачи этого этапа включают настройку ядра, создание минимальной операционной системы, а также повышение суммарной производительности.

Для начала необходимо выбрать подходящее ядро Linux. Это наиболее важно, поскольку версия ядра будет определять совместимость с различным оборудованием, производительность системы и доступность нужных драйверов. Для образовательной ОС желательно использовать стабильную версию ядра, которая поддерживает как современные, так и старые устройства. Важно, чтобы ядро обеспечивало все необходимые функциональные возможности для базовой работы системы, но при этом не включало лишних модулей. Например, если система не будет работать с определенными устройствами, их драйверы можно исключить на этапе сборки.

Настройка ядра Linux требует внимательного подхода к выбору и оптимизации различных параметров. Важно отключить все ненужные модули и драйверы, чтобы не перегружать систему лишними процессами, что особенно важно для слабых и устаревших устройств. Кроме того, настройка ядра должна обеспечивать стабильную работу всех критически важных функций системы, таких как файловая система, сетевые соединения и базовые устройства ввода/вывода.

После сборки ядра следует разработать минимальный набор компонентов, входящих в состав операционной системы. В первую очередь необходимо выбрать и настроить систему инициализации системы — компонента, который управляет запуском всех процессов при старте системы. Для легковесной образовательной ОС рекомендуется использовать более простые и быстрые решения, такие как **OpenRC** или **runit**, которые потребляют меньше ресурсов по сравнению с более тяжелыми альтернативами, такими как **systemd**.

Затем необходимо собрать минимальный набор утилит, составляющих основу будущей операционной системы. Данный набор может включать такие программы, как **busybox** (программу, которая объединяет множество утилит в одном файле) и **coreutils** (утилиту, предназначенную для работы с файлами и каталогами). Эти утилиты обеспечат базовую функциональность для взаимодействия с системой, позволяя пользователю выполнять основные операции, такие как создание и копирование файлов, управление процессами и другие действия, необходимые для работы с Linux.

Также на этом этапе важно создать базовую файловую структуру, которая будет включать все необходимые каталоги, такие как, в частности, **/bin** (для утилит), **/etc** (для конфигурационных файлов), **/home** (для домашних каталогов пользователей). Эта структура должна быть простой и понятной, чтобы пользователи могли легко ориентироваться в системе.

Подключение драйверов и настройка сетевых утилит также являются важными этапами. Для минималистичной образовательной системы важно оставить только те драйверы и утилиты, которые реально необходимы для базовой работы, чтобы система оставалась легковесной и не перегружалась лишними компонентами.

Когда базовый набор компонент операционной системы будет установлен и настроен, следующим шагом будет создание минимального установщика, который позволит развернуть операционную систему на целевых устройствах. Установщик должен быть максимально простым и автоматизированным, чтобы пользователи могли без проблем установить систему на компьютерах. Важно, чтобы установщик автоматически определял настройки устройства и минимизировал количество действий, требующих вмешательства пользователя.

После сборки и настройки базовой системы необходимо провести тестирование ее работоспособности. Оно включает в себя проверку функциональности всех основных утилит и компонентов системы, а также тестирование уровня ее производительности для того, чтобы убедиться, что система работает эффективно, даже на слабых устройствах. Тестирование на реальных устройствах поможет выявить возможные проблемы и недостатки, которые могут возникнуть при использовании системы в учебных заведениях.

3. Интеграция пользовательского интерфейса и образовательного контента

На данном этапе разработки образовательной операционной системы основное внимание уделяется созданию удобного и интуитивно понятного интерфейса, а также интеграции образовательных материалов и инструментов. Система должна быть не только функциональной, но и доступной для новичков, что предполагает особое внимание к пользовательскому интерфейсу и обучающим компонентам. Важной задачей является обеспечение доступности и легкости восприятия интерфейса пользователем. При этом сам процесс работы с операционной системой не должен вызывать трудностей.

Прежде всего необходимо определить, будет ли система использовать графический интерфейс или ограничится консольным режимом работы. Для образовательной ОС на базе Linux предпочтительным вариантом является легковесный графический интерфейс, поскольку это значительно упрощает взаимодействие с системой, особенно для новичков. В качестве оконного менеджера можно использовать такие решения, как **Openbox**, **Fluxbox** или **i3**, которые требуют минимальных системных ресурсов и могут быть настроены для упрощенной навигации. Эти менеджеры окон позволяют создать чистый, ненагруженный интерфейс с необходимыми функциями и большими возможностями для кастомизации.

Кроме того, можно рассмотреть вариант текстового режима с простыми подсказками, который будет хорош для обучения командной строке. Такая система будет включать только базовые текстовые инструменты и редакторы, такие как **nano** или **vim**, что позволит пользователям сосредоточиться на изучении команд и основ работы с терминалом.

Важной частью этого этапа является создание удобной навигации по системе. Интерфейс должен быть простым, с минимальным количеством окон и кнопок, чтобы новички могли быстро ориентироваться в том, что необходимо для выполнения заданий. Меню и панели инструментов должны быть максимально упрощены. Каждый элемент интерфейса должен быть выполнен в строгом соответствии с принципами удобства и простоты. К примеру, можно добавить визуальные подсказки или всплывающие окна с краткими инструкциями по выполнению задач.

Образовательный контент — основная ценность операционной системы, и его интеграция в систему требует тщательного подхода. Система должна включать базовые учебные материалы, такие как пошаговые инструкции по работе с командной строкой, примеры использования команд Linux, а также упражнения и задания для практического освоения. Можно интегрировать интерактивные курсы, которые помогут пользователям освоить основные концепции Linux в игровой форме, например, с выполнением задач и получением обратной связи о правильности выполнения.

Для этого можно использовать сценарии и интерактивные уроки, которые будут встроены в систему. Каждый пользователь при запуске системы мог бы пройти начальное ознакомление с интерфейсом, а затем постепенно переходить к более сложным заданиям, которые проверяют усвоение материала. Важно, чтобы уроки были интерактивными, с элементами обратной связи. Например, задания могут проверять, выполнил ли пользователь команду правильно, а в случае ошибки предлагать подсказки или рекомендации.

Не менее важным является создание встроенной справочной системы, которая должна быть всегда доступна, предоставляя пользователям справочную информацию по ключевым

аспектам системы. Это могут быть локальные мануалы для базовых команд Linux, примеры использования утилит, а также объяснения по базовым принципам работы с операционной системой. Для новичков важно, чтобы эти материалы были четкими и краткими, сфокусированными на практическое использование, а не на теоретические концепции. Например, можно интегрировать man-страницы с простыми примерами команд, которые пользователь может немедленно протестировать в терминале.

Для улучшения понимания можно добавить визуальные элементы, такие как схемы или инфографика, объясняющие, как работает файловая система Linux, что такое процессы и как управлять ими с помощью команд. Эти элементы могут быть встроены прямо в систему или представлены как отдельные файлы, которые могут быть открыты пользователем по запросу.

Помимо текстовых и графических материалов, полезным дополнением будут видеоуроки и демонстрации, которые могут быть интегрированы в систему как локальные файлы. Это особенно важно для визуалов, которые воспринимают информацию через демонстрации и примеры. Например, видео может показывать, как установить программы через пакетный менеджер, или объяснять, как работать с текстовыми редакторами.

Таким образом, система должна включать не только основную функциональность операционной системы, но и компоненты, направленные на развитие навыков пользователей. Главное здесь — сделать обучение доступным, интересным и максимально вовлекающим. Интерактивность и простота интерфейса помогут новичкам уверенно осваивать Linux и использовать операционную систему для решения практических задач, а интеграция образовательных материалов сделает процесс обучения эффективным и увлекательным.

4. Тестирование, оптимизация и внедрение

Тестирование, оптимизация и внедрение являются заключительными этапами разработки образовательной операционной системы. Эти процессы направлены на обеспечение стабильной работы системы, улучшение её производительности и подготовку к запуску на реальных устройствах.

Тестирование начинается с функциональных проверок всех ключевых компонентов системы, включая загрузку, установку программ, работу с файлами и каталогами. Особое внимание следует уделить проверке интерфейса пользователя, чтобы он был интуитивно понятным для новичков. Важно протестировать систему на различных устройствах и в условиях ограниченных ресурсов, чтобы выявить возможные проблемы совместимости. Также необходимо провести тестирование безопасности, чтобы гарантировать защиту данных пользователей от возможных угроз.

Внедрение системы в образовательные учреждения должно быть максимально простым и доступным. Для этого разрабатывается автоматизированный установщик, который легко устанавливает ОС на различные устройства. Важно обеспечить доступ к подробной документации и инструкциям, а также создать систему поддержки пользователей, чтобы они могли обращаться за помощью в случае возникновения проблем. После внедрения нужно продолжать собирать отзывы и обновлять систему, чтобы она оставалась актуальной и безопасной.

Тестирование, оптимизация и внедрение являются ключевыми этапами, которые обеспечивают стабильность, безопасность и удобство использования операционной системы в образовательной среде.

Заключение

Разработка упрощенной операционной системы Linux для образовательных целей включает в себя несколько ключевых этапов: постановку целей, проектирование, сборку ядра, ин-

теграцию интерфейса и тестирование системы. Каждый из этих этапов ориентирован на создание стабильной, производительной и безопасной системы, которая легко воспринимается пользователями с разным уровнем подготовки.

Создание такой системы должно способствовать эффективному обучению пользователей современным технологиям.

Литература

1. *Матвеев М. Д.* Ядро Linux. Сборка, настройка, управление. – М. : Вертекс, 2023. – 352 с.

2. *Лав Р.* Операционные системы: описание процесса разработки ядра Linux. –М. : Диалектика, 2018. – 416 с.

3. *Таненбаум А. С., Бос Х.* Современные операционные системы. 4-е изд. – М. : Вильямс, 2014. – 976 с.

4. *Керрис М.* Программирование в Linux. – СПб. : Питер, 2010. – 736 с.

5. *Стадник С. В.* Администрирование Linux. Подробное руководство. – М. : БХВ-Петербург, 2017. – 560 с.

6. «Linux From Scratch». Linux From Scratch – Руководство по сборке системы с нуля. – URL: https://www.linuxfromscratch.org/ (дата обращения: 26.11.2024).

УДК 004.056.5

ПРОЦЕССНАЯ МОДЕЛЬ ОРГАНИЗАЦИИ МЕЖОБЪЕКТНОГО ВЗАИМОДЕЙСТВИЯ НА УРОВНЕ СУБЪЕКТА КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

С. С. Долженков

МИРЭА — Российский технологический университет

Аннотация. Исследование посвящено актуальным вопросам по взаимодействию субъектов критической информационной инфраструктуры посредством принадлежащих им объектов критической информационной инфраструктуры — межобъектное взаимодействие на уровне субъектов. Автор отмечает важность развития защитных мероприятий по предотвращению инцидентов в области информационной безопасности на субъектах критической информационной инфраструктуры, уделяет внимание потенциальным угрозам от образовавшихся уязвимостей на уровне программного обеспечения субъектов критической информационной инфраструктуры.
Ключевые слова: информационные процессы, информационные потоки, информационная безопасность, критическая информационная инфраструктура, субъект критической информационной инфраструктуры, объект критической информационной инфраструктуры, программное обеспечение, уязвимость, инцидент.

Введение

Основа любого успешно развивающегося государства в период цифровой трансформации 21 века — это грамотное, хорошо спланированное и прогнозируемое движение в сторону повсеместного внедрения информационных технологий: во все сферы жизни общества с целью автоматизации повседневных и емких задач, для упрощения аналитики протекающих процессов и сбора данных, а также для централизации контроля разного рода и уровня. Однако в условиях ограниченности ресурсов постоянно будет выбор их рационального использования. Сфера информационных технологий исключением не является: каким отраслям автоматизации жизни общества отдать большее предпочтение остается открытым вопросом ежедневно, но существуют особые направления развития информационных технологий, которым необходимо особое и отдельное внимание, например, критическая информационная инфраструктура, которой и будет посвящено данное исследование.

Критическая информационная инфраструктура (КИИ) — информационные системы, информационно-телекоммуникационные сети, автоматизированные системы управления субъектов КИИ, а также сети электросвязи, используемые для организации взаимодействия таких объектов. Из определения КИИ следует вывод, что КИИ состоит из одного или нескольких субъектов КИИ (СКИИ), которым принадлежат объекты КИИ, работающие в отраслях, признанных критической информационной инфраструктурой. Примером КИИ может служить юридическое лицо, занятое в сфере здравоохранения, которое использует в своей деятельности медицинские информационные системы медицинских организаций, где СКИИ будет служить сама медицинская компания, а объектами КИИ являться ее информационные системы, информационные-телекоммуникационные сети и автоматизированные системы управления медицинским оборудованием [1].

Государству крайне важно прилагать достаточное количество усилий в плане контроля и безопасности сферы КИИ, для устойчивой и безотказной работы СКИИ. Во многом от состояния КИИ зависит безопасность и стабильность государства: в случае значительных нарушений и проблем в сфере КИИ последствия могут затронуть все сферы жизни общества.

1. Межобъектное взаимодействие на уровне субъекта критической информационной инфраструктуры

Исследуя безопасность СКИИ, следует принимать во внимание организацию межобъектного взаимодействия с целью сохранения целостности объекта защиты, а также стремление к эффективному выполнению всех процессов путем межобъектного взаимодействия на уровне СКИИ.

Для организации и исследования процессной модели и межобъектного взаимодействия на уровне СКИИ обратимся к определению и требованиям разрабатываемых моделей: модель — физический или абстрактный объект, адекватно отражающий исследуемую систему.

Ко всем разрабатываемым моделям предъявляются два противоречивых требования:

1. Простота модели. Требование простоты модели обусловлено необходимостью построения модели, которая может быть рассчитана доступными методами.

Степень сложности (простоты) модели определяется уровнем ее детализации, зависящим от принятых предположений и допущений.

2. Адекватность исследуемой системы. Соответствие моделей оригиналу, характеризуемое степенью близости свойств модели свойствам исследуемой системы [2].

Межобъектное взаимодействие на уровне СКИИ представляет собой совокупность взаимосвязанных информационных систем — комплексов. Подобные системы, в свою очередь, являются совокупностью взаимосвязанных элементов, объединенных в одно целое для достижения некоторой цели, определяемой назначением системы. В таких условиях минимально неделимым объектом, рассматриваемым как единое целое будет приниматься элемент [3].

Проецируя вышеописанное на межобъектное взаимодействие на уровне СКИИ, объект — это комплекс, которым является информационная система (ИС), составом информационной системы служат компоненты ИС (К), состоящие из минимально неделимой величины – элементов (Э), которые обеспечивают нормальное функционирование процессов обратимся к рис. 1 для описания взаимодействия, выразив межобъектное взаимодействие на уровне СКИИ.

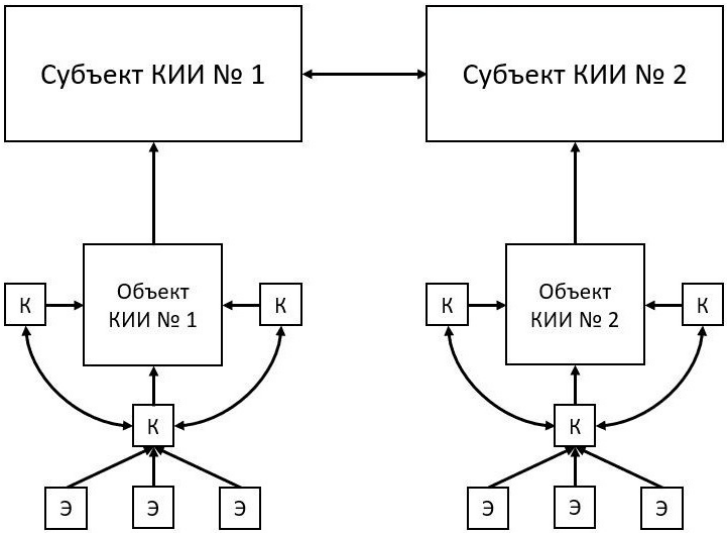


Рис. 1. Межобъектное взаимодействие на уровне СКИИ

Процессная модель организации межобъектного взаимодействия на уровне СКИИ также может быть представлена на основе системы массового обслуживания, но с учетом возможных уязвимостей в программном обеспечении объектов КИИ — рис. 2 [4].

Система массового обслуживания (СМО) — математический (абстрактный) объект, содержащий один или несколько приборов П (каналов), обслуживающих заявки, поступающие

в систему, и накопитель Н, в котором находятся заявки, образующие очередь и ожидающие обслуживания.

Подробное представление работы СМО описаны в трудах [2].

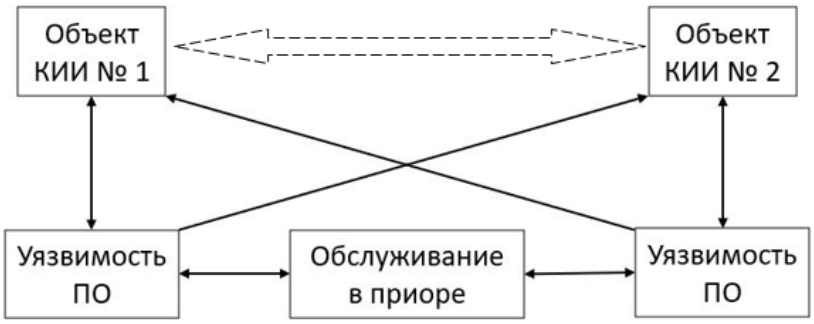


Рис. 2. Межобъектное взаимодействие объектов КИИ в контексте теории массового обслуживания

Таким образом, процессную модель организации межобъектного взаимодействия на уровне СКИИ с учётом возможных уязвимостей можно охарактеризовать следующим образом:

Информационная заявка, в виде информационного сообщения или информационного запроса, должна быть доставлена от объекта КИИ № 1 к объекту КИИ № 2 с последующим обратным взаимодействием: от объекта КИИ № 2 к объекту КИИ № 1 посредством обслуживания в приборе — серверу. С учетом потенциальных уязвимостей на объектах КИИ на уровне программного обеспечения (ПО), необходимо учитывать риск искажения информации, для всех поступающих заявок в систему [5].

Исходящая заявка от объекта КИИ № 1 к объекту КИИ № 2 потенциально может быть искажена уязвимостью в ПО на этапе до обслуживания в приборе и на этапе после обслуживания в приборе. В момент попадания заявки в информационную инфраструктуру объекта КИИ № 2, которая потенциально аналогично объекту КИИ № 1 также подвержена уязвимостям на уровне ПО. Следовательно, заявка, попадающая в объект КИИ № 2, дополнительно может подвергаться искажению своего исходного содержания [6].

Заключение

Таким образом, при рассмотрении вопросов организации межобъектного взаимодействия, дополнительно требуется исследование воздействий уязвимостей объектов КИИ на их взаимодействие и деятельность самого объекта КИИ, подвергшегося воздействию уязвимости другого объекта КИИ, а также изучение потенциальных возможностей от взаимодействия уникальных уязвимостей от двух объектов КИИ: их слияние, нарастающую угрозу и влияние на работу систем и выдаваемых ими результатов.

Литература

1. Федеральный закон от 26.07.2017 г. № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации»

2. *Алиев Т. И.* Основы моделирования дискретных систем / Т. И. Алиев; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики. – Санкт-Петербург : Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2009. – 363 с. – EDN WCWMBZ.

3. *Долженков С. С.* Оптимизация системы менеджмента информационной безопасности объектов критической информационной инфраструктуры / С. С. Долженков // Студенческая

наука для развития информационного общества : Материалы XУ Всероссийской научно-технической конференции с приглашением зарубежных ученых, Ставрополь, 28 ноября 2023 г. – Ставрополь: Северо-Кавказский федеральный университет, 2024. – С. 124–130. – EDN EYGAVD.

4. *Максимова Е. А.* Межсубъектное взаимодействие как источник деструктивных воздействий на субъекте критической информационной инфраструктуры / Е. А. Максимова, Н. П. Садовникова // Прикаспийский журнал: управление и высокие технологии. – 2021. – № 2(54). – С. 71–80. – DOI 10.21672/2074-1707.2021.53.1.071-0800. – EDN MJJRWJ.

5. *Гончаренко М. Д.* Безопасность критической информационной инфраструктуры / М. Д. Гончаренко, К. С. Холкина // Научное и техническое творчество молодежи : Материалы Всероссийской научно-практической конференции с международным участием, Новосибирск, 19–20 апреля 2023 года / Под редакцией А.В. Ефимова, Т.И. Монастырской. Том Часть 5 (дополнительная). – Новосибирскn: Сибирский государственный университет телекоммуникаций и информатики, 2023. – С. 200–208. – EDN JGPDIW.

6. *Ерохин С. Д.* Многоуровневая политика безопасности сети передачи данных / С. Д. Ерохин, П. Л. Пилюгин // DSPA: Вопросы применения цифровой обработки сигналов. – 2022. – Т. 12, № 2. – С. 13-22. – EDN RBLNMD.

УДК 510.52, 004.423.25

О ЗАДАЧЕ ТОТАЛЬНОЙ ВЫВОДИМОСТИ ПУСТОГО СЛОВА В КОНТЕКСТНО-СВОБОДНОЙ ГРАММАТИКЕ

С. М. Дудаков

Тверской государственный университет,
НИУ Высшая школа экономики

Аннотация. Мы изучаем задачу тотальной выводимости пустого слова в контекстно-свободной грамматике. Б. Н. Карловым показано, что если тотальность задавать произвольным множеством правил, то такая задача будет *NP*-полной. Наш же результат показывает, что при классическом понимании тотального вывода, то есть применении всех правил грамматики, ответ может быть получен за полиномиальное время. Для этого мы предлагаем алгоритм, который по *КС*-грамматике строит мультиграф, затем — его конденсацию, а дальше проверяет некоторые свойства путей в этой конденсации, что можно сделать классическими алгоритмами за полиномиальное время.
Ключевые слова: контекстно-свободная грамматика, нетерминал, тотальная выводимость, пустое слово, полиномиальный алгоритм, мультиграф, компонента сильной связности, конденсация графа, путь.

Введение

Мы рассматриваем задачу тотальной выводимости пустого слова в контекстно-свободной грамматике.

Вывод в грамматике называется тотальным, если каждое правило грамматики применяется в нём хотя бы один раз. Это понятие было введено в работе [1] и использовалось для построения множества аксиом для некоторого варианта исчисления Ламбека.

Ранее в [1] было показано, что в общем случае задача тотальной выводимости для произвольной грамматики и произвольного слова является NP-полной.

Недавно Б. Н. Карловым было доказано, что NP-полным является следующий вариант задачи: для произвольной контекстно-свободной грамматики $G = (\sum, N, P, S)$ и под(мульти) множества $R \subseteq P$ правил вывода выяснить, выводимо ли пустое слово так, что каждое правило из R применяется хотя бы один раз. Такой вариант задачи называется R-тотальной выводимостью.

С. Л. Кузнецовым был поставлен вопрос о том, можно ли ослабить условие последней задачи до просто тотальной выводимости, то есть можно ли для NP-полноты задачи требовать, чтобы каждое правило P использовалось при выводе пустого слова не менее одного раза.

Мы отвечаем на этот вопрос и показываем, что ответ является отрицательным (в предположении $PTIME \neq NP$, разумеется). Для этого мы в явном виде приводим алгоритм, требующий полиномиального времени и позволяющий ответить на вопрос о тотальной выводимости пустого слова.

Заметим, что задача тотальной выводимости пустого слова может иметь и вполне значимую практическую интерпретацию. Будем рассматривать нетерминалы как события (сигналы, триггеры, реагирующие на эти события/сигналы), а каждое правило грамматики — как возможные реакции на события. Иными словами, правило $A \to B_1 B_2 \ldots B_k$ можно рассматривать как указание того, что в ответ на событие A будут последовательно возбуждены события $B_1, B_2, \ldots, B_k$. Тогда вопрос о тотальной выводимости пустого слова эквивалентен вопросу о том, может ли начальное событие последовательно привести в действие все возможные реакции.

1. Основные определения

Напомним, что **контекстно-свободная грамматика** (далее — ***KC*-грамматика**) — это четвёрка вида $G = (\sum, N, P, S)$, в которой $\sum$ — множество **терминалов**, N — множество **нетерминалов**, P — множество **правил вывода**, а $S \in N$ — **начальный нетерминал**. Каждое правило вывода имеет вид $A \to x_1 \ldots x_n$, где $A \in N$, $x_1 \ldots x_n \in \sum \cup N$. **Вывод** слова w в KC-грамматике G — это последовательность слов $(w_0, \ldots, w_k)$ в алфавите $\sum \cup N$, в которой $w_0 = S$, $w_k = w$, и для каждого $i = 1, \ldots, n$ слово w_i получено из слова w_{i-1} применением одного из правил вывода. Иначе говоря, существуют слова α и β такие, что $w_i = \alpha u \beta, w_{i-1} = \alpha A \beta$, и в P есть правило $A \to u$. Вывод $(w_0, \ldots, w_k)$ называется **тотальным**, а слово w — **тотально выводимым**, если в выводе каждое правило P применяется хотя бы один раз.

Прежде всего сделаем несколько замечаний. Очевидно, что для задачи тотальной выводимости пустого слова множество $\sum$ терминальных символов грамматики, можно считать, является пустым, так как при наличии хотя бы одного правила $A \to u$, в котором слово u содержит хотя бы один терминал никакой тотальный вывод пустого слова не породит. Поэтому даже если терминалы формально и входят в состав грамматики, то ни в одном правиле они встречаться не могут.

Далее, мы можем считать, грамматика не содержит **бесполезных** нетерминалов: пустых и недостижимых. Напомним, что **пустым** называется нетерминал, из которого нельзя вывести никакого терминального слова, в нашем случае — пустого. Если нетерминал A является пустым и имеется правило $A \to u$, то оно не может быть использовано в тотальном выводе, так как для его использования необходимо получить A, а после этого пустого слова мы не получим.

Недостижимым называется нетерминал, который нельзя получить из начального нетерминала A. И снова, если в грамматике есть правило $A \to u$ для недостижимого нетерминала A, то оно никогда не может быть применено, поскольку сам A не может быть получен.

Итак, имеется грамматика $G = (\sum, N, P, S)$, не содержащая терминальных символов и бесполезных нетерминалов. Построим по этой грамматике размеченный ориентированный мультиграф M_G следующим образом. Напомним, что **мультиграф**, в отличие от просто графа, позволяет иметь несколько рёбер из вершины V в вершину U. Такие рёбра называются **кратными**.

Вершины мультиграфа M_G будут помечены нетерминалами из N или пустыми словами ε. При этом для каждого нетерминала A из N должна быть в точности одна вершина, помеченная A, а для каждого пустого правила $A \to \varepsilon$ из P имеется «своя» вершина, помеченная ε, будем её обозначать ε_A.

Если в P есть правило $A \to B_1 B_2 \ldots B_k$, то имеются рёбра из вершины A в вершины $B_1, B_2, \ldots, B_k$, где $B_1, B_2, \ldots, B_k$ — нетерминалы. Для правила $A \to \varepsilon$ из A имеется ребро в вершину ε_A. В любом случае рёбра помечены тем правилом, с помощью которого они построены. Если среди нетерминалов $B_1, B_2, \ldots, B_k$ есть одинаковые, то в мультиграфе M_G будет соответствующее количество кратных рёбер.

Напомним, что **компонентой сильной связности** (КСС) в (мульти)графе называется максимальное множество попарно взаимно достижимых вершин, то есть таких, которые входят в один и тот же цикл. КСС являются классами эквивалентности на множестве вершин по отношению взаимной достижимости, поэтому КСС разбивают всё множество вершин на непересекающиеся подмножества. В дальнейшем с помощью A^* для вершины A обозначаем (единственную) КСС, которой она принадлежит.

Далее, для (мульти)графа M можно построить новый граф M^* — **конденсацию**. Её вершинами являются КСС старого (мульти)графа M, а каждое ребро (мульти)графа M с меткой D, ведущее из вершины A в вершину B, индуцирует ребро с такой же меткой в M^*, ведущее из A^* в B^*, при условии, что A и B принадлежат разным КСС. Петли, которые возникли бы

в случае $A^* = B^*$, мы в конденсацию не включаем. Таким образом, конденсация M^* — это ациклический граф. Заметим, что поскольку мы рассматриваем именно мультиграфы, то у нас не возникнет проблема с кратными рёбрами, которые будет содержать конденсация.

Итак, для построенного по KC-грамматике G мультиграфа M_G найдём КСС и построим конденсацию M_G^*. Введём для этой конденсации M_G некоторую терминологию. Если в графе M_G из A исходят не менее двух рёбер с одним и тем же правилом, которые снова ведут в вершины КСС A^*, то будем называть такую КСС A^* **мультипликативной**. Если в графе M_G из A исходят не менее двух рёбер с одним и тем же правилом, хотя бы одно из которых снова ведёт в вершину КСС A^*, а другое — в вершину B из другой КСС, то будем говорить, что соответствующее ребро конденсации, ведущее из A^* в B^*, является **мультипликативным**. Остальные КСС и рёбра конденсации называем **цепными**.

2. Главный результат

Критерий, который мы предлагаем для тотальной выводимости пустого слова, заключается в следующем.

Теорема. *В KC-грамматике G тотально выводимо пустое слово тогда и только тогда, когда выполнено следующее: в конденсации M_G^* для любой цепной КСС K, из которой выходят хотя бы два цепных ребра, помеченных разными правилами, существует путь из начальной КСС S^* в K, проходящий хотя бы через одну мультипликативную КСС L или хотя бы по одному мультипликативному ребру E.*

Заметим, что критерий, приведённый в теореме, может быть **проверен за полиномиальное время**. Нахождение КСС и конденсации — это классическая задача теории графов, для решения которой существует много эффективных алгоритмов. Определение типов КСС и рёбер конденсации выполняется даже непосредственной проверкой определения за как максимум квадратичное время. Ну и, наконец, само условие из теоремы проверяется так: перебираем все КСС K и L, а также — рёбра E, о которых идёт речь. Далее проверяем, что есть пути из S^* в L и из L в K, либо же пути из S^* в L_1 и из L_2 в K, если ребро E ведёт из L_1 в L_2. Поиск пути в графе из одной вершины в другую может быть выполнен за полиномиальное время. Поэтому даже простой перебор указанных объектов тоже даст полиномиальную оценку времени.

Что касается самого **доказательства** теоремы, то его грубый набросок заключается в следующем. Если предположить, что критерий выполнен, то возьмём любой вывод пустого слова и постепенно дополним его до тотального. Для этого выберем нетерминал A и правило $A \to u$, которое не используется в выводе и для которого КСС A^* является минимально возможной для отношения достижимости в конденсации M_G^*. Тогда вершины из всех КСС меньших или равных A^* при выводе достигнуты. В частности, достигнута какая-то вершина B из КСС A^*.

Если правило $A \to u$ не соответствует цепному ребру конденсации M_G^*, то в КСС A^* существует цикл, который начинается и кончается в B и проходит по каждому из рёбер внутри A^* хотя бы раз. Тогда в выводе, получив нетерминал B, мы можем последовательно применить правила, соответствующие рёбрам цикла, и снова получить B. В результате мы сможем в наш вывод вставить применение правила $A \to u$, сохранив применение всех имеющихся.

Если правило $A \to u$ соответствует цепному ребру, исходящему из A^*, и оно является единственным таковым, то правило $A \to u$ должно быть применено, так как это — единственный способ избавиться от нетерминалов из КСС A^*, а они, напомним, в выводе присутствуют.

Наконец, если кроме цепного ребра $A \to u$ из A^* исходят цепные рёбра, помеченные иными правилами, то по условию существует путь из S^* в A^*, проходящий через мультипликативную КСС L или по мультипликативному ребру E, ведущему из L_1 в L_2. Напомним, что мы выбрали A, который является минимальным, поэтому какие-то вершины из L или, соот-

ветственно, L_1 были достигнуты в ходе нашего вывода. Тогда мы можем в КСС L или, соответственно, L_1 обнаружить цикл и, используя правила, соответствующие рёбрам этого цикла, породить в конечном итоге нужное количество нетерминалов из КСС A^*, чтобы их хватило для применения всех правил вида $A \to u$, соответствующих цепным рёбрам.

Пусть теперь, наоборот, критерий не выполнен, то есть имеется цепная КСС, из которой исходят хотя бы два цепных ребра, соответствующих правилам $A \to u$ и $A \to v$ и никакой путь из S^* в КСС $A^* = B^*$ не проходит через мультипликативную КСС или по мультипликативному ребру. Без ограничения общности считаем, что такая КСС $A^* = B^*$ выбрана минимальной для отношения достижимости в конденсации M_G^*. Иными словами, все меньшие $A^* = B^*$ КСС $K_1, K_2, \ldots, K_n$ являются цепными и из каждой из них в направлении КСС $A^* = B^*$ исходит по одному ребру, необходимо цепному. Тогда в любом выводе в каждом его слове присутствует не более одного нетерминала из множества $K_1 \cup K_2 \cup \ldots \cup K_n \cup A^*$. Но тогда никакой вывод не может одновременно содержать применение правил $A \to u$ и $A \to v$, то есть не является тотальным.

Заключение

Мы показали разрешимость задачи тотальной выводимости пустого слова в произвольной KC-грамматике за полиномиальное время. Естественным обобщением будет вопрос о существовании полиномиального алгоритма для решения следующей задачи: для заданного фиксированного (терминального) слова w и произвольной KC-грамматики G определить, является ли слово w тотально выводимым в грамматике G.

Литература

1. Сложность исчислений Ламбека с модальностями и тотальной выводимости в грамматиках / С. М. Дудаков, Б. Н. Карлов, С. Л. Кузнецов, Е. М. Фофанова // Алгебра и логика. – 2021. – Т. 60, № 5. – С. 471–496. – DOI 10.33048/alglog.2021.60.502. – EDN SOVAEC.

УДК 519.7

ОБ ИДЕАЛЬНОЙ АРИФМЕТИЧЕСКОЙ АВТОКОРРЕЛЯЦИИ ЧЕТВЕРТИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

В. А. Едемский, Р. А. Вахитов

Новгородский государственный университет им. Ярослава Мудрого

Аннотация. Исследована арифметическая автокорреляция четвертичных последовательностей, синтезируемых регистром сдвига с обратной связью с переносом. Показано, что не существует семейства таких последовательностей, в котором каждая последовательность имеет идеальную арифметическую автокорреляцию. Для изучения арифметической автокорреляции рассматривается распределение четвертичных вычетов по простому модулю и его связь с суммами характеров Дирихле.
Ключевые слова: регистр сдвига с обратной связью с переносом, четвертичные последовательности, арифметическая автокорреляция, характеры Дирихле.

Введение

Различные регистры сдвига широко применяются для получения псевдослучайных последовательностей, обладающих свойствами, важными для их разнообразных практических применений. В частности, с этой целью используется регистр сдвига с обратной связью с переносом (feedback with carry shift register), предложенный в [1]. Его свойства подробно описаны в [2]. Последовательности, синтезируемые этим регистром, обычно называют FCSR-последовательностями.

Арифметическая корреляция или корреляция с переносом является аналогом классической корреляции и является важной характеристикой последовательностей [1]. Имеется множество статей посвященных исследованию арифметической корреляции различных последовательностей [2–6]. Но, для N-ичных последовательностей известны только результаты о арифметической автокорреляции l-последовательностей, период которых равен значению функции Эйлера от числа связи регистра. В этой статье рассмотрим арифметическую автокорреляцию четвертичных FCSR-последовательностей. Четвертичные последовательности наиболее часто применяемый вид последовательностей после бинарных.

1. Основные определения

Прежде всего, напомним основные определения из [2]. Пусть s^∞ — N-ичная FCSR-последовательность с периодом T и числом связи q. Тогда, согласно теореме 4.8.1 из [2], члены последовательности s^∞ могут быть найдены по следующей формуле:

$$s_n = A(f 4^{-n} \bmod q) \bmod N \tag{1}$$

для $n = 0,1,2,...$, где $A = -q^{-1} \bmod N$. Здесь через $x \bmod q$ обозначаем наименьший неотрицательный вычет x по модулю q.

Любой N-ичной последовательности s^∞ с периодом T можно поставить в соответствие N-адическое $s_0 + s_1 \cdot N + \ldots + s_n \cdot N^n + \ldots$ и рациональное число

$$-\frac{f}{q} = \frac{\sum_{j=0}^{T-1} s_j N^j}{1 - N^T}, \gcd(f,q) = 1.$$

Несбалансированность N-ичной последовательности и соответствующего N-адического числа определяется как

$$Z(s) = \sum_{j=0}^{N-1} a_j \xi^j, \tag{2}$$

где a_j — число появлений j на одном периоде последовательности s^∞ для $j = 0,1,\ldots,N-1$ и $\xi = \exp(2\pi \cdot i / N)$ — комплексный корень N-й степени из единицы, $i = \sqrt{-1}$.

Пусть последовательность s_τ^∞ получается посредством циклического сдвига посследовательности s^∞ на $\tau : 0 < \tau < T$. Обобзначим через α и α_τ N-адические числа, соответствующие последовательностям s^∞ и s_τ^∞, и рассмотрим последовательность, определяемую N-адическим числом $\alpha - \alpha_\tau$. Она является почти периодической с периодом, делящим T. Её несбалансированность на периоде T называется арифметической автокорреляцией последовательности s^∞ при сдвиге τ и обозначается $\mathcal{A}_s^A(\tau)$ [2]. Согласно [2], имеем

$$\mathcal{A}_s^A(\tau) = Z\left(\frac{f(N^{-\tau}-1) \bmod q}{q}\right) \tag{3}$$

для $\tau = 1,2,\ldots,T-1$.

Если $\mathcal{A}_s^A(\tau) = 0$ для всех $\tau = 1,2\ldots,T-1$, то s^∞ называется последовательностью с идеальной арифметической автокорреляцией. В этой статье покажем, что при любом q семейство четвертичных FCSR-последовательностей не может обладать идеальной автокорреляцией. Ясно, что в силу формулы (3) это достаточно доказать для случая, когда $q = p$, где p — простое число.

2. Вспомогательные леммы

В этом разделе докажем насколько вспомогательных утверждений, необходимых для дальнейшего. Пусть $q = p$, где p — простое число, и четвертичная последовательность s^∞ с периодом T определена по (1), тогда значение T равно порядку 4 по модулю p. Согласно (1), $A = 3$, если $p \equiv 1 (\mathrm{mod}\, 4)$ и $A = 1$, если $p \equiv 3 (\mathrm{mod}\, 4)$.

Обозначим через $\mathbb{Z}_p^*$ мультипликативную группу поля $\mathbb{Z}_p$ порядка p. Хорошо известно, что $\mathbb{Z}_p^*$ — циклическая группа. Пусть $H_0 = \{4^{-n} \bmod p, n = 0,1,\ldots,T-1\}$. Тогда H_0 — подгруппа $\mathbb{Z}_p^*$ и фактор-группа $\mathbb{Z}_p^* / H_0$ также является циклической, то есть существует элемент g такой, что классы смежности $H_0, gH_0, \ldots, g^{(p-1)/T-1}H_0$ образуют разбиение $\mathbb{Z}_p^*$. Обозначим $g^k H_0$ через H_k. Определим несбалансированность $Z(H_k)$ множества H_k как для последовательности, расматривая его элементы по модулю 4.

Лемма 1. *Пусть s определена по (1) с числом связи p. Последовательность s имеет идеальную арифметическую автокорреляцию для каждого $f = 1,2,\ldots,p-1$ тогда и только тогда, когда $Z(H_k) = 0$ для $k = 0,1,\ldots,(p-1)/T$.*

Доказательство. Предположим, что s имеет идеальную арифметическую автокорреляцию для каждого $f = 1,2,\ldots,p-1$. Тогда $Z\left(\frac{f(4^{-\tau}-1) \bmod p}{p}\right) = 0$ для $\tau = 1,2,\ldots,T-1$. Пусть $a = f(4^{-\tau}-1) \bmod p$ для некоторого τ. Рассмотрим последовательность u^∞ с $Z(u) = 0$, соответствующую рациональному числу a/p. Тогда $u_n = A(a4^{-n} \bmod p) \bmod 4$, $n = 0,1,\ldots,T-1$. Пусть $D = \{a4^{-n} \bmod p,\ n = 0,1,\ldots,T-1\}$. Как отмечено выше, $A \equiv \pm 1 (\mathrm{mod}\, 4)$, следовательно, $Z(D) = 0$. Множество D совпадает с одним из классов смежности H_0. Так как $f = 1,2,\ldots,p-1$, то таким образом получаем все различные классы смежности H_0, то есть $Z(H_k) = 0$ для $k = 0,1,\ldots,(p-1)/T$.

Пусть $Z(H_k) = 0$ для $k = 0,1,\ldots,(p-1)/T$. Тогда множества $\{\pm a4^{-n} \bmod 4,\ n = 0,1,\ldots,T-1\}$ сбалансированы для всех значений $a = 1,2,\ldots,p-1$. Следовательно, по (3)

$Z\left(\frac{f(N^{-\tau}-1) \bmod p}{p}\right)=0$ для каждого $\tau=1,2,\ldots,T-1$ и $\mathcal{A}_s^A(\tau)=0$ для каждого $\tau=1,2\ldots,T-1$.

Согласно (2), для четвертичной последовательности

$$Z(s)=a_0+a_1\cdot i-a_2-a_3\cdot i.$$

Следовательно, если $Z(s)=0,$ то

$$a_0=a_2,\, a_1=a_3. \tag{4}$$

Определим $I_k=\{a\in\mathbb{Z}\,|\,[kp/4]\le a<[(k+1)p/4],\, k=0,1,2,3,$ где $[x]$ — целая часть числа x.

Лемма 2. *Пусть s определена по (1) с числом связи p и $D=\{f4^{-n} \bmod p, n=0,1,\ldots,T-1\}$. Тогда:*

1) $a_k=|I_k\cap D|,\ k=0,1,2,3$, если $p\equiv 3(\bmod 4)$;

2) $a_0=|I_0\cap D|,\ a_k=|I_{4-k}\cap D|,\ k=1,2,3$, когда $p\equiv 1(\bmod 4)$.

Доказательство. Докажем данную лемму для $p\equiv 3(\bmod 4)$. В этом случае $Z(s)=Z(D)$, где $D=\{f4^{-n} \bmod p, n=0,1,\ldots,T-1\}$. Пусть $a\in D$ и $a\in I_k$. По определению D имеем, что $4a \bmod p\in D$. Рассмотрим четыре случая.

1) Пусть $k=0$. Тогда $4a \bmod p=4a$ и $(4a \bmod p) \bmod 4=0$;

2) Если $k=1$, то $a=(p-3)/4+b$, где b — целое и $1\le b<(p+1)/4$. Значит $4a=p-3+4b$, $4a \bmod p=-3+4b$ и $(4a \bmod p) \bmod 4=1$;

3) Пусть $k=2$, тогда $a=(p-1)/2+b$. Тогда $4a=2p-2+4b$, $4a \bmod p=-2+4b$ и $(4a \bmod p) \bmod 4=2$;

4) Наконец, если $k=3$, то $a=(3p-1)/4+b$. Значит, $4a=3p-1+4b$, $4a \bmod p=-1+4b$ и $(4a \bmod p) \bmod 4=3$. Таким образом, $a_k=|I_k\cap D|$.

Если же $p\equiv 1(\bmod 4)$, то $D=\{-(f4^{-n} \bmod p), n=0,1,\ldots,T-1\}$ и второе утверждение этой леммы может быть показано тем же самым способом, что и первое.

Лемма 3. *Пусть четвертичная последовательность s определена по (1) с числом связи p и $D=\{f4^{-n} \bmod p, n=0,1,\ldots,T-1\}$. Тогда $|D\cap(0,p/2)|=T/2$, более того, если $-1\in H_0$, то $|D\cap(0,p/4)|=T/4$.*

Доказательство. Первое утверждение леммы очевидным образом следует из (4) и предыдущей леммы. Если $-1\in H_0$, то $(p-a)\in D$ для любого $a\in D$. Следовательно, $a_0=a_1$ или $a_0=a_3$ для $p\equiv\pm 1(\bmod 4)$. Использование формул (4), завершает доказательство.

3. Арифметическая автокорреляция

Определим характер χ из $\mathbb{Z}_p^*$ в множество комплексных чисел $\mathbb{C}$ следующим образом

$$\chi(a)=\xi^k,\ \textit{если}\ a\in H_k, \tag{5}$$

где ξ — первообразный комплексный корень степени $(p-1)/T$ из 1.

Характер χ можно распространить на множество целых чисел и рассматривать его как характер Дирихле по модулю p считая, что $\chi(a)=0$, когда $НОД(a,p)\ne 1$, и $\chi(a)=\chi(a \bmod p)$ в противном случае.

Теорема 4. *FCSR-последовательности не могут иметь идеальную арифметическую автокорреляцию при каждом $f=1,2,\ldots,p-1$.*

Доказательство. Будем доказывать теорему от противного. Пусть для любого $f=1,2,\ldots,p-1$ последовательности имеют идеальную арифметическую автокорреляцию.

Рассмотрим два случая.

1. Пусть $-1\notin H_0$. Тогда χ — нечетный характер и по лемме 1 получаем, что

$$\sum_{1\le a<p/2}\chi(a)=\sum_{k=0}^{(p-1)/T-1}\ \sum_{a\in(I_0\cup I_1)\cap H_k}\chi(a)=\frac{T}{2}\sum_{j=0}^{T/N-1}\xi^j=0. \tag{6}$$

С другой стороны, так как χ — нечетный, притмитивный и нетривиальный, то по теореме 3.2 из [7] имеем, что

$$\sum_{1\le a<p/2}\chi(a)=\frac{G}{2\pi i}\big(2-\chi(2)\big)L\Big(1,\overline{\chi}\Big),$$

где $i=\sqrt{-1}$, G — гауссова сумма и $L(1,\overline{\chi})=\sum_{n=1}^{\infty}\overline{\chi(n)}n^{-1}$ — Дирихле L-функция. Так как значения гауссовой суммы и Дирихле L-функции не равны нулю, получаем противоречие с (6).

2. Пусть $-1\in H_0$. Значит, здесь существует m такое, что $4^m\equiv -1(\mathrm{mod}\ p)$ или $2^{2m}\equiv -1(\mathrm{mod}\ p)$. Следовательно, 4 делит $p-1$ и $p\equiv 1(\mathrm{mod}\,4)$. Тогда, как и в первом случае по лемме 1 получаем, что

$$\sum_{1\le a<p/4}\chi(a)=0, \tag{7}$$

в этом случае χ — четный характер. Тогда по теореме 3.7 из [7] имеем, что

$$\sum_{1\le a<p/4}\chi(a)=\frac{G}{\pi}L\Big(1,\overline{\chi}_{4p}\Big).$$

Снова получаем противоречие.

Заключение

В настоящей работе рассмотрена арифметическая автокорреляция семейства четвертичных последовательностей, синтезируемых регистром сдвига с обратной связью с переносом. Показано, что в отличии то двоичных последовательностей здесь не существует семейств с идеальной арифметической автокорреляцией каждой последовательности.

Благодарности

Работа выполнена при поддержке Российского научного фонда, проект № 24–21–00442

Литература

1. *Goresky M.* Arithmetic crosscorrelations of feedback with carry shift register sequences / M. Goresky, A. Klapper // IEEE Trans. Inf. Theory Hofer R., Winterhof A. – 1997. – V. 43. – C. 1342–1345.

2. *Goresky M.* Algebraic Shift Register Sequences / M. Goresky, A. Klapper. – Cambridge University Press, 2012.

3. *Klapper A.* Feedback shift registers, 2-adic span and combiners with memory / A. Klapper, M. Goresky // J. Cryptol. – 1997.– V. 10. – C. 111–147.

4. *Chen Z.* Arithmetic autocorrelation of binary m-sequences / Z. Chen, Z. Niu, Y. Sang, C. Wu // Cryptologia.–2023. – V. 47. C. 449–458.

5. *Chen Z.* Arithmetic crosscorrelation of pseudorandom binary sequences of coprime periods / Z. Chen, Z. Niu, A. Winterhof // IEEE Trans. Inf. Theory. – 2022. –V. 68. – C. 7538–7544.

6. *Chen Z.* Arithmetic correlation of binary half-l-sequences / Z. Chen, V. Edemskiy, Z. Niu, Y. Sang // IET Inf. Secur. – 2023. – V. 17. – C. 289–293.

7. *Berndt B.* Classical theorems on quadratic residues / B. Berndt // Enseignement Math. – 1976. – V. 22. – C. 261–304.

УДК 004

ИНТЕГРАЦИЯ УМНЫХ ПОМОЩНИКОВ В ОРГАНИЗАЦИЮ КАК СЛЕДУЮЩИЙ ШАГ РАЗВИТИЯ ПРОГРАММНОЙ РОБОТИЗАЦИИ

А. В. Калач[1], Т. Е. Смоленцева[2], Я. А. Акатьев[2]

[1]Воронежский институт ФСИН России
[2]МИРЭА – Российский технологический университет

Аннотация. В работе рассмотрены особенности развития технологии программной роботизации и выделен обособленный вид — умный помощник. Представлена внутренняя структура умного помощника и типовой функционал. Представлено обоснование внедрения умного помощника в структуру управления организацией, а также особенности интеграции в организацию. Представлены результаты внедрения умного помощника в Институт информационных технологий РТУ МИРЭА. Делается вывод о необходимости дополнительной классификации термина «программная роботизация» на «интеллектуальные» и «автоматизированные».
Ключевые слова: умный помощник, RPA, структура управления, IT-архитектура, нейронные сети, работа с данными, принятие решений, чат-бот, виртуальный ассистент, внутренняя структура, корпоративные информационные системы.

Введение

Программные роботы активно используются для автоматизации рутинных задач, связанных с выполнением бизнес-процессов в организациях. Их внедрение способствует значительной оптимизации работы: выполнение типовых задач становится более быстрым и менее затратным в плане человеческих ресурсов. Это позволяет организациям улучшить производительность, минимизировать ошибки, характерные для ручного труда, и оптимально перераспределять рабочую нагрузку. Благодаря этому сотрудники могут быть перенаправлены на более сложные, аналитические или стратегически важные задачи, требующие высокого уровня квалификации и творческого подхода. Таким образом, программные роботы играют ключевую роль в повышении эффективности бизнес-процессов, а также в достижении более высокой адаптивности организаций к изменяющимся условиям рынка.

Рынок программных роботов в Российской Федерации находится в стадии активного роста. Данный рост обусловлен множеством факторов, включая необходимость импортозамещения и усиление внутренней технологической независимости после ухода ряда иностранных компаний с российского рынка в 2022 году. В ответ на это на рынке появляются новые отечественные решения и разработки в сфере роботизации процессов. Отечественные компании стремятся удовлетворить потребности в автоматизации бизнес-процессов и предлагают новые подходы и технологии, адаптированные к особенностям российского рынка и требованиям локальных организаций.

Рассмотренный процесс способствует созданию более конкурентной среды на рынке программных роботов в России. Повышение конкуренции стимулирует разработчиков к созданию совершенных, надежных и функциональных решений, что, в свою очередь, обеспечивает организациям доступ к широкому спектру инструментов для оптимизации их бизнес-процессов.

Ключевые Российские компании в области RPA – ElectroNeek, Robin, TensorFlow RPA, RPA. PRO, TsuruRobotics, RAZR Technology. Эти компании предоставляют локализованные продукты, которые соответствуют требованиям российских предприятий и адаптированы под отечественные регуляторные нормы (рис. 1).

Рис. 1. Рынок RPA в Российской Федерации

Большинство организаций используют роботов для следующего набора задач (табл. 1).

Таблица 1

Класс задач умных помощников

Класс задач	Описание
Работа с данными	Роботы могут работать сразу с несколькими программными решениями и базами данных, необходим только написанный алгоритм
Принятие решений	Роботы могут самостоятельно принимать решения, основанные на заранее заложенном в них алгоритме. Принятие решения сопровождается строгим набором правил, а в случае возникновения исключения робот может уведомлять об этом сотрудника организации эскалируя решение возникшей проблемы
Проверка данных	Проверка данных в документах или генерация документов по типовым шаблонам на основе данных
Сбор данных	Работа с неструктурированным объемом информации либо сбор данных в полуавтономном режиме.

Сфера применения программных роботов непрерывно расширяется. Ведущие компании, стремящиеся сохранить лидирующие позиции на рынке цифровых решений, активно исследуют новые направления для применения программных роботов в различных предметных областях, а также работают над расширением их функциональных возможностей и анализом перспектив их дальнейшего развития.

Исходя из этого возникла потребность в формализации и классификации таких программных компонентов. Авторами предлагается использование терминологии «Умный помощник».

Интеллектуальный помощник обладает рядом функциональных особенностей, которые определяются его внутренней структурой и предназначены для обеспечения высокой адаптивности и эффективности в выполнении различных задач. Эти особенности можно выделить по следующим основным направлениям:

Автоматизируемые бизнес-процессы. Умный помощник охватывает широкий спектр бизнес-процессов, которые могут быть автоматизированы без привязки к конкретной предметной области. Такой подход позволяет умному помощнику решать разнообразные задачи, универсально применимые в разных сферах деятельности организации. Данный фактор обеспечивает возможность его использования для поддержки как фронтальных, так и бэк-офисных процессов, независимо от их функциональной направленности.

Интеграция с корпоративными информационными системами (КИС). Важной характеристикой умного помощника является его интеграция с развернутыми в организации информационными системами, такими как системы управления взаимоотношениями с клиентами (CRM), системы управления ресурсами предприятия (ERP), и другие. Такая интеграция позволяет помощнику эффективно взаимодействовать с данными и процессами в рамках единой IT-архитектуры, повышая оперативность и качество взаимодействия с пользователями и снижая нагрузку на другие ИТ-системы организации (рис. 2).

Рис. 2. Улучшение бизнес-операция с помощью умных помощников

Роль в организационной структуре. Умный помощник занимает определенное место в организационной структуре компании, выполняя функции как для клиентов, так и для сотрудников. С точки зрения клиентов, помощник может стать интерфейсом для взаимодействия с компанией, обеспечивая поддержку пользователей и обработку запросов. С точки зрения сотрудников, умный помощник выступает в роли инструмента, облегчающего выполнение рутинных и повторяющихся задач, позволяя специалистам сосредоточиться на более сложных и стратегических аспектах своей работы. Такой подход способствует улучшению клиентского опыта и повышению производительности труда (рис. 3).

Функциональные возможности. Умный помощник обладает широким спектром функциональных возможностей, которые включают в себя автоматизацию рутинных задач, анализ данных, поддержку в принятии решений и другие виды поддержки. Данные возможности могут быть адаптированы под потребности конкретного подразделения или бизнес-процесса, что делает помощника гибким инструментом, способным решать задачи различной сложности.

Кроме того, модель принятия решений является ключевым модулем умного помощника, обобщающим его функциональность с учетом технологий искусственного интеллекта. Модель

Рис. 3. Роли умного помощника

опирается на генеративные нейронные сети, а также методы, основанные на разметке данных, что позволяет интеллектуальному помощнику эффективно обучаться и адаптироваться к новым условиям и задачам. Модель принятия решений позволяет помощнику обрабатывать более сложные запросы, которые требуют не только автоматизации, но и элементарного уровня рассуждений и адаптивности.

В общем виде предлагаемое решение представляет собой информационную систему, взаимодействие с которой осуществляется через обмен сообщениями в социальных сетях. Примером такого взаимодействия может служить чат-бот, с которым сотрудники организации могут взаимодействовать для выполнения задач. В этом случае, вместо непосредственного отчёта перед руководителями, сотрудники могут взаимодействовать с умным помощником, который получает и передаёт данные в систему управления, ориентированную на контроль и координацию поставленных задач.

Таким образом, умный помощник или чат-бот становится частью системы управления организацией, занимая определённое место в её иерархии. Независимо от структуры управления, умный помощник выполняет задачи уровня низшего менеджмента, обеспечивая выполнение определённых управленческих функций и поддержку рабочих процессов.

Внедрение умного помощника в организацию существенно упрощает множество процессов управления организацией. Достигается это в том числе применением асинхронности процесса. Менеджер среднего звена избавлен от необходимости ожидания отклика сотрудника на поставленное задание, так как оно автоматически интегрируется в его календарный план. Сотрудник, в свою очередь, может оперативно получать справочную информацию без запроса согласований или подтверждений от вышестоящих подразделений, обращаясь напрямую к системе умного помощника.

Для оценки практической значимости концепции интеграции умного помощника в структуру управления приведён пример. В Институте информационных технологий РТУ МИРЭА обучается более шести тысяч студентов и работает несколько сотен преподавателей.

В данном случае частичная замена или дополнение функций нижнего управленческого звена умным помощником позволяет эффективно решить перечисленные задачи. В данной предметной области виртуальный ассистент может взять на себя часть обязанностей старосты и куратора студенческой группы из состава преподавателей.

Подробное описание интеграции представлено в более ранних работах.

Заключение

Развитие концепции непрямого управления через умного помощника требует разработки классификации программных роботов на «интеллектуальные» и «автоматизированные», что станет предметом дальнейших научных исследований.

Литература

1. *Гурлев И. В.* Цифровизация экономики России и проблемы роботизации / И. В. Гурлев // Вестник евразийской науки. – 2020. – Т. 12, № 4. – С. 36.

2. *Лобачева А. С.* Этика применения искусственного интеллекта в управлении персоналом / А. С. Лобачева, О. В. Соболь // E-Management. – 2021. – Т. 4, № 1. – С. 20–28.

3. *Сидоров А. В.* Роботизация бизнес-процессов как инструмент повышения производительности труда сотрудников компании / А. В. Сидоров // Хроноэкономика. – 2019. – № 4(17). – С. 64–68.

4. *Шермухамедов Б. А.* Системы искусственного интеллекта в банковской сфере / Б. А. Шермухамедов // Россия: тенденции и перспективы развития : Ежегодник, Курск, 04–05 июня 2021 г. Том Выпуск 16. Часть 2. – Москва: Институт научной информации по общественным наукам РАН, 2021. – С. 523–525.

5. *Смоленцева Т. Е.* Концепция применения умных помощников в управлении организацией на примере института информационных технологий РТУ МИРЭА / Т. Е. Смоленцева, Я. А. Акатьев // Труды международного симпозиума «Надежность и качество». – 2023. – Т. 1. – С. 344–347.

УДК 510.52, 519.766.23

ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ ПРОБЛЕМЫ ТОТАЛЬНОЙ ВЫВОДИМОСТИ В НЕУКОРАЧИВАЮЩИХ ГРАММАТИКАХ

Б. Н. Карлов

Тверской государственный университет

Аннотация. В работе изучается проблема тотальной выводимости в неукорачивающих и контекстно-зависимых грамматиках. Проблема состоит в том, чтобы по грамматике и терминальному слову определить, существует ли вывод этого слова, в котором каждое правило используется не менее некоторого заданного числа раз. Доказывается, что эта проблема является NP-полной для любого фиксированного слова длины не менее 2, а также что она разрешима за полиномиальное время для слов длины 1. Аналогичные результаты получены и для варианта проблемы тотальной выводимости с ограничением на число использований нетерминалов в выводе.

Ключевые слова: неукорачивающая грамматика, контекстно-зависимая грамматика, тотальная выводимость, вычислительная сложность, NP-полная проблема.

Введение

Одной из основных алгоритмических проблем в теории формальных языков является проблема принадлежности. В этой задаче на вход подаётся некоторое описание языка (грамматика или автомат) и слово. Алгоритм должен определить, принадлежит ли слово языку. Известно, что эта проблема неразрешима в классе всех порождающих грамматик (см. [5]) и является PSPACE-полной в классе всех контекстно-зависимых грамматик (см. [3]). Для контекстно-свободных грамматик существуют более эффективные алгоритмы анализа. Так, алгоритмы Кока — Янгера — Касами и Эрли (см. [5]) решают проблему принадлежности за время $O(n^3)$, где n — длина входного слова, а алгоритм Валианта (см. [6]) — за время $O(M(n))$, где $M(n)$ — время умножения двух булевых матриц размера $n \times n$.

В статье [4] в связи с исследованием некоторых вариантов исчисления Ламбека было введено понятие тотальной выводимости слова в грамматике. В этом определении требуется, чтобы каждое правило грамматики применялось в выводе не менее некоторого указанного числа раз. В этой же статье была исследована вычислительная сложность проблемы тотальной выводимости для некоторых классов порождающих грамматик. В частности, было доказано, что она является m-полной для произвольных порождающих грамматик, PSPACE-полной для неукорачивающих и контекстно-зависимых грамматик, NP-полной для контекстно-свободных грамматик. При этом рассматривался наиболее общий вариант проблемы, когда на вход алгоритму подаётся как грамматика, так и слово.

В настоящей работе мы изучаем вариант проблемы тотальной выводимости, когда слово фиксировано, а на вход алгоритму подаётся только неукорачивающая или контекстно-зависимая грамматика. Кроме того, мы определяем другой вариант тотальной выводимости, в котором накладываются ограничения не на количество применений правил, а на количество вхождений нетерминалов в вывод. Мы доказываем, что такие варианты проблемы тотальной принадлежности являются NP-полными для слов длины не менее 2. Если же длина слова равна 1, то обе проблемы разрешимы за полиномиальное время.

1. Основные определения

Напомним определения основных понятий, которые будут использоваться в дальнейшем. *Алфавитом* называется конечное множество символов, а *словом* — конечная последовательность символов. Через Σ^* обозначается множество всех слов в алфавите Σ. *Длиной* слова называется число символов в нём. *Порождающая грамматика* — это четвёрка $G = (N, \Sigma, P, S)$, где N — алфавит *нетерминалов*, Σ — алфавит *терминалов*, при этом $N \cap \Sigma = \varnothing$, P — множество *правил вывода*, $S \in N$ — *начальный* нетерминал. Правила вывода имеют вид $\alpha \to \beta$, где $\alpha, \beta \in (N \cup \Sigma)^*$, при этом α содержит хотя бы один нетерминал. Слово y выводимо из слова x за один шаг в грамматике G (обозначается $x \Rightarrow_G y$), если $x = u\alpha v$, $y = u\beta v$ для некоторых $u, v \in (N \cup \Sigma)^*$, $\alpha \to \beta \in P$. Если грамматика G ясна из контекста, то мы будем опускать индекс и писать просто $\Rightarrow$. Через $\Rightarrow^*_G$ обозначается рефлексивное транзитивное замыкание отношения $\Rightarrow_G$, а через $\Rightarrow^+_G$ — его транзитивное замыкание. Слово x *выводимо* из слова y в грамматике G, если $x \Rightarrow^*_G y$. Язык, *порождаемый* грамматикой G, — это множество всех слов в алфавите Σ, выводимых из начального нетерминала S. Порождающая грамматика называется *неукорачивающей*, если для любого правила $\alpha \to \beta$ выполняется неравенство $|\alpha| \le |\beta|$. Порождающая грамматика называется *контекстно-зависимой*, если все её правила имеют вид $\alpha A \beta \to \alpha\gamma\beta$, где $A \in N$, $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$, при этом γ непусто.

Мультимножество — это произвольная совокупность элементов, в которой каждый элемент может встречаться произвольное количество раз (в этом состоит отличие от множества). Число вхождений элемента x в мультимножество A называется его *кратностью*. *Суммой* мультимножеств A и B называется мультимножество $A \uplus B$, в котором кратность каждого элемента равна сумме его кратностей в A и B. Запись $A \subseteq B$ означает, что кратность каждого элемента мультимножества A не превосходит его кратности в B. Мы будем использовать также *обобщённые* мультимножества, где элементы могут иметь кратность ∞. При этом $\infty \ge n$ и $\infty + n = \infty$ для любого числа n. При реализации алгоритмов ∞ представляется булевым флагом. В определении R-тотальной выводимости, однако, R будет мультимножеством без элементов кратности ∞.

Определение 1. Пусть R — некоторое мультимножество правил грамматики G. Вывод в грамматике называется *R-тотальным*, если каждое правило используется в нём не меньшее число раз, чем кратность этого правила в R.

Определение 2. Пусть M — некоторое мультимножество нетерминалов грамматики G. Вывод в грамматике называется *M-тотальным*, если каждый нетерминал встречается в нём не меньшее число раз, чем кратность этого нетерминала в M.

Через P (соответственно NP) обозначается класс всех алгоритмических проблем, разрешимых детерминированными (соответственно недетерминированными) алгоритмами за полиномиальное время (точные определения этих понятий можно найти в [3]). Алгоритмическая проблема A *полиномиально сводится* к проблеме B (обозначается $A \le_p B$), если существует функция f, вычислимая детерминированным алгоритмом за полиномиальное время и такая, что для любого входа x справедлива следующая эквивалентность: $x \in A$ тогда и только тогда, когда $f(x) \in B$. Проблема A называется *NP-трудной*, если $B \le_p A$ для любой проблемы $B \in \mathrm{NP}$. Проблема A называется *NP-полной*, если $A \in \mathrm{NP}$ и A NP-трудна. Для доказательства NP-трудности используется следующее свойство (см. [3]).

Лемма 1. *Если $A \le_p B$ и проблема A является NP-трудной, то проблема B тоже NP-трудна.*

Для доказательства результатов об NP-полноте мы будем использовать задачу *о вершинном покрытии*. Пусть $G = (V, E)$ — неориентированный граф. Множество вершин $V' \subseteq V$ называется вершинным покрытием, если для любого ребра из E хотя бы один из его концов принадлежит V': из $(u, v) \in E$ следует, что $u \in V'$ или $v \in V'$. Задача о вершинном покрытии состоит в

следующем: по неориентированному графу $G=(V,E)$ и натуральному числу $k \le |V|$ нужно определить, существует ли в G вершинное покрытие мощности не более k. Известно, что задача о вершинном покрытии является NP-полной (см. [3]).

2. Алгоритмы проверки тотальной выводимости слова в неукорачивающих грамматиках

Сначала мы опишем недетерминированный алгоритм для проверки R-тотальной выводимости слова в неукорачивающей грамматике для мультимножества правил R.

Теорема 1. *Пусть $G=(N,\Sigma,P,S)$ — неукорачивающая грамматика, $w \in \Sigma^*$ — терминальное слово, R — мультимножество правил. Пусть $|w|=n$, $|\Sigma|=k$, $|N|=m$, $|P|=p$. Тогда R-тотальную выводимость слова w в грамматике G можно проверить недетерминированным алгоритмом за время $O(n(k+m)^n((k+m)^n+p))$.*

Доказательство. Будем называть правило $\alpha \to \beta$ удлиняющим, если $|\alpha| < |\beta|$. Применение удлиняющего правила увеличивает длину слова по меньшей мере на 1. Так как начальное слово уже содержит один символ S, то вывод слова w длины n не может содержать больше $n-1$ применения удлиняющих правил. Если общее число удлиняющих правил в R с учётом кратности больше $n-1$, то R-тотального вывода не существует. Далее мы будем считать, что число удлиняющих правил в R не превосходит $n-1$.

Пусть $\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_l$ — произвольный вывод (не обязательно R-тотальный). Будем называть его ациклическим, если $\alpha_i \ne \alpha_j$ при $i \ne j$. Вывод называется циклом, если $\alpha_1 = \alpha_l$. Если при этом $\alpha_i \ne \alpha_j$ при $1 \le i < j < l$, то цикл называется простым. Задача R-тотальной выводимости решается приведённым ниже недетерминированным алгоритмом. В этом алгоритме A и Z — мультимножества, а W — множество.

1. Угадать ациклический вывод $\pi : \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_l$, где $\alpha_1 = S$, $\alpha_l = w$.
2. Поместить в мультимножество A номера всех правил, встретившихся в π, с учётом кратности.
3. Положить $W = \{\alpha_1, \ldots, \alpha_l\}$.
4. Угадать число q и последовательность непустых попарно различных слов $U = (\beta_1, \beta_2, \ldots, \beta_q)$ в алфавите $N \cup \Sigma$ длины не более n.
5. Для каждого $i = 1, 2, \ldots, q$ выполнить пункты 6–8.
6. Угадать слово $\beta \in W$ и простой цикл $\pi' : \beta \Rightarrow^+ \beta_i \Rightarrow^+ \beta$.
7. Добавить в W все слова, встретившиеся в π'.
8. Добавить в A номера всех правил, встретившихся в π', с учётом кратности.
9. Положить $Z = \varnothing$.
10. Для каждого правила $\rho \in P$ выполнить шаги 11–12.
11. Либо перейти к следующему правилу, либо угадать слово $\gamma \in W$ и простой цикл $\pi'' : \gamma \Rightarrow^+ \gamma$, в котором применяется правило ρ.
12. Добавить в Z номера всех правил, встретившихся в выводе π'', с кратностью ∞.
13. Если $R \subseteq A \uplus Z$, то вернуть «да», иначе вернуть «нет».

Докажем, что алгоритм возвращает «да» тогда и только тогда, когда существует R-тотальный вывод $S \Rightarrow^* w$.

Пусть некоторое вычисление алгоритма возвращает «да». Это значит, что на шаге 1 алгоритм успешно угадал ациклический вывод $S \Rightarrow^+ w$, а далее на шагах 5–8 и 10–12 расширил его циклами. При этом кратности всех правил, использованных до шага 10, были непосредственно учтены в мультимножестве A. Поскольку все правила из Z находятся в циклах, то их можно применить сколь угодно много раз, повторяя эти циклы. Из включения $R \subseteq A \uplus Z$ следует, что такой расширенный вывод является R-тотальным.

Теперь предположим, что существует R-тотальный вывод $\pi : S \Rightarrow^+ w$. Если он ациклический, то алгоритм просто угадывает его в строке 1, пропускает расширение вывода циклами и возвращает «да»» в строке 13. Предположим, что в выводе есть циклы. Удалим их все и получим новый вывод π', который уже является ациклическим. В строке 1 алгоритм угадывает π', а в строке 4 он угадывает слова, которые встречаются в π, но не в π'. В строках 5–8 алгоритм угадывает циклы, которые позволяют получить слова β_i, и добавляет в A все использованные при этом правила. В результате получается вывод, который ещё не обязательно является R-тотальным, но который уже содержит все слова, встречающиеся в π. Если кратности некоторых правил всё ещё меньше их кратностей в R, то в строках 9–12 алгоритм угадывает циклы, в которых применяются эти правила, и добавляет их в Z с бесконечной кратностью. Для получившегося вывода имеет место включение $R \subseteq A \uplus Z$, так как ациклическая часть вывода была угадана непосредственно. Поэтому в строке 13 алгоритм возвращает «да».

Теперь оценим время работы алгоритма. Мы будем предполагать, что в качестве модели вычислений выбраны машины с произвольным доступом к памяти (см. [2]). Общее число непустых слов длины не больше n в алфавите $N \cup \Sigma$ равно

$$r = (k+m) + (k+m)^2 + \ldots + (k+m)^n = (k+m)\frac{(k+m)^n - 1}{k+m-1} = O((k+m)^n).$$

Поэтому сумма длин всех таких слов составляет $O(nr) = O(n(k+m)^n)$. Следовательно, шаги 1–4 алгоритма могут быть выполнены за время $O(nr)$. Аналогично одна итерация цикла в строках 5–8 выполняется за время $O(nr)$. Так как число итераций не превосходит r, то весь цикл выполняется за время $O(nr^2)$. Одна итерация цикла в строках 10–12 также требует $O(nr)$ времени, а число итераций равно p, поэтому цикл выполняется за время $O(nrp)$. Проверка включения $R \subseteq A \uplus Z$ выполняется за время $O(p)$. Значит, общее время работы алгоритма составляет $O(nr + nr^2 + rnp + p) = O(nr(r+p)) = O(n(k+m)^n((k+m)^n + p))$. □

Аналогичный результат справедлив и для проблемы M-тотальной выводимости.

Теорема 2. *Пусть $G = (N, \Sigma, P, S)$ — неукорачивающая грамматика, $w \in \Sigma^*$ — терминальное слово, M — мультимножество нетерминалов. Пусть $|w| = n$, $|\Sigma| = k$, $|N| = m$. Тогда M-тотальную выводимость слова w в грамматике G можно проверить недетерминированным алгоритмом за время $O(n(k+m)^{2n})$.*

Доказательство. Задача решается аналогичным алгоритмом. Мультимножества A и Z хранят нетерминалы, встретившиеся в выводе. Цикл в строках 10–12 перебирает не правила, а нетерминалы, поэтому p в оценке времени заменяется на m. □

Следствие 1. *Проблемы R-тотальной и M-тотальной выводимости фиксированного слова w в данной неукорачивающей грамматике G для данного мультимножества правил R или данного мультимножества нетерминалов M принадлежат классу NP.*

В следующем разделе мы покажем, что если слово w фиксировано и содержит хотя бы два символа, то обе проблемы тотальной выводимости оказываются NP-полными. Для слов длины 1 справедлив следующий результат.

Теорема 3. *Проблемы R-тотальной и M-тотальной выводимости данного слова w длины 1 в данной неукорачивающей грамматике G для данного мультимножества правил R или данного мультимножества нетерминалов M принадлежат классу P.*

Доказательство. Поскольку длина w равна 1, а грамматика неукорачивающая, то в выводе могут использоваться только цепные правила, то есть правила вида $A \to B$, где A и B — нетерминалы, а также одно правило вида $A \to w$. Если грамматика содержит хотя бы одно другое правило, то тотального вывода не существует. Рассмотрим грамматику $G = (N, \Sigma, P, S)$ без «лишних» правил и построим по ней ориентированный граф $G' = (V, E)$ следующим образом. Вершинами графа являются нетерминалы грамматики и слово w: $V = N \cup \{w\}$. Рёбра графа соответствуют правилам: $(u, v) \in E$ тогда и только тогда, когда $u \to v \in P$. Непосредственно из

определения графа G' следует, что существование R-тотального вывода эквивалентно существованию пути из S в w, который проходит через каждое ребро по меньшей мере столько раз, какова кратность соответствующего правила в R. Чтобы найти такой путь, выделим в графе сильно связные компоненты и построим его граф компонент $G^K = (V^K, E^K)$. Эта задача решается стандартным алгоритмом за время $O(|V| + |E|)$ (см. [1]). Отметим в графе G^K все вершины, содержащие хотя бы одно ребро графа G, для которого соответствующее правило имеет ненулевую кратность, а также отметим все рёбра, соответствующие правилам ненулевой кратности. Пусть K_S — компонента, содержащая S, и пусть Если путь попал в некоторую компоненту, то он может посетить каждое её ребро сколь угодно много раз, поскольку все вершины взаимно достижимы. Поэтому нужно найти путь из в содержащий все отмеченные вершины $K_w = \{w\}$. и рёбра. Если приписать каждому отмеченному элементу стоимость 1, а всем остальным — стоимость 0, то задача сведётся к поиску самого длинного пути в ациклическом графе. Эта задача также решается стандартным алгоритмом за время $O(|V| + |E|)$ (см. [1]). Поэтому общее время работы составляет $O(|V| + |E|) = O(m + p)$, где, как и ранее, $m = |N|$, $p = |P|$.

Задача M -тотальной выводимости решается аналогичным методом, но единичные стоимости приписываются тем сильно связным компонентам, которые содержат хотя бы один нетерминал с ненулевой кратностью. □

3. NP-полнота проблемы тотальной выводимости

Теперь исследуем нижнюю границу сложности тотальной выводимости. Сначала докажем, что проблема M -тотальной выводимости фиксированного слова является NP-полной, даже если $M = N$.

Теорема 4. *Пусть Σ — произвольный алфавит, и пусть $w \in \Sigma^*$ — произвольное фиксированное слово длины не меньше 2. Проблема N-тотальной выводимости слова w в данной неукорачивающей грамматике $G = (N, \Sigma, P, S)$ является NP-полной.*

Доказательство. В силу следствия 1 нужно доказать только NP-трудность.

Мы сведём задачу о вершинном покрытии к задаче тотальной выводимости. Пусть $G = (V, E)$ — неориентированный граф с n вершинами $v_1, \ldots, v_n$ и m рёбрами $e_1, \ldots, e_m$. Обозначим через r_i количество соседей вершины v_i, а через $e_{i,j}$ — ребро, соединяющее вершину v_i с её j-м соседом в порядке увеличения номеров вершин, так что $e_{i,1}, e_{i,2}, \ldots, e_{i,r_i}$ — все рёбра, выходящие из v_i. Построим по графу G неукорачивающую грамматику $G' = (N, \Sigma, P, S)$. Множество нетерминалов определяется как

$$N = \{S, B\} \cup \{A_{i,j} : 1 \le i \le n, 1 \le j \le k\} \cup \{E_i : 1 \le i \le m\}.$$

Для нетерминалов E_i мы сохраним те же обозначения, что и для рёбер. Например, если $e_i = e_{j,k}$, то есть i-е ребро соединяет вершину v_j с её k-м соседом, то мы будем также считать, что $E_i = E_{j,k}$. Множество P содержит следующие правила:

$$\begin{aligned}
&S \to E_{i,1} A_{i,1}, && 1 \le i \le n,\\
&E_{i,j} A_{i,l} \to E_{i,j+1} A_{i,l}, && 1 \le i \le n, 1 \le j \le r_i - 1, 1 \le l \le k,\\
&E_{i,r_i} A_{i,l} \to E_{j,1} A_{j,l+1}, && 1 \le i \le n-1, i+1 \le j \le n, 1 \le l \le k-1,\\
&E_{i,r_i} A_{i,k} \to A_{1,1} B, && 1 \le i \le n,\\
&A_{i,j} B \to A_{i,j+1} B, && 1 \le i \le n, 1 \le j \le k-1,\\
&A_{i,k} B \to A_{i+1,1} B, && 1 \le i \le n-1,\\
&A_{n,k} B \to w.
\end{aligned}$$

Множество N содержит $nk+m+2$ нетерминала, в число правил равно

$$n+k\sum_{i=1}^{n}(r_i-1)+(k-1)\sum_{i=1}^{n-1}(n-i)+n+n(k-1)+(n-1)+1=3n+2km-kn+\frac{1}{2}(k-1)n(n+1),$$

поэтому построение грамматики может быть выполнено за полиномиальное время.

В нетерминалах $A_{i,j}$ первый индекс соответствует номеру вершины, а второй — количеству найденных вершин покрытия. Вывод слова w состоит из двух частей. Первая часть начинается с применения правила вида $S\to E_{i,1}A_{i,1}$. Дальше в выводе перебираются нетерминалы $E_{i,1},\ldots,E_{i,r_i}$, которые соответствуют рёбрам, выходящим из v_i. После посещения последнего такого нетерминала грамматика порождает строку вида $E_{i',1}A_{i',2}$ и посещает нетерминалы, которые соответствуют рёбрам, выходящим из вершины $v_{i'}$. Процесс продолжается до тех пор, пока не получится нетерминал вида $A_{i,k}$ со вторым индексом k. После этого начинается вторая часть вывода. Грамматика порождает слово $A_{1,1}B$, перебирает все нетерминалы $A_{i,j}$ и завершает вывод, порождая w. Если в первой части вывода были использованы нетерминалы $A_{i_1,1},\ldots,A_{i_k,k}$, то вывод посещает в точности те нетерминалы $E_{p,q}$, которые соответствуют рёбрам, инцидентным вершинам с номерами $i_1,\ldots,i_k$. Это и значит, что в графе G существует вершинное покрытие мощности не более k тогда и только тогда, когда существует N-тотальный вывод слова w в грамматике G'. □

Теперь мы докажем, что проблема R-тотальной выводимости фиксированного слова для множества правил R также является NP-полной.

Теорема 5. *Пусть Σ — произвольный алфавит, и пусть $w\in\Sigma^*$ — произвольное фиксированное слово длины не меньше 2. Проблема R-тотальной выводимости слова w в данной неукорачивающей грамматике G для данного множества правил R является NP-полной.*

Доказательство. Сводимость аналогична сводимости из теоремы 4. Построим по графу $G=(V,E)$ неукорачивающую грамматику $G'=(N,\Sigma,P,S)$. Множество нетерминалов определяется как

$$N=\{S\}\cup\{A_{i,j}:1\le i\le n,1\le j\le k\}\cup\{E_i,F_i:1\le i\le m\},$$

а множество P содержит следующие правила:

$$\begin{aligned}
&S\to E_{i,1}A_{i,1}, && 1\le i\le n,\\
&E_i\to F_i, && 1\le i\le m,\\
&F_{i,j}A_{i,l}\to E_{i,j+1}A_{i,l}, && 1\le i\le n,1\le j\le r_i-1,1\le l\le k,\\
&F_{i,r_i}A_{i,l}\to E_{j,1}A_{j,l+1}, && 1\le i\le n-1,i+1\le j\le n,1\le l\le k-1,\\
&F_{i,r_i}A_{i,k}\to w, && 1\le i\le n.
\end{aligned}$$

В качестве множества R мы возьмём $R=\{E_i\to F_i:1\le i\le m\}$.

В отличие от грамматики из теоремы 4, новая грамматика не содержит правил с B для посещения всех нетерминалов, поэтому вывод слова w завершается применением правила вида $F_{i,r_i}A_{i,k}\to w$. Другое отличие состоит в том, что применение одного правила $E_{i,j}A_{i,l}\to E_{i,j+1}A_{i,l}$ заменяется на участок из двух правил $E_{i,j}A_{i,l}\Rightarrow F_{i,j}A_{i,l}\Rightarrow E_{i,j+1}A_{i,l}$. Как и в доказательстве теоремы 4 устанавливается, что в графе G существует вершинное покрытие мощности не больше k тогда и только тогда, когда существует вывод слова w, содержащий все нетерминалы E_i. Но после получения слова с нетерминалом E_i обязательно должно применяться правило $E_i\to F_i$, поскольку другие правила не содержат E_i в левых частях. Следовательно, в графе G существует вершинное покрытие мощности не более k тогда и только тогда, когда существует R-тотальный вывод слова w. □

Несколько усложнив конструкцию, можно доказать аналогичные результаты и для контекстно-зависимых грамматик.

Теорема 6. *Проблемы R-тотальной и N-тотальной выводимости фиксированного слова w длины не меньше 2 в данной контекстно-зависимой грамматике G для данного множества правил R являются NP-полными.*

Доказательство. Нужно стандартным способом перестроить неукорачивающие грамматики в контекстно-зависимые (см. [5]). Для этого нужно заменить каждое правило вида $AB \to CD$ на правила $AB \to AB'$, $AB' \to CB'$, $CB' \to CD$, где B' — новый нетерминал, уникальный для каждого правила. Аналогично, если $w = av$, где $a \in \Sigma$, то правило вида $AB \to w$ заменяется на правила $AB \to aB$, $aB \to av$. Поскольку это преобразование не затрагивает правила вида $E_i \to F_i$, то доказательство теоремы 5 полностью сохраняется. Однако некоторые из новых нетерминалов могут не встречаться в выводе, который соответствует выводу, содержащему все нетерминалы исходной грамматики. Пусть $X_1,\ldots,X_p$ — все новые нетерминалы. Чтобы учесть их, мы заменим в доказательстве теоремы 4 правило $A_{n,k}B \to w$ на $A_{n_k}B \to X_1B$, а также добавим правила

$$X_iB \to X_{i+1}B, \quad 1 \le i \le p-1,$$
$$X_pB \to aB,$$
$$aB \to av.$$

Эти правила добавляются в конец любого вывода слова w и обеспечивают использование новых нетерминалов. □

В теоремах 4–6 установлена NP полнота проблемы тотальной выводимости для частного случая, когда мультимножество правил является обычным множеством, а мультимножество нетерминалов совпадает с множеством N. Поскольку по следствию 1 эти проблемы принадлежат классу NP и в том случае, когда R и M являются мультимножествами, то мы получаем следующий результат.

Следствие 2. *Проблемы R-тотальной и M-тотальной выводимости фиксированного слова w длины не меньше 2 в данной неукорачивающей или контекстно-зависимой грамматике G для данного мультимножества правил R или данного мультимножества нетерминалов M являются NP-полными.*

Заключение

В работе была исследована проблема тотальной выводимости фиксированного слова в неукорачивающих и контекстно-зависимых грамматиках для мультимножества правил или нетерминалов. Было установлено, что оба варианта проблемы являются NP-полными для любого слова длины не менее 2, а также что они полиномиально разрешимы для слов длины 1. Дальнейшие исследования могут включать поиск широких классов грамматик, для которых проблема тотальной выводимости разрешима за полиномиальное время.

Литература

1. Алгоритмы: построение и анализ / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. – 2-е изд. – М. : Вильямс, 2011. – 1296 с.

2. *Ахо А.* Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М. : Мир, 1979. – 536 с.

3. *Гэри М.* Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М. : Мир, 1982. – 416 с.

4. Сложность исчислений Ламбека с модальностями и тотальной выводимости в грамматиках / С. М. Дудаков, Б. Н. Карлов, С. Л. Кузнецов, Е. М. Фофанова // Алгебра и логика. – 2021. – Т. 60, №5. – С. 471–496.

5. *Shallit J.* A second course in formal languages and automata theory / J. Shallit. – Cambridge : Cambridge University Press, 2008. – XII, 240 p.

6. *Valiant L. G.* General context-free recognition in less than cubic time / L. G. Valiant // Journal of Computer and System Sciences. – 1975. – Vol. 10. – P. 308–315.

УДК 517.977

ПРОБЛЕМЫ ПОСТАВОК И РАСПРЕДЕЛЕНИЯ КОМПЛЕКТУЮЩИХ ДЛЯ ЭЛЕКТРОТЕХНИЧЕСКОГО ОБОРУДОВАНИЯ

Ю. А. Клименко, А. П. Преображенский, Ю. П. Преображенский

Воронежский институт высоких технологий

Аннотация. В статье затронута проблема работы механизма снабжения авторизированных центров запасными частями электротехнического оборудования потребителей. Это зависит от четкой работы складской сети по определенному алгоритму с учетом спецификации товаров. Нами поведен анализ поставщиков России по выполнению заказов интернет-сервисами. Выявлены основные недостатки системы снабжения и рассмотрены различные стратегии по управлению резервами запасных частей. Предложена система разделения параметров товаров на отдельные группы. Определены пути выхода из ситуации определения наименований товаров при неисправности электротехнического оборудования.

Ключевые слова: система, снабжение, неисправность, спецификация, склад, запас, стратегия, параметр, группа, наименование, оборудование, электротехнический.

Введение

На сегодняшний день в Центрально-Европейской части России, надежно работает механизм обеспечения авторизованных сервисных центров (АСЦ) запасными частями. Целью этой системы — обеспечение пользователей/потребителей электротехнического оборудования запасными частями и аксессуарами через сеть региональных складов. Внедрение в рыночную систему проходило с использованием накопленного зарубежного опыта, и сейчас эта система получила большое сходство с Европейскими системами сервиса. На рынке электротехнического оборудования появились представители всемирно известных брендов, которые обеспечили поставку комплектующих и запасных частей по схеме, удобной производителю с учетом специфики Российской экономики.

Складские сети стали более требовательны к поддержанию имиджа торговой марки, а так же переориентировались на максимальное удовлетворение потребностей клиентов.

1. Анализ ситуации на рынке снабжения по обеспечению запасными частями

На сегодняшний момент, перемещение запасных частей в складских сетях, как у нас, так и за рубежом, происходит следующим образом: фирма-производитель электротехнического оборудования и запасных частей имеет центральный склад хранения, на котором хранится основное количество ассортимента запасных частей (до 85 %). Это количество обеспечивает потребность всего парка обслуживаемых электроустановок. Производитель поддерживает сеть региональных складов, на которых хранится значительное количество запасных частей, при этом по каждой запасной части создается запас, необходимый для бесперебойного функционирования сервис-партнеров региона в течение двух недель. Размер регионального склада зависит от размера региона и парка электроустановок, находящихся в эксплуатации.

Эти склады обеспечивают АСЦ, а также более мелкие организации — сервисные центры (СЦ), осуществляющие сервис и пусковую наладку оборудования электроустановок.

Все они имеют официальный статус сервис-партнера (СП) предприятия-изготовителя.

При отсутствии на региональном складе необходимой запчасти, осуществляется ее заказ с федерального склада. Применение интернет-сервисов позволяет получить информацию о

наличии этой запчасти на федеральном складе. В случае отсутствия запчасти на федеральном складе она заказывается с центрального склада фирмы-изготовителя.

Если запасной части нет на центральном складе, она изготавливается фирмой-производителем.

Такой уровень взаимодействия возможен только при заказе оригинальных запасных частей. Оригинальной считается такая запасная часть, которая произведена фирмой-изготовителем отопительного оборудования или произведена другим предприятием, которое имеет законное право производства модулей, элементов, деталей под торговой маркой фирмы-изготовителя электротехнического оборудования.

Недостатками такой системы снабжения АСЦ являются:

1. Достаточно длительные сроки поставки заказа запасных частей с центрального склада (от 2-х до 12 недель).

2. Производство запасных частей кустарным способом. Неоригинальные запчасти имеют более низкую стоимость, и как следствие, более низкое качество.

3. Далеко не дешёвая оригинальная продукция (производитель оставляет за собой право неразглашения ценообразования).

4. Не всем АСЦ и СЦ удобен график работы федеральных и региональных складов.

Работа СП в большинстве случаев зависит от региональных складов. Официальные сервисные центры обязаны осуществлять заказ только у партнеров производителя. Использование оригинальных запчастей регламентировано договором сервисного обслуживания производителя и СЦ.

Сервисные центры, осуществляющие ремонт электротехнического оборудования и склады всех уровней, стараются отладить методику прогнозирования товарных запасов. Цель методики состоит в анализе товарооборота, расчета оптимальных запасов, прогнозировании движения запасных частей и оборудования и т.д.

Как правило, специалисты решают задачи, которые касаются двух аспектов логистики: если заказывать, то сколько и когда заказывать?

Крупные АСЦ и СЦ в рабочем процессе составляют и реализуют множество заказов. Всё это учитывается при формировании заказа на склад, а точнее, для его пополнения.

Чтобы максимально развернуто отобразить «картину» взаимодействия системы, нужно определить наиболее важные технико-экономические показатели. Эти показатели могут быть следующими:

- Общее количество запасных частей на складе.
- Детали, которые часто выходят из строя.
- Условия эксплуатации оборудования.
- Интенсивность использования оборудования.
- Не стоит забывать про человеческий фактор.

Нужно держать на вооружении, что имеют место быть сезонные «падения» и «взлеты» спроса на запасные части. Решить эти проблемы позволит создание научно обоснованной системы управления запасами запасных частей на станции технического обслуживания.

Таким образом, исследования, направленные на повышение уровня материально-технического снабжения являются актуальными.

Широко применяемые на практике стратегии управления запасами приведены ниже:

- стратегия с критическим уровнем. При ее использовании, когда запасы снижаются до критического уровня n — производится пополнение.
- периодическая стратегия. При ее применении пополнение запасов происходит в заранее заданные промежутки.

В текущих стратегиях запас восстанавливается до постоянного объема V, или до максимального N. В итоге формируются последующие комбинации, определяющие стратегии управления запасами:

• $< T, N >$ — периодическая до максимального на данном этапе производственной программы по ремонту и техническому обслуживанию.

• $< T, V >$ — периодическая до постоянного объема;

• $< n, N >$ — критическая до максимального уровня;

• $< n, V >$ — критическая до постоянного объема;

Стратегия $< T, N >$ является наиболее гибкой. Она требует от поставщиков гарантий, но при этом предоставляет возможность оперативно реагировать на динамику спроса. Стратегия $< T, V >$ весьма работоспособна в условиях стабильного спроса и гарантированного снабжения. Процесс управления запасами в этой ситуации считается стационарным. Он более продуктивен в реализации массовых производственных работ по ТО и ремонту отопительного оборудования для крупных стабильно работающих предприятий. Для стратегии $< n, N >$ и $< n, V >$ требуется оперативный контроль наличия гарантированных, в текущее время, источников поставки всей номенклатуры запчастей и постоянный учет расхода запасов. По приведенным причинам, данные стратегии в практическом применении являются достаточно сложными и трудоемкими.

2. Имитационная модель системы управления снабжением запасными частями

Под имитационной моделью (ИМ) понимается некая система, целью которой является управление запасными деталями отопительного оборудования, при этом выполняя задачи формирования последовательности тех или иных событий в моменты времени $t_1, t_2, ..., t_i, ..., t_N$, для каждого из которых задается преобразование состояния модели $Y_i \to Y_{i+1}$, $i = 1, 2, ..., N-1$, где $Y_i = (y_1^i, y_1^i, ..., y_n^i)$ — является вектором состояния модели. Большим преимуществом данной модели является то, что она включает в себя учет внешних факторов $X_i = (x_1^i, x_1^i, ..., x_n^i)$, влияющих на потребность покупателей в тех или иных деталях, также в модель включен вектор управления $U_i = (u_1^i, u_1^i, ..., u_n^i)$.

Определить преобразование в рекуррентной форме может только оператор, указав задание F:

$$Y_{i+1} = F\left(Y_i, X_{x+1}, U_{x+1}\right), \quad i = 1, 2, ..., N-1. \tag{1}$$

Стоит отметить, что рекуррентная схема имитационного алгоритма может быть расширена, при необходимости учета в нем латентных параметров, которые является в свою очередь неконтролируемыми $E_i = (e_1^i, e_2^i, ..., e_q^i)$.

$$Y_{i+1} = F^*\left(Y_i, X_{x+1}, U_{x+1}, E_{i+1}\right), \quad i = 1, 2, ..., N-1. \tag{2}$$

Основными компонентами при проведении эксперимента и разработки модели являются датчики случайных чисел и модель управляемого объекта и др.

Рассмотренные компоненты подразумевают под собой:

• количество поступающих заказов на деталь;

• необходимые характеристики для качественного размещения полученных деталей от разных поставщиков, так же включает;

• стратегию по эффективному управлению запасами на складе и другими компонентами.

Для удобства разобьем все необходимые параметры на группы:

• 1-я группа — $Z = (Z_1, Z_2, ..., Z_k)$ включает входные параметры, которые не подлежат контролю;

• 2-я группа — $U = (U_1, U_2, ..., U_k)$ включает входные параметры подлежащие контролю;

• 3-я группа — $V = (V_1, V_2, ..., V_k)$ включает входные параметры, подлежащие как управлению, так и контролю;

• 4-я группа — $Y = (Y_1, Y_2, ..., Y_k)$ включает выходные параметры.

Отметим, что в рамках проведения эксперимента, с целью анализа поведения модели, входные параметры, которые подлежат контролю, имеют одну общую совокупность $X = (V, U)$. Что касается схемы проведения эксперимента, то она имеет вид Z и Y — которые являются стохастическими величинами, а $Y = \varphi(X, Z)$ является детерминированной функцией $M(X|M) = \int \varphi(X, Z)\omega(Z|X)dz$, в отношении которой $\omega(Z|X)$ является плотностью случайного вектора Z при заданных величинах вектора X.

Из всего выше сказанного, можно сделать вывод о том, что для построения имитационной модели необходимо иметь формальные алгоритмы и модели:

1. Количество заказов на определенную деталь с учетом характеристик, и возможностей изменения ситуации внешней среды, которая может повлиять на поток заказов. Эти параметры являются — контролируемыми неуправляемыми.

2. Стратегия для эффективного управления запасами деталей на складе, с учетом их параметризации по количественным показателям, обладающими определенными методами управления решением, и выражены определенной спецификой. Эти параметры являются — контролируемыми управляемыми.

3. Несвоевременные поставки заказов от поставщиков, а так же резкое изменение ценовой политики от поставщиков на необходимые запасные детали. Эти параметры являются — неконтролируемыми.

4. Критерии оценки эффективности деятельности склада (как в целом, так и отдельно) по управлению запасными частями, выраженных в интегральных и локальных характеристиках. Эти параметры являются — выходными.

Далее разобьем номенклатуру на отдельные группы, это необходимо для проведения более детального анализа по расходу деталей на складе, деление номенклатуры будет проводиться на основании некой взаимосвязи между запасными деталями, т. е. в случаи замены одной детали необходимо будет провести замену и других, связанных с ней.

Для этого возьмем группу деталей для замены с учетом статистики наиболее частых неисправностей силовых трансформаторных подстанций и последующих действий по оформления документов после проведения ремонтных работ.

Например, при выходе из рабочего состояния силового понижающего трансформатора комплектной трансформаторной подстанции 10/0,4 кВ часто происходит «отгорание» шпилек крепления электрических соединений входных (10 кВ) и выходных (0,4 кВ) токопроводящих шин. Чтобы устранить неисправность необходимо: произвести замену шпилек соединительных токопроводящих контактных соединений токопроводящих элементов. При этом, также, необходимо провести замену резиновых изолирующих и герметизирующих прокладок крышки емкости силового трансформатора и изоляторов крепления входных и выходных токоведущих шин, что повлечет за собой, в последствии, при окончании ремонтных работ проведение электрических испытаний трансформаторного масла силового трансформатора и электрических испытаний параметров самого силового трансформатора.

Получаем цепь последствий замены деталей при ремонте электрооборудования: изолирующие прокладки крышки емкости силового трансформатора $\Rightarrow$ герметизирующие прокладки крышки емкости силового трансформатора $\Rightarrow$ изолирующие прокладки изоляторов крепления входных и выходных токоведущих шин $\Rightarrow$ герметизирующие прокладки изоляторов крепления входных и выходных токоведущих шин $\Rightarrow$ шпильки крепления электрических соединений входных (10 кВ) и выходных (0,4 кВ) токопроводящих шин $\Rightarrow$ электрические испытания трансформаторного масла силового трансформатора $\Rightarrow$ электрические испытания параметров силового трансформатора $\Rightarrow$ оформление актов проведения электрических испытаний силового трансформатора электротехнической лабораторией $\Rightarrow$ ввод в эксплуатацию трансформаторной подстанции после ремонта.

Заключение

В результате проведенного анализа проработана формальная модель оптимизации отдельных управляемых параметров для описания и оценки влияния различных факторов на своевременную поставку запасных частей к электротехническому оборудованию.

Литература

1. *Тухватуллин Б. Т., Левченко Д. В., Стебайлов М. Ю.* Формирование структурно-энергетических параметров рабочих поверхностей высокоточных деталей при эксплуатации автомобильного транспорта // International Journal of Advanced Studies. – 2019. – Т. 9, № 3. – С. 64–70.

2. *Лысанов Д. М., Пономаренко Н. А.* Количественная оценка уровня качества оборудования // International Journal of Advanced Studies. – 2018. – Т. 8, № 4–2. – С. 56–61.

3. *Лысанов Д. М., Бикмухаметова Л. Т.* Анализ показателей качества и конкурентоспособности оборудования // International Journal of Advanced Studies. – 2018. – Т. 8, № 4–2. – С. 50–55.

4. *Гуськова Л. Б.* О построении автоматизированного рабочего места менеджера // Успехи современного естествознания. – 2012. – № 6. – С. 106.

5. *Самойлова У. А.* О некоторых характеристиках управления предприятием // Вестник Воронежского института высоких технологий. – 2014. – № 12. – С. 176–179.

6. *Пеньков П. В.* Экспертные методы улучшения систем управления // Вестник Воронежского института высоких технологий. – 2012. – № 9. – С. 108–110.

7. *Максимов И. Б.* Принципы формирования автоматизированных рабочих мест // Вестник Воронежского института высоких технологий. – 2014. – № 12. – С. 130–135.

8. *Максимов И. Б.* Классификация автоматизированных рабочих мест // Вестник Воронежского института высоких технологий. – 2014. – № 12. – С. 127–129.

УДК 004.054

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ ПРОГНОЗИРОВАНИЯ ЛОКАЛЬНОЙ ЦЕЛОСТНОСТИ ЭЛЕМЕНТОВ ИНФРАСТРУКТУРЫ

П. М. Курчавов

МИРЭА – Российский технологический университет

Аннотация. Произведена аналитика необходимая для практической реализации программного прототипа, в основе которого лежит многокритериальная методика анализа элементов КИИ. Рассмотрена архитектура программного комплекса для произведения расчета и прогнозирования инфраструктурной целостности систем взаимодействующих объектов. Приведены первые результаты работы программного средства и анализ их точности.
Ключевые слова: целостность, информационная безопасность, субъект, критическая информационная инфраструктура, многокритериальный анализ, инфраструктурный деструктивизм, сверточные нейронные сети, python.

Вступление

Проблематика инфраструктурного деструктивизма, на сегодняшний день является актуальной. Исходя из 187 федеральному закону [1], следует, что предприятия, которые официально являются объектами критической информационной инфраструктуры (далее КИИ) функционируют в огромном количестве сфер, таких как — здравоохранение, наука, транспорт, связь, энергетика, банковская сфера и т. п. Такие предприятия должны функционировать бесперебойно и с минимальными потерями данных, в виду чего остро встает вопрос об обеспечении их стабильной работы.

При нарушении инфраструктурной целостности у системы может понизиться уровень безопасности, причем как отдельного элемента системы, так и всей системы [2]. Данная ситуация, в свою очередь будет увеличивать шансы развития инфраструктурного деструктивизма, который является фактом саморазрушение инфраструктуры субъекта [3]. В рамках обозначенной выше проблемы необходима анализ существующих проблем на объектах КИИ, формализация методик и их реализация для применения в существующих системах.

1. Проектирование программного комплекса для расчёта и прогнозирования инфраструктурной целостности систем взаимодействующих объектов

Для решения поставленной цели была проведена аналитика необходимая для практической реализации программного прототипа, в основе которого будет лежать методика, рассмотренная в предыдущем разделе. Предполагается, что программный комплекс должен быть представлен в виде следующей архитектуры (рис. 1), представленном на верхнем уровне декомпозиции.

На рис. 1 представлены основные составляющие программного комплекса.

• Модуль многокритериального анализа SAW основанный на методике использования метода SAW для оценки инфраструктурной целостности объекта КИИ [4]. Данный модуль представлен в виде декомпозированного элемента. Данный модель представляет собой набор скриптов, реализующих математические операции, описанные ранее в третьем разделе, а так же автоматически формирующий отчет в виде csv файла, для просмотра пользователем, и числовой ряд представляющий собой показатели целостности. Данный числовой ряд необходим для дальнейшей постобработки и прогнозирования ситуации в системе взаимодействующих объектов;

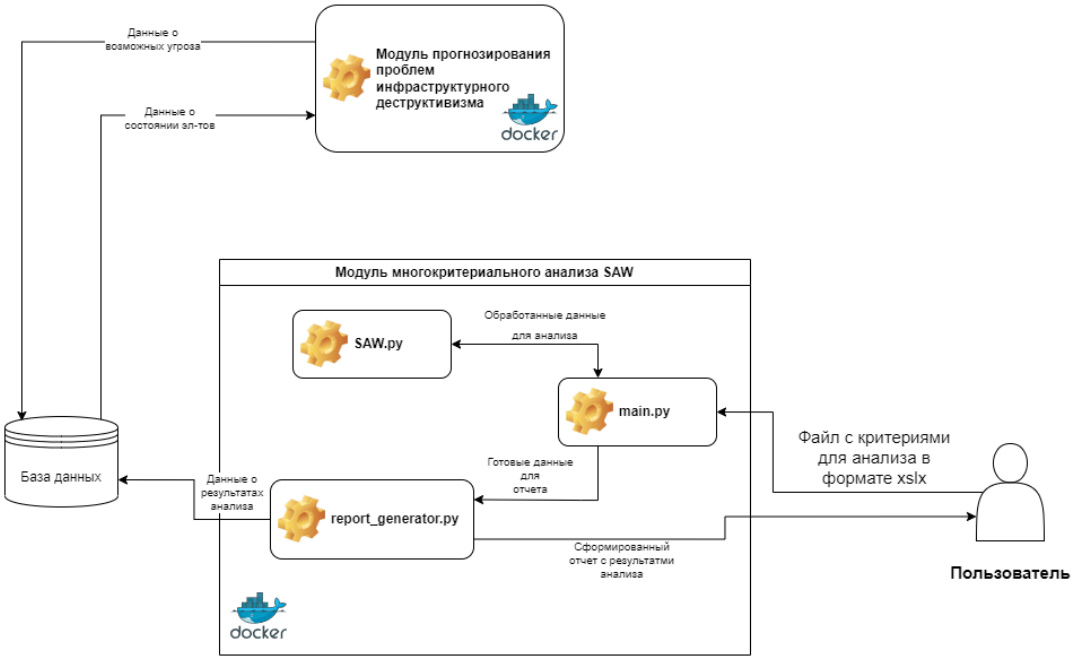


Рис. 1. Архитектура программного комплекса для расчёта и прогнозирования инфраструктурной целостности систем взаимодействующих объектов

• База данных, представляет собой реляционное хранилище данных для долгосрочного содержания промежуточных результатов работы модуля многокритериального анализа, для ведения дальнейшей статистики;

• Модуль прогнозирования проблем инфраструктурного деструктивизма — это экспериментальный модуль, в рамках которого предполагается использование нейронной сети, для прогнозирования движения числового ряда в положительную или отрицательную сторону в будущем исходя из входной выборки, сформированной и подготовленной заранее с помощью ранее описанных модулей.

2. Анализ практик и реализация программных прототипов

В рамках исследования, был разработан первый прототип Модуля многокритериального анализа SAW. Она представляет собой программу для ЭВМ — программная реализация метода многокритериального анализа SAW (с дополнениями в рамках тематики). Область применения программного продукта — анализ целостности субъектов критической информационной инфраструктуры. В соответствии с функциональным назначением включает следующие компоненты: основной модуль (main.py), необходимый для получения входных и промежуточных данных, модуль автоматизированного расчета по методу многокритериального анализа SAW, модуль генерации выходного файла с результатами расчетов.

Описанный выше модуль был реализован на языке python и предназначен в первую очередь для произведения итоговой оценки целостности на основании заранее подготовленных данных. В своей основе данное программное средство работает по схеме приведенной ниже (рис. 2).

В соответствии со всем вышеописанным, возникла идея использования моделей сверхточных нейронных сетей для прогнозирования временных рядов, состоящих из просчитанных, в рамках вышеописанной методики, усредненных оценок целостности объекта КИИ. В соответствии с чем, можно обозначить конкретную цель – исследование возможности использования моделей сверточных нейронных сетей, для решения выше обозначенного вопроса.

Рис. 2. Схема использования метода SAW для оценки инфраструктурной целостности объекта КИИ

Временной ряд — это совокупность значений какого-либо показателя за несколько последовательных моментов или периодов времени. В соответствии с чем, можно сделать вывод, что последовательность из значений усредненных оценок целостности объекта КИИ.

В рамках исследования было выявлено несколько типов нейронных сетей, которые могли бы подойти для решения поставленной цели, а именно прогнозирования временных рядов.

1. Рекуррентная нейронная сеть без сохранения состояния — это тип рекуррентной нейронной сети, в которой окна с данными передаются смешанно и могут пересекаться друг с другом. В конце каждого окна будет образовываться вектор состояния, который будет сброшен и не получит дальнейшего применения. Из-за данной особенности будет проблематично заставить рекуррентную нейронную сеть без сохранения состояния запомнить длительные паттерны временного ряда (максимум — на несколько дюжин временных шагов) [5].

2. В рекуррентной нейронной сети с сохранением состояния окна с данными должны передаваться упорядоченно и не пересекаться. На выходе одного окна будет образовываться вектор состояния, который не будет сброшен сразу. Он будет передан следующей итерации и будет сброшен в конце эпохи. Это позволит сети получить имеющий смысл вектор состояний в начале каждой итерации и запомнить более долгие паттерны. Она сложнее в реализации и обучение проходит намного дольше. [5]

3. Long Short-Term Memory — техника для предоставления возможности рекуррентной нейросети запоминать долгосрочные паттерны. LSTM нейросеть имеет более долгую память — способна запомнить паттерны, длиной более 100 временных шагов. Тем не менее, с более долгими паттернами справиться ей будет сложно [6].

4. Сверточные нейронные сети (CNN) могут быть крайне эффективны для решения задач на подобии классификации изображений, но также они эффективны при работе с последовательностями. В данных нейросетях происходит сдвиг фильтра по временному ряду, получая взвешенную сумму и применяя к ней функцию активации. Сверточные нейросети не обладают памятью, однако при составлении многих слоев получается, что верхние слои видят большую часть последовательности, поэтому сверточные нейросети могут охватывать длительные паттерны [5].

Исходя из вышеприведенных характеристик, было решено остановиться, в рамках данного исследования, на сверточной нейронной сети (далее CNN), т. к. она более всего удовлетворяет поставленной цели, а именно из-за их способности охватывать длительные паттерны, что, потенциально, и нужно для реализации прогнозирования временных рядов, состоящих из просчитанных с помощью многокритериального анализа усредненных оценок целостности объекта КИИ. Это обосновывается тем, что мониторинг объектов КИИ будет происходить в реальном времени, а значит ряд может получится достаточно объемным.

Реализацию необходимо разделить на два этапа:

1. Подготовка данных для сверточной нейронной сети;

2. Использование модели CNN для решения задачи прогнозирования временного ряда.

Было принято решение, формировать обучающую выборку, с помощью случайно сгенерированного множества, в которое будет входить с одна сотня значений, где каждое значения будет равно числу от 0 до 100, с округлением до одного знака после запятой (именно в таком виде представлены усредненные оценки целостности объекта КИИ).

Ниже представлен кода, генерирующий обучающую выборку:

```
import random
from numpy import array
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D

def prepare_dataset(sequence, n_steps):
    X, y = list(), list()
    for i in range(len(sequence)):
        # find the end of this pattern
        end_ix = i + n_steps
        # check if we are beyond the sequence
        if end_ix > len(sequence)-1:
            break
        # gather input and output parts of the pattern
        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)

raw_seq_test = []
n = 0
while n < 100:
    raw_seq_test.append(round(random.uniform(0, 100), 3))
    n += 1

raw_seq_high = sorted(raw_seq_test)
raw_seq_low = sorted(raw_seq_test, reverse=True)
raw_seq_random = raw_seq_test

a = 0
start_arrays = [raw_seq_high, raw_seq_low, raw_seq_random]

n_steps = 10

for raw_seq in start_arrays:
    a += 1
    X, y = prepare_dataset(raw_seq, n_steps)
```

В рамках исследования предполагается проверить CNN на трех вариантах датасета. В первом случае, временной ряд формируется по возрастанию, во втором случае по убыванию, а в третьем, разброс величин по ряду будет случайным (что наиболее соответствует реальности),

для этого и было прописано три массива, которые представляют собой 3 разных варианта представления данных.

В результате работы скрипта, будет сформирована обучающая выборка для модели CNN. Пример результата представлен ниже (рис. 3).

```
 [0.944  2.236  2.847  3.084  4.247  4.877  5.901  6.284  6.535  6.842]  9.224
 [2.236  2.847  3.084  4.247  4.877  5.901  6.284  6.535  6.842  9.224] 10.059
 [2.847  3.084  4.247  4.877  5.901  6.284  6.535  6.842  9.224 10.059] 11.311
 [3.084  4.247  4.877  5.901  6.284  6.535  6.842  9.224 10.059 11.311] 15.001
 [4.247  4.877  5.901  6.284  6.535  6.842  9.224 10.059 11.311 15.001] 15.146
 [4.877  5.901  6.284  6.535  6.842  9.224 10.059 11.311 15.001 15.146] 17.126
 [5.901  6.284  6.535  6.842  9.224 10.059 11.311 15.001 15.146 17.126] 17.191
 [6.284  6.535  6.842  9.224 10.059 11.311 15.001 15.146 17.126 17.191]  17.35
 [6.535  6.842  9.224 10.059 11.311 15.001 15.146 17.126 17.191  17.35]  21.22
 [6.842  9.224 10.059 11.311 15.001 15.146 17.126 17.191  17.35  21.22] 21.561
 [9.224 10.059 11.311 15.001 15.146 17.126 17.191  17.35  21.22 21.561] 21.911
[10.059 11.311 15.001 15.146 17.126 17.191  17.35  21.22 21.561 21.911] 22.654
[11.311 15.001 15.146 17.126 17.191  17.35  21.22 21.561 21.911 22.654] 22.721
[15.001 15.146 17.126 17.191  17.35  21.22 21.561 21.911 22.654 22.721]  23.21
[15.146 17.126 17.191  17.35  21.22 21.561 21.911 22.654 22.721  23.21] 23.342
[17.126 17.191  17.35  21.22 21.561 21.911 22.654 22.721  23.21 23.342] 23.476
[17.191  17.35  21.22 21.561 21.911 22.654 22.721  23.21 23.342 23.476] 23.843
 [17.35  21.22 21.561 21.911 22.654 22.721  23.21 23.342 23.476 23.843] 24.486
 [21.22 21.561 21.911 22.654 22.721  23.21 23.342 23.476 23.843 24.486] 26.021
[21.561 21.911 22.654 22.721  23.21 23.342 23.476 23.843 24.486 26.021] 26.205
[21.911 22.654 22.721  23.21 23.342 23.476 23.843 24.486 26.021 26.205] 26.961
[22.654 22.721  23.21 23.342 23.476 23.843 24.486 26.021 26.205 26.961] 28.611
[22.721  23.21 23.342 23.476 23.843 24.486 26.021 26.205 26.961 28.611] 29.328
```

Рис. 3. Пример сгенерированной обучающей выборки

После того, как была сформированна обучающая выборка можно переходить непосредственно к построению модели. В рамках работы используется одномерный CNN - это модель CNN, которая имеет сверточный скрытый слой, который работает над одномерной последовательностью. За этим следует, возможно, второй сверточный слой в некоторых случаях, например, очень длинные входные последовательности, и затем объединяющий слой, задача которого состоит в том, чтобы перевести вывод сверточного слоя в наиболее заметные элементы.

За сверточным и объединяющим слоями следует плотный полностью связанный слой, который интерпретирует особенности, извлекаемые сверточной частью модели. Сглаженный слой используется между сверточными слоями и плотным слоем, чтобы свести карты объектов к одному одномерному вектору.

Одномерная модель CNN для одномерного прогнозирования временных рядов определяется следующим образом (рис. 4).

В модели используется оптимизатор «Адам версия стохастического градиентного спуска» с использованием среднеквадратичной ошибки, или «MSELoss» [7]. Последняя строка из участка листинга выше, используется для того, чтобы обучить модель, воспользовавшись обучающей выборкой сформированной ранее.

После всего вышеописанного можно начинать тестировать модель. Входными данными станет также случайно сгенерированный временной ряд, который сформирован по тем же принципам, что и данные для формирования обучающей выборки.

Результаты ее работы представлены на рис. 5.

```
model = Sequential()
model.add(Conv1D(filters=64, kernel size=2, activation='relu'
input shape=(n_steps, n_features)))
model.add(MaxPoolinglD(pool_size=2))
model.add(Flatten())
model.add(Dense(50, activation='relu'))
model.add(Dense(l))
model.compile(optimizer='adam', loss='mse')
model.fit(X, y, epochs=1000, verbose=0)
```

Рис. 4. Одномерная модель CNN для одномерного прогнозирования временных рядов

```
======== HIGH ========
[5.571, 9.667, 16.619, 23.981, 56.459, 57.507, 79.016, 82.161, 90.418, 90.658]
[[78.11318]]
========= LOW ========
[80.959, 79.419, 77.269, 73.947, 71.399, 58.075, 51.571, 17.926, 9.797, 3.633]
[[8.136717]]
======= RANDOM =======
[21.407, 29.036, 6.525, 58.278, 84.074, 94.56, 12.652, 86.141, 72.467, 29.767]
[[52.352158]]
```

Рис. 5. Результат работы CNN для работы с тремя представлениям временных рядов

Заключение

В общем и целом, можно сделать вывод, что CNN достаточно интересна для проработки прогнозирования временных рядов. Однако в ходе исследования были выявлены явные недостатки, которые, возможно будут исправлены, если более детально проработать модель.

1.Модель не всегда очевидно прогнозирует продолжение временного ряда, что может стать большой проблемой, в рамках поставленной цели, а именно оценки инфраструктурной целостности субъекта КИИ;

2.При обработке временного ряда с неотсортированными данными, порой модель прогнозировала числа, которые, в рамках поставленной цели, являются нереалистичными (пример 112,323 или –37,456), однако это недостаток явно связан с тем, что модель нужно более детально проработать.

Однако в общем, данная модель достаточно хорошо подходит для использования в контексте исследований на предмет анализа инфраструктурной целостности субъекта КИИ. В целом использование механизма для прогнозирования результатов, звучит очень многообещающе и полезно. Впрочем, достаточно важным фактором является отсутствие реальных данных для обучающей выборке, что негативным образом влияет на все исследование.

Литература

1. О безопасности критической информационной инфраструктуры Российской Федерации // Федеральный закон от 26.07.2017 N 187-ФЗ;

2. Основы информационной безопасности. Часть 1: Виды угроз [Электронный ресурс]: Хабр URL: https://habr.com/ru/company/vps_house/blog/343110/;

3. *Максимова Е. А.* Инфраструктурный деструктивизм субъектов критической информационной инфраструктуры [монография] / Е. А. Максимова // Министерство науки и высшего образования Российской Федерации, МИРЭА-Российский технологический университет. – Москва : Издательство Волгоградского государственного университета. – 2021. – С. 170–180.

4. *Курчавов П. М., Максимова Е. А.* Многокритериальная оценка целостности субъекта критической информационной инфраструктуры // Прикаспийский журнал: Управление и высокие технологии. Учредители: Астраханский государственный университет им. В. Н. Татищева.

5. *Дауб И. С.* Обзор методов прогнозирования временных рядов с помощью искусственных нейронных сетей // StudNet. – 2020. – № 10. – URL: https://cyberleninka.ru/article/n/obzor-metodov-prognozirovaniya-vremennyh-ryadov-s-pomoschyu-iskusstvennyh-neyronnyh-setey (дата обращения: 27.11.2024).;

6. LSTM – сети долгой краткосрочной памяти [Электронный ресурс]: Habr URL: https://habr.com/ru/companies/wunderfund/articles/331310/;

7. Стохастический градиентный спуск. Часть 3 [Электронный ресурс]: dmitrymakarov.ru URL: https://www.dmitrymakarov.ru/opt/sgd-03/;

8. Рекомендации по оценке показателей критериев экономической значимости объектов КИИ от ФСТЭК [Электронный ресурс]: SecurityLab.ru. URL: https://www.securitylab.ru/blog/personal/valerykomarov/350276.php

9. Логические операции и их свойства [Электронный ресурс]: Справочник Автор24 URL: https://spravochnick.ru/informatika/algebra_logiki_logika_kak_nauka/logicheskie_operacii_i_ih_svoystva/

10. КИИ: обзор нормативной базы ФСТЭК России [Электронный ресурс]: Kaspersky ICS CERT URL: https://ics-cert.kaspersky.ru/publications/reports/2018/06/25/obzor-normativnoy-bazy-fstek-rossii/

УДК 002:001(470)

ОНТОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЛОГИСТИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ

В. И. Меденников

ФИЦ «Информатика и управление» РАН

Аннотация. Целью работы является разработка научно-обоснованного онтологического описания логистической деятельности в стране, актуальность которого проявилась как следствие сформированных основных принципов зрелой цифровой трансформации мирового развития. Указанные принципы вызвали разработку механизмов интеграции информационных ресурсов и систем не только внутри одной отрасли, но и большинства смежных отраслей, для чего должно быть осуществлено онтологическое моделирование их предметных пространств, понятийное описание которых за многие годы существенно стало разниться. Особенно это касается логистики, создающей цепочки добавленной стоимости из значительного числа участников, включая многочисленных производителей комплектующих, поставщиков ресурсов и услуг, потребителей изделий и самих логистических компаний.

Ключевые слова: логистическая деятельность, онтологическое моделирование, цифровая платформа управления, интеграция информационных ресурсов, цифровые стандарты.

Введение

Глобализация мировой экономики привела к ситуации, когда в производстве значительного числа товаров принимает участие все большее количество компаний из различных стран. В результате даже возник термин «Цепочки добавленной стоимости». Технологии такого дробления бизнеса выдвигают все более жесткие требования информационной и алгоритмической совместимости используемого потока данных по всей цепочке. Однако недостаточный уровень состояния интеграционных технологий применяемых информационных систем организаций, участвующих в конкретных цепях поставок, неупорядоченное, хаотичное освоение новых возможностей Интернет-технологий вынуждает всех участников ориентироваться на самые отсталые из них, присущие некоторым из них. Можно сказать, что в цепочке проявляется своеобразный принцип Либиха, или закон ограничивающего фактора, открытый ещё в 1840г. немецким учёным Юстусом фон Либихом, правда, в области сельского хозяйства. Данный принцип гласит, что для любого организма (системы) критически значим тот фактор, который находится в минимальном значении. В результате проявления его в логистике возникает большое число посредников во всей цепочке создания добавленной стоимости.

Одним из инструментов интеграционных технологий является метод построения онтологического пространства знаний [1]. Онтологическое моделирование применяется для описания и интеграции знаний о предметных областях. Онтология — это способ описания исследуемой предметной области в форме концептуальной модели, позволяющей однозначно трактовать применяемые при этом понятия с четким выстраиванием иерархических отношений между ними, а также со структуризацией и осмыслением знаний о предметной области.

Учитывая масштаб затрат на цифровую трансформацию экономики в мире, растущую вовлеченность многих компаний в глобальные цепочки поставок, актуальными становятся исследования по формированию эффективного онтологического механизма функционирования логистической деятельности, то есть единого понятийного пространства знания для всех участников ее. В некоторых исследованиях приводятся даже числовые оценки влияния отсутствия интеграции данных и алгоритмов на эффективность цифровой экономики в размере 62 % [2].

1. Эволюция интеграционных технологий в мировой логистической деятельности

Начиная с 1980-х происходило стремительное развитие логистики в индустриальных государствах, потребовавшего теоретического осмысления и комплекса возникших при этом междисциплинарных и междисциплинарных проблем эффективной интеграции данных и алгоритмов, онтологического механизма их реализации. Для этого проанализируем эволюционные уровни интеграции логистических систем в свете подготовки к переходу на единую цифровую логистическую платформу. Принцип управления логистикой в настоящее время состоит в формировании комплексной системы взаимодействия компаний и организаций на основе механизмов промышленного аутсорсинга. Такой механизм включает в себя организацию кооперационных взаимоотношений с интеграцией управляющих воздействий на все звенья цепочки поставок, формирование единого информационного пространства для координации и коммуникации также всех участников их. Аутсорсинг же подразумевает передачу некоторых логистических функций сторонним компаниям. По некоторым оценкам общий объем подобной логистики на мировом рынке составляет немногим более 172 миллиарда евро. Наиболее востребован аутсорсинг следующих услуг: складирование, экспедирование, возврат грузов, консолидация их, перегрузочные услуги, управление транспортными потоками и автопарком, услуги по поддержке покупателей и ряд других.

В результате совершенствования ИКТ, интернета и на их основе аутсорсинга постепенно сложились следующие уровни интеграции и координации логистической деятельности [3].

1.1. Субконтрактинговая сеть поставок (ССП) — это объединение на основе договоров юридически независимых компаний, взаимосвязанно действующих по реализации индивидуальных этапов общего процесса перемещения продукции, начиная от владельцев сырьевых ресурсов, кончая конечным потребителем. ССП характеризуются в основном лишь процессами складирования (хранения) и транспортировки с исключением производства продукции. Поскольку ССП основаны на достаточно длительных, существующих устойчивых хозяйственных связях, предприятия-участники которых связаны долгосрочными договорами, то это снижает отрицательное влияние на них случайных природных и рыночных факторов с уменьшением степени неопределенности и уменьшением трансакционных издержек.

1.2. Информационная СС (ИСС) — это более развитое, чем ССП, объединение с включением производственного блока по оказанию услуг или изготовлению конечного продукта. В этом случае участники ИСС формируют общие базы данных (БД), включающие информацию об участниках, их производственных возможностях с описанием технологических операций. Поскольку в последнее время БД носят облачный характер с размещением на разработанном для этого специальном вэб-сайте, то уже любой желающий, а не только участники ИСС могут самостоятельно войти на сайт и найти поставщика, производственный заказ, а также разместить и свой заказ. Основной целью ИСС является быстрое нахождение поставщика продукции (услуг) или, наоборот, заказчика для себя. Над участниками не довлеет никакой организатор ИСС, поэтому они добровольно и самостоятельно устанавливают договорные отношения между собой.

1.3. Производственно-логистическая сеть (ПЛС) уже использует функции ССП и ИСС и на настоящий момент является вершиной развития интеграционных возможностей логистической деятельности. В концепции ПЛС заложена разработка единой организационно-технологической среды и информационного пространства с возможностями объединения ресурсов на некоторое время различных автономных субъектов для повышения функциональной эффективности их и повышения конкурентоспособности. Если в ИСС облачная БД используется в пассивном режиме в виде некоторой доски объявлений, то в ПЛС она активно используется для планирования и управления реализацией договорных обязательств. Тогда помимо общей БД, в которую заносятся данные по всем совершенным логистическим операциям, классифи-

каторы, справочники, общие для всех участников ПЛС, имеется подобие базы знаний (функциональное ядро) с заданными алгоритмами оперативного управления.

1.4. Единая цифровая логистическая платформа (ЕЦЛП) на базе единой цифровой платформы управления (ЦПУ) экономикой страны [4, 5]. Если при формировании ПЛС в единое информационное пространство (ЕИП) которой по определению включено лишь незначительное число компаний опять же с ограниченным массивом информации и ограниченным функциональным набором, то в концепции единой ЦПУ участниками ее становятся все предприятия и организации страны с созданием единой системы сбора, хранения и анализа первичной учетной, статистической, технологической информации, интегрированной как между собой, так и с единой системой классификаторов, справочников, нормативов, представляющих реестры практически всех материальных, интеллектуальных и человеческих ресурсов страны на основе онтологического моделирования данных видов информационных ресурсов (ИР). Так, на рис. 1 показан цифровой стандарт сбора и хранения первичной учетной информации по единому формату.

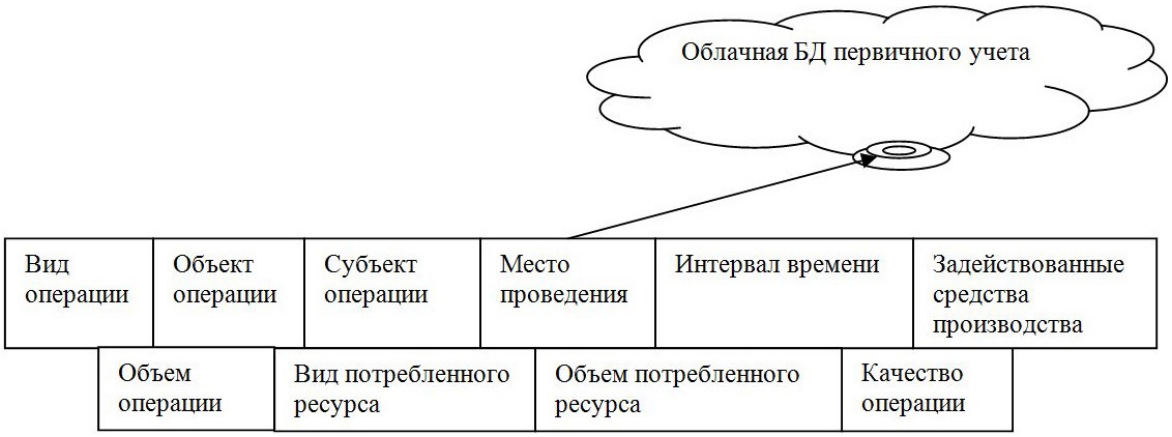


Рис. 1. Цифровой стандарт сбора и хранения первичной учетной информации

ЕЦЛП, основанная на представленной облачной структуре единой ЦПУ, позволит планировать и поддерживать логистические цепочки произвольной конфигурации с участием уже большинства предприятий страны, если не всех. Внедрение ЕЦЛП потенциально дает шанс отслеживания грузов в режиме реального времени, сокращения времени выполнения технологических операций логистических процессов со значительным повышением их прозрачности. Согласно ВТО, устранение барьеров в международных логистических поставках товаров сможет увеличить мировой ВВП на 5 % с общим объёмом перевозок на 15 %.

2. Онтологическое моделирование логистической деятельности

Поскольку в единой ЦПУ ключевым понятием служит операция, то и в ЕЦПЛ значение понимания операции как элементарного действия либо как совокупности действий не снижает своей актуальности. Классификация их должна являться базой для создания нормативов и регламентов планирования и управления логистикой, которой должны следовать как участники цепочек в виде производственных компаний, так и посредники, предприятия сферы услуг и торговли. Хранение информации в облаке о совершенных логистических операциях всех участников ЕЦПЛ, технологии по обработке ее обеспечат оперативный учет и анализ затрат на их реализацию всех видов ресурсов: финансовых, трудовых, материальных и информационных. В связи с важностью данной процедуры в мире неоднократно предпринимались попытки сформировать онтологическую модель в логистике, что завершилось разработкой неких подобий на стандарты относительно терминологии, одобренной многими ведущими мировы-

ми логистическими компаниями. Одной из версий подобного подхода является SCOR-модель (Рекомендуемая модель операций в цепях поставок), разработанная международным Советом по цепям поставок [6].

Указанная модель состоит из четырехуровневой пирамиды онтологического описания взаимосвязей участников логистической цепи в виде иерархии, где метрики верхнего уровня интегрируют метрики более нижнего уровня.

Рассмотрим более подробней элементы пирамиды по уровням.

К метрике 1-го уровня относят пять базисных бизнес-процессов:

Make («делать») — действия, характеризующие непосредственно производство изделия или услуги.

Source («снабжать») — действия, характеризующие факты снабжения различными ресурсами самого производства либо продажу товара.

Deliver («доставлять») — действия, характеризующие перемещение изделия потребителям.

Return («возвращать») — действия, характеризующие возврат бракованных изделий, тары, утилизацию брака и т.п.

Plan «Планирование» — действия, интегрирующие и координирующие предыдущие четыре.

К метрике 2-го уровня относят логистические процессы, под которыми понимается некоторым определенным образом спланированная по времени очередность исполнения логистических операций (функций), обеспечивающая достижимость заданных целевых показателей логистической деятельности. Всего выделено 26 базовых логистических процессов, являющихся пересечением пяти базисных бизнес-процессов метрики 1-го уровня со следующими участниками цепей поставок: поставщики поставщиков со следующими классами — начальный поставщик, поставщик 2-го круга, поставщик 1-го круга; поставщик; фокусная компания; потребитель; потребитель потребителей также со следующими классами — потребитель 1-го круга, потребитель 2-го круга, конечный потребитель.

К метрике 3-го уровня относят логистические функции [7], которые трактуются как ряд операций, обеспечивающих достижимость заданных целевых показателей при управлении потоками материальных ресурсов. К данной метрике относят такие функции логистики: закупочная, производственная, распределительная, транспортная, формирования запасов, складирования, сервиса, информационная.

К метрике 4-го уровня относят логистические операции [7], которые трактуются как любое действие, не подлежащее большему квантованию в рамках стоящих перед логистикой управленческих задач и необходимое при изменениях или модификации основных либо сопутствующих потоков товаров. Необходимость классификации таких операций в каждом бизнес-процессе определяется практикой реализации цепочек с потребностью ведения учета различных экономических и бухгалтерских показателей: финансовых и временных затрат, трудоемкостью и производительностью операций и пр. В настоящее время из-за неудачи по унификации логистических операций для всех возможных цепочек, считается, что набор их является уникальным для каждой компании. В результате такого положения зачастую наблюдается неудовлетворенность пользователями стандартным программным обеспечением (ПО) в сфере логистики, поскольку требует процедуры кастомизации его под потребности их. А это может оказаться очень непростой процедурой. В целом логистические операции с материальными поставками (товарами) можно определить как: погрузку, разгрузку, перегрузку с одного транспортного средства на другой их, экспедирование, распределение по складу, перевозку, приёмку на склад и отпуск со склада, хранение и сортировку, комплектацию и маркировку их.

Заключение

В работе на основе анализа эволюции интеграционных технологий в мировой логистической деятельности, а также формирующихся принципах цифровой трансформации производственной экономики рассмотрено онтологическое моделирование логистики, создающей цепочки добавленной стоимости из значительного числа участников, включая многочисленных производителей комплектующих, поставщиков ресурсов и услуг, потребителей изделий и самих логистических компаний.

Литература

1. *Кульба В. В., Микрин Е. А., Павлов Б. В., Платонов В. Н.* Теоретические основы проектирования информационно-управляющих систем космических аппаратов. – М. : Наука, 2006.

2. *Амелин С. В., Щетинина И. В.* Организация производства в условиях цифровой экономики // Организатор производства. – 2018. – Т. 26, № 4. – С. 7–18.

3. *Меденников В. И.* Трансформация технологий управления производственно-логистической цепочкой продукции при формировании единой цифровой платформы управления экономикой России // Цифровая экономика. – 2022. – 1(17). – С. 15–31.

4. *Алексеева Н. А., Осипов А. К., Меденников В. И. [и др.].* Экономические и управленческие проблемы землеустройства и землепользования в регионе. – Ижевск : ООО «Издательство «Шелест», 2022. – 225 с.

5. *Меденников В. И.* Математическая модель формирования цифровых платформ управления экономикой страны // Цифровая экономика. – 2019. – №1(5). – С. 25–35.

6. Supply Chain Operations Reference (SCOR) model. URL: https://en.wikipedia.org/wiki/Supply_chain_operations_reference (дата обращения: 01.10.2024).

7. *Толуев Ю. И., Планковский С. И.* Моделирование и симуляция логистических систем. – Киев : Миллениум, 2009. – 85 с.

УДК 004.75

МАСШТАБИРУЕМАЯ БЛОКЧЕЙН-АРХИТЕКТУРА ДЛЯ ЭФФЕКТИВНОГО УПРАВЛЕНИЯ ДАННЫМИ В IOT С ИСПОЛЬЗОВАНИЕМ ЛЕГКОВЕСНОГО КОНСЕНСУСА

М. О. Мельников

Елецкий государственный университет им. И. А. Бунина

Аннотация. Недавние исследования сосредоточены на применении технологии блокчейн для решения проблем безопасности в сетях Интернета вещей (IoT). Однако присущие блокчейну проблемы масштабируемости становятся очевидными при наличии большого количества IoT-устройств и значительных объемов данных, генерируемых этими сетями. В данной работе предлагается масштабируемая блокчейн-основа для управления данными IoT, способная обслуживать большое число устройств. Для решения указанных проблем используется легковесный алгоритм консенсуса — Делегированное Доказательство Доли Владения (Delegated Proof of Stake, DPoS), обеспечивающий высокую производительность и эффективность в условиях ограниченных ресурсов IoT-сетей. DPoS, как легковесный алгоритм, опирается на ограниченное число избранных делегатов для валидации и подтверждения транзакций, что позволяет уменьшить деградацию производительности и эффективности в блокчейн-сетях, связанных с IoT. В рамках исследования был реализован Межпланетный файловый обмен (IPFS) (контентно-адресуемый, одноранговый гипермедийный протокол связи) для распределенного хранения данных, а также использовался Docker для оценки производительности сети по таким параметрам, как пропускная способность, задержка и использование ресурсов. Анализ был разделен на четыре части: задержка, пропускная способность, использование ресурсов и время загрузки файлов, а также скорость оценки в распределенном хранилище.
Ключевые слова: блокчейн, консунсус-алгоритмы, IoT, интернет вещей, DPoS, PoW, PoS, EOSIO, распределённые системы.

Введение

Блокчейн-ориентированные сети Интернета вещей (IoT) обеспечивают безопасное и надежное соединение, а также обмен информацией между физическими или виртуальными объектами, оснащенными сенсорами, через Интернет [1]. Число устройств IoT демонстрирует устойчивый и значительный рост, увеличиваясь с каждым годом. По прогнозам, это число может достичь 29,42 миллиарда к 2030 году [2]. Распространение IoT-устройств с ограниченными ресурсами, которые являются доступными по цене и экономически эффективными, а также развитие современной информационно-коммуникационной инфраструктуры способствовали широкому использованию IoT-сетей в различных приложениях, таких как здравоохранение, промышленность, умные дома, интеллектуальные электросети, безопасность, видеонаблюдение и другие. Традиционно IoT-сети строились с использованием централизованной инфраструктуры и технологий, при этом данные с IoT-устройств собирались и обрабатывались через центральный сервер. Однако такой подход подвержен уязвимостям, связанным с безопасностью и конфиденциальностью данных, из-за киберугроз и физических атак [5]. Чтобы устранить эти проблемы, технология блокчейн рассматривается как один из основных кандидатов для создания защищенных реализаций IoT-сетей [7].

Современное состояние блокчейна имеет ряд проблем, при которых увеличение числа IoT-узлов в сети приводит к ухудшению производительности и эффективности [10]. Более того, согласование состояния блокчейна и верификация транзакций в сети узлов сложны и не способны справиться с большим числом транзакций одновременно в большинстве случаев,

таких как PoW, PoS [5]. Эти проблемы решаются с помощью алгоритмов консенсуса, таких как Делегированное Доказательство Доли Владения (DPoS). DPoS — это легковесный алгоритм консенсуса, который помогает устранить некоторые проблемы, связанные с PoW и PoS [3]. Процесс сохраняет безопасность, проверяя все легитимные транзакции, при этом DPoS использует выбранное число избранных делегатов.

Данная статья включает в себя следующее:

1. В работе предлагается четырехуровневая архитектурная модель для прозрачного и безопасного обмена данными IoT. Дизайн основан на двойной топологии блокчейна, включающей легковесный блокчейн (локальный блокчейн) и публичный блокчейн. Кроме того, в рамках предложенной модели используется IPFS, позволяющий хранить большие объемы данных в распределенной одноранговой системе хранения.

2. Статья затрагивает масштабируемость блокчейн-ориентированных IoT-сетей путем внедрения легковесного алгоритма консенсуса, что особенно важно для развертываний IoT с участием большого числа устройств.

3. В работе вводится некоторый уровень централизации, предусматривающий назначение небольшого числа избранных делегатов для валидации транзакций. Тем не менее, этот компромисс может быть приемлемым для некоторых случаев использования IoT, где важнее эффективность и масштабируемость, чем абсолютная децентрализация.

4. Для повышения эффективности сети в работе предлагается классификация устройств IoT на устройства с достаточным энергопотреблением и устройства с ограниченными ресурсами, в зависимости от их характеристик.

1. Используемые технологии

В данном разделе приводится краткое описание блокчейн-технологии, лежащей в основе предлагаемого решения, а также рассказывается о блокчейн-фреймворке, который лег в основу данного исследования.

Предложенная нами блокчейн-сеть использует подход к консенсусу, известный как Делегированное Доказательство Доли Владения (DPoS). На рис. 1 представлено сравнение ведущих блокчейн-систем. Кроме того, проблемы масштабируемости решаются за счет использования различных хэш-алгоритмов [4].

Features	Bitcoin	Ethereum 2.0	Hyperledger-Fabric	IoTA	EOSIO
Consensus	POW	POS	PBFT	Tangle	DPOS
Consensus finality	×	×	✓	×	✓
Run Smart Contract	×	✓	✓	×	✓
Interchain	×	×	×	×	✓
Feeless	×	×		✓	✓
Scalable	×	✓	×	✓	✓
Energy efficient	×	×	×	✓	✓
TX throughput (TPS)	7	100+	1,000	7-12	4000+

Рис. 1. Сравнение блокчейн-систем

PoS предлагается как альтернатива PoW [1]. Он основан на принципе, что участники, обладающие долей в сети, могут участвовать в процессе консенсуса, способствуя расширению блокчейна и проверке транзакций. В отличие от PoW, который требует от майнеров выполнения вычислительно сложных хэш-алгоритмов для подтверждения транзакций, PoS требует

от пользователей демонстрации владения определенным количеством средств, также известным как их доля в сети. Однако в DPoS вместо того, чтобы каждый узел напрямую участвовал в проверке и подтверждении транзакций, как это происходит в PoW и PoS, используется меньший круг избранных делегатов для верификации транзакций и добавления новых блоков. DPoS ускоряет процесс принятия решений, вовлекая меньшее количество участников — от 21 до 101 делегата. Участники могут присоединяться, не нуждаясь в обширных вычислительных ресурсах, передавая свои полномочия на голосование делегатам, которым они доверяют [3]. Эти характеристики делают DPoS легковесным, масштабируемым и эффективным алгоритмом консенсуса для блокчейн-ориентированных IoT-сетей.

Помимо известных Bitcoin и Ethereum, существует множество других блокчейн-фреймворков с различными особенностями. В данной работе исследуется полезность блокчейна EOSIO [6]. EOSIO обладает несколькими привлекательными качествами, такими как гибкость, признанный статус, наличие множества инструментов для разработчиков и мощный набор средств для разработки контрактов (CDT). Кроме того, в EOSIO возможно программировать блокчейн, используя смарт-контракты (СК), написанные на языке C++, которые могут храниться в блокчейне без ограничений по размеру.

EOSIO [6] представляет собой блокчейн-систему для криптовалюты, называемой enterprise operating system (EOS). Подобно Bitcoin и Ethereum, EOSIO является децентрализованной, разрешенной блокчейн-сетью. Однако между ними существуют существенные различия в целях и возможностях. В отличие от Bitcoin и Ethereum, EOSIO использует алгоритм консенсуса DPoS (Delegated Proof of Stake), который предполагает назначение ограниченного числа делегатов, известных как производители блоков (Block Producers, BPs). В сети EOSIO процесс голосования определяет 21 BP, которым доверено принятие решений по включению вновь созданных блоков в существующую цепь. Использование DPoS позволяет EOSIO значительно повысить скорость обработки транзакций. Основные компоненты блокчейна EOSIO включают Nodeos, Cleos и Keosd. Nodeos отвечает за валидацию и синхронизацию блоков в сети, Cleos предоставляет интерфейс командной строки для взаимодействия с блокчейном, а Keosd хранит приватные ключи на локальных компьютерах.

В EOSIO использование ресурсов требуется для каждой транзакции и выполнения смарт-контрактов. Эти ресурсы подразделяются на вычислительную мощность CPU, пропускную способность сети NET и оперативную память RAM. Для выполнения смарт-контракта пользователю необходимо обладать достаточными ресурсами, чтобы покрыть затраты на потребление ресурсов. Поэтому платформа требует от SCP и пользователей покупки RAM или «стейкинга» CPU и NET. «Стейкинг» подразумевает выделение определенного количества токенов (EOS) для резервирования ресурсов BP.

2. Платформа распределенного хранения данных IoT на основе блокчейна

В данном разделе представлена четырехуровневая архитектурная конструкция для прозрачной и безопасной системы обмена данными в Интернете вещей (IoT), как показано на рис. 2.

Все уровни функционируют автономно и децентрализованно для управления вычислениями и хранением данных. Основная цель этой архитектуры — обеспечить масштабируемость блокчейна в плане пропускной способности и задержки транзакций. Главная задача состоит в том, чтобы снизить нагрузку на блокчейн, вызванную миллионами локальных транзакций в IoT-сетях.

Уровень сети IoT

На этом уровне устройства разделены на две группы. Первая группа включает ограниченные IoT-устройства с ограниченными вычислительными возможностями, памятью и сете-

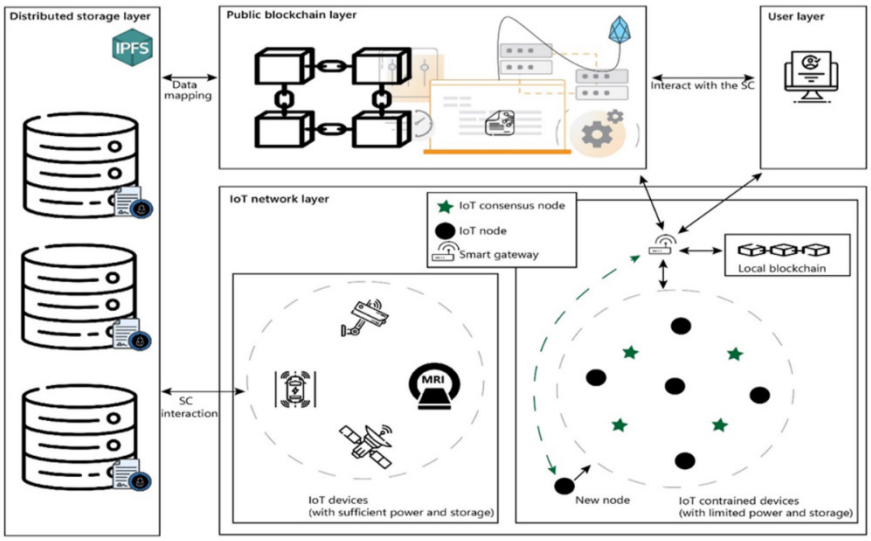


Рис. 2. Архитектура предложенного блокчейна

выми ресурсами. Вторая группа состоит из устройств потоковой передачи данных IoT с достаточной вычислительной мощностью, объемом памяти и сетевыми возможностями. Как показано на рис. 1, устройства потоковой передачи данных могут напрямую взаимодействовать с смарт-контрактами (СК) и загружать данные в хранилище без необходимости в дополнительных устройствах для упрощения взаимодействия. Они также могут напрямую связываться с компонентами хранилища. В то же время ограниченные устройства IoT используют умные шлюзы для взаимодействия с блокчейном, что позволяет преодолеть их ограниченные возможности в работе с СК. Уровень также включает узлы консенсуса и узлы IoT. Узлы IoT собирают данные из окружающей среды и отправляют их в локальный блокчейн через заданные интервалы. Узлы консенсуса, помимо сбора данных, также выполняют консенсусные алгоритмы, такие как DPoS (Delegated Proof of Stake).

Уровень публичного блокчейна

Публичный блокчейн представляет собой децентрализованную сеть, состоящую из блокчейн-хранилищ, каждое из которых имеет полную копию всей системы. Это обеспечивает устойчивость системы в случае выхода из строя большого числа узлов сети и потери данных. Систему можно восстановить с помощью одного узла, который хранит полную копию блокчейна. Смарт-контракты реализуются на уровне публичного блокчейна, и их функции специально разработаны для IoT, такие как регистрация новых узлов и обеспечение взаимодействия между публичным блокчейном и шлюзом. Смарт-контракты автоматизируют и обеспечивают безопасность процесса взаимодействия.

Уровень распределенного хранения

Совмещение конфиденциальности и прозрачности на блокчейне вызывает трудности [8]. Хранение исходных данных на блокчейне ведет к проблемам приватности и масштабируемости. Для решения этой задачи применяется сочетание внешнего хранения данных и проверки на блокчейне. Основная ответственность за хранение данных возложена на внешнее хранилище, реализованное с использованием протокола IPFS (Inter-Planetary File System), который объединяет вычислительные устройства в единую файловую систему, устраняя единые точки отказа. Устройства потоковой передачи данных могут напрямую загружать данные в IPFS, который предлагает преимущества, такие как отсутствие единых точек отказа и обеспечение глобального распределенного хранения.

Пользовательский уровень

Пользователи могут взаимодействовать со шлюзом, чтобы получить необходимые данные IoT, поскольку шлюз также сохраняет данные локального блокчейна. Однако, если пользова-

тель получает данные от устройств потоковой передачи, их объем может быть значительным. Поток данных IoT разбивается на части в соответствии с интервалами выборки и передается для хранения вне блокчейна, а на блокчейне фиксируются только детали, такие как номер фрагмента, временная метка и хэш. Таким образом, можно определить различные требования для получения данных из блокчейна.

3. Анализ производительности

В данной секции приводятся результаты оценки производительности, выполненной на ноутбуке с операционной системой Ubuntu 22.04 LTS, оснащенном процессором Intel Core i3, 8 ГБ ОЗУ и жестким диском объемом 1 ТБ. В рамках эксперимента была установлена EOSIO (v2.2), включающая такие компоненты, как Nodeos, Cleos и Keosd. Также использовался инструмент EOSIO Contract Development Tool (v1.8.1) для компиляции смарт-контрактов, а системные контракты (v1.6.0) предоставляли базовые функциональные возможности для блокчейна EOSIO. В качестве части настройки применялся Docker для инициализации локального узла EOSIO.

Данный раздел представляет результаты оценки производительности, включая сравнительный анализ консенсусных алгоритмов PoS и DPoS. Основная цель этой оценки — определить потенциал масштабируемости в контексте IoT-сетей. Исследуется способность системы подключать повседневные устройства к Интернету и оценивать ее возможности обработки все большего количества устройств со временем, даже с ограниченными вычислительными ресурсами. В начальной версии рассматриваемой IoT-архитектуры отсутствовали механизмы интеграции протокола связи внутри IoT-сети. Обычно смарт-шлюз требует подключения к сети, которое может быть установлено как проводным, так и беспроводным способом. Однако в рамках данной работы было смоделировано использование протокола связи LoRaWAN. Как показано в [29], один шлюз LoRaWAN способен управлять до 100,000 узлов-датчиков, передающих пакеты данных размером 50 байт каждый час. Важно отметить, что рассматриваемая архитектура ориентирована на создание масштабируемой системы, где блокчейн обеспечивает безопасность вне зависимости от используемого протокола связи. Мы провели сравнение с недавними подходами, описанными в литературе, такими как подход на базе PoS для «умных сообществ» [1]. В дальнейших экспериментах производительность архитектуры оценивается по таким метрикам, как пропускная способность, задержка, использование ресурсов (NET-пропускная способность и время CPU), а также требования к хранению данных.

Оценка производительности IoT-архитектуры с учетом алгоритма консенсуса DPoS в первую очередь предполагает измерение задержки и пропускной способности. Этот процесс включает несколько этапов, таких как валидация, добавление данных в блок и измерение пропускной способности транзакций. Задержка — это метрика, показывающая время, затраченное на то, чтобы пакет данных достиг шлюза и стал частью блокчейна. Более высокие значения задержки указывают на большую сложность добавления пакетов данных в блоки и усложнение масштабирования IoT-блокчейна. Вторая метрика — это пропускная способность, измеряемая количеством успешных транзакций с момента первой до последней транзакции в цепочке. Пропускная способность демонстрирует, как система справляется с ростом числа узлов блокчейна IoT на один шлюз. Для оценки масштабируемости были проведены тесты, в которых число узлов блокчейна IoT варьировалось от 500 до примерно 20,000. Размер блока, содержащего пакеты данных, составлял 1 МБ, а размер нагрузки — 50 байт.

На рис. 3 показаны тенденции изменения задержки при добавлении пакета данных. В случае использования алгоритма PoS задержка для добавления пакета данных увеличивается. Например, при 500 узлах задержка составляет 55.4 мс для PoS, тогда как для DPoS — всего 0.976 мс. Это связано с тем, что при DPoS небольшое количество делегатов подтверждает транзакции, а при PoS процесс валидации занимает больше времени из-за отсутствия мгновенного

выполнения и большего количества валидаторов. В результате пакеты данных в PoS ставятся в очередь на валидацию, что увеличивает время обработки.

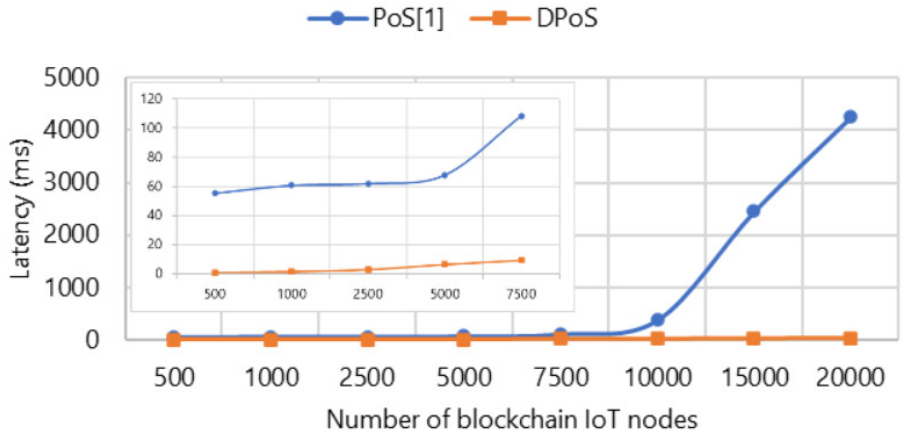


Рис. 3. Тенденции изменения задержки при добавлении пакета данных

Результаты второй метрики — пропускной способности — представлены на рис. 4. Очевидно, что алгоритм DPoS превосходит PoS в эффективности обработки транзакций. Например, при 20,000 узлов пропускная способность достигает максимума в DPoS, поскольку система справляется с ростом числа узлов, в то время как PoS насыщается при достижении 16,006.73 транзакций, что ограничивает его производительность.

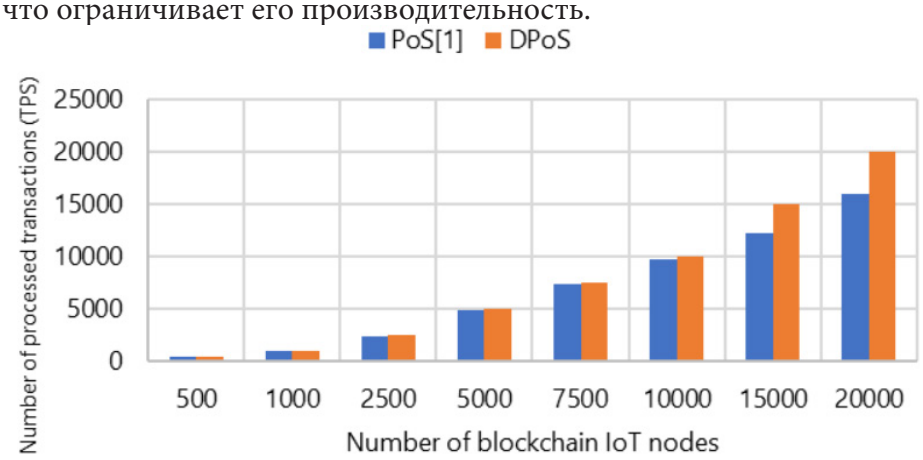


Рис. 4. Динамика пропускной способности

Узлы, участвующие в DPoS, существенно влияют на общие энергозатраты. Поэтому делегаты не только снижают потребление энергии, но и ускоряют обработку транзакций. В отличие от этого, подход на базе PoS требует от устройств IoT взаимодействия с большим числом валидаторов, что повышает энергопотребление. Таким образом, подход DPoS превосходит PoS по энергоэффективности.

Результаты, представленные на рис. 5, демонстрируют влияние на использование ресурсов NET и CPU. В эксперименте варьировалось число узлов от 500 до 20,000 для оценки влияния на NET/CPU. На рис. 5 видно, что при 500 узлах время CPU составляет 1.136 мс, а при 20,000 узлах — почти 27.326 мс. NET-пропускная способность остаётся стабильной на уровне 104, что объясняется небольшим размером пакета данных.

Результаты анализа на рис. 3–5 показывают, что при увеличении числа узлов блокчейна в PoS наблюдается линейный рост задержки, тогда как в DPoS наблюдается хорошая линейная масштабируемость пропускной способности при увеличении числа узлов. Например, при 500 узлах DPoS достигает пропускной способности 500 TPS, а PoS — 496.31 TPS, что демонстрирует превосходство DPoS при увеличении числа узлов.

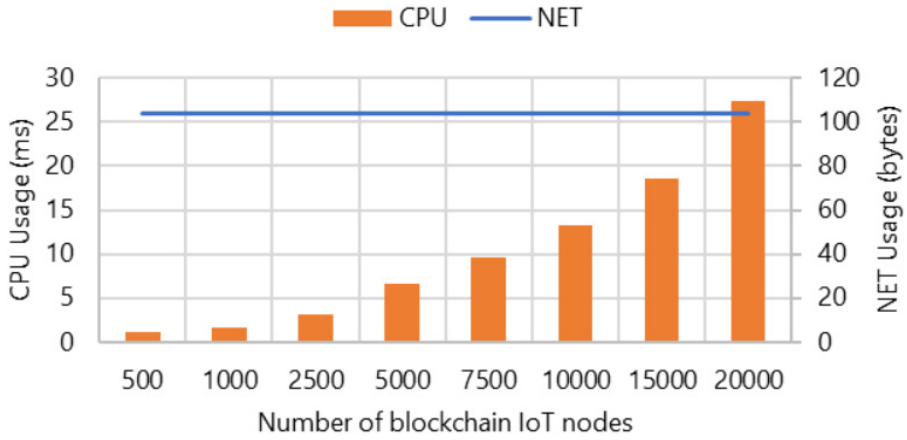


Рис. 5. Влияние на использование ресурсов NET и CPU

В рамках оценки производительности в публичной сети проводился анализ времени загрузки и скорости IPFS. Конфигурация системы включала 8 ГБ ОЗУ, 2 ядра и полосу пропускания 8 МБ. На рисунке 6 показано, что скорость загрузки составляла примерно 7 МБ/с. На рис. 6 видно, что хэш файла был загружен и извлечен из блокчейна. Время загрузки оказалось больше времени извлечения, так как при загрузке сохраняется идентификатор контента на блокчейне.

Test	File size	Upload time	Upload speed
1	50MB	6.12s	7.85 MB/s
2	100MB	15.03s	6.74 MB/s
3	500MB	117.12s	6.53 MB/s
4	1GB	236.63s	7.28 MB/s

Рис. 6. Результаты сравнения в рамках скорости загрузки

Заключение

В данной работе предлагается масштабируемая блокчейн-архитектура для управления данными в сетях Интернета вещей (IoT) с ограниченными ресурсами, основанная на использовании алгоритма консенсуса DPoS (Delegated Proof of Stake) для обеспечения сквозной безопасности. DPoS обеспечивает такую безопасность через механизмы верификации и валидации, которые выполняются ограниченным числом выбранных делегатов, что позволяет уменьшить влияние деградации производительности устройств IoT. Основные исследуемые метрики включают задержку, пропускную способность и использование ресурсов в диапазоне от 500 до 20,000 узлов сети. Алгоритм консенсуса DPoS был реализован на платформе EOSIO, распределенное хранение данных организовано с помощью IPFS, а Docker использовался для оценки производительности сети по показателям пропускной способности, задержки и использования ресурсов устройств IoT. Анализ был разделен на четыре части: задержка, пропускная способность, использование ресурсов и время загрузки файлов с оценкой скорости в распределенной системе хранения. Экспериментальные результаты показали, что DPoS превосходит PoS по пропускной способности, задержке и эффективности использования ресурсов в устройствах IoT. Кроме того, было продемонстрировано, что подход DPoS эффективен в IoT-приложениях, где требуется низкая задержка или высокая энергоэффективность.

Литература

1. *Maftei A., Petrariu P., Popa V.* Massive Data Storage Solution for IoT Devices Using Blockchain Technologies // Sensors. – 2023. – 23(3). – P. 1570.

2. *Asif Y., Rochester M., Ousat B., Ghaderi M.* Throughput, Coverage and Scalability of LoRa LP-WAN for Internet of Things // In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). – 2018. – P. 1–10.

3. *Zheng W., Zheng Z., Dai H., Chen Xu., Zheng P.* Xblockeos: Extracting and exploring blockchain data from eosio // Information Processing & Management. 2021. – 58(3). – P. 102477.

4. *Zhang Q., Zhao Z.* Distributed storage scheme for encryption speech data based on blockchain and IPFS // The Journal of Supercomputing. – 2023. – 79(1). – P. 897–923.

5. *Kaur M., Gupta S., Raboaca M.* Ipfs: An Off-Chain Storage Solution for Blockchain // In Proceedings of International Conference on Recent Innovations in Computing: ICRIC 2022. – Springer, 2023. – Vol. 1. – P. 513–525.

6. *Kummar S., Bhushan B., Bhatia S.* Blockchain based big data solutions for Internet of Things (IoT) and smart cities // In New Trends and Applications in Internet of Things (IoT) and Big Data Analytics. – Springer International Publishing, 2022. – P. 225–253.

7. *Xing F., Baoning N., Zhenliang L.* Scalable blockchain storage systems: research progress and models // Computing. – 2022. – 104(6). – P. 1497–1524.

8. *Kutub T., Al-Sakib P., Sadia I.* Internet of things (IoT) // In Emerging ICT Technologies and Cybersecurity: From AI and ML to Other Futuristic Technologies. – Springer, 2023. – P. 165–183.

9. *Chandra P., Ponsy R.* Performance analysis on Diversity Mining-based Proof of Work in bifolded consortium blockchain for Internet of Things consensus // Concurrency and Computation: Practice and Experience. – 2021. – 33(16).

10. *Lin Y., Zhang L., Li J., Ji L., Sun Y.* A survey of application research based on blockchain smart contract // Wireless Networks. – 2022. – 28(2). – P. 635–690.

УДК 004.8

ИСПОЛЬЗОВАНИЕ КОНВЕЙЕРА ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ДАННЫХ В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ СХОЖИХ ТИПОВ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Г. В. Митин, А. В. Панов

Московский государственный университет информационных технологий, радиотехники и электроники

Аннотация. В статье описывается архитектура системы интеллектуальной обработки данных, решающая задачу идентификации пользователей информационной системы по манере их поведения. Представлены результаты эксперимента на модели данных, подтверждающие действенность данной архитектуры.
Ключевые слова: информационная система, интеллектуальная обработка данных, машинное обучение, аутентификация пользователей.

Введение

Развитие информационных технологий позволяет автоматизировать многие сферы человеческой деятельности, относящиеся к обработке данных. Все больше задач, ранее считавшихся по силам лишь человеку, теперь решаются при помощи специального программного обеспечения.

Комбинируя алгоритмы анализа данных, можно находить в потоках данных информацию, неочевидную при простом считывании этих данных. Таким образом, можно находить полезные закономерности, опираясь на косвенные признаки.

В данной работе рассматривается использование архитектуры конвейера обработки потоковых данных для поиска поведенческих шаблонов пользователей информационной системы.

1. Актуальность проблемы

Проблема идентификации человека, который пользовался рабочей станцией (компьютером) в некоторый период времени, является весьма актуальной. Учетные записи пользователей не позволяют решить эту проблему. Так, например, сотрудник А может использовать учетную запись сотрудника Б, чтобы воспользоваться правами доступа Б к функциям корпоративной информационной системы, которых нет у А. Одним из путей решения данной проблемы является распознавание «почерка» работы человека за компьютером, на основании записей о его действиях. Подобный «почерк» называют поведенческой биометрией [1].

Подобные закономерности являются сложными для распознавания, поэтому для их поиска нужно использовать многоуровневые системы анализа потоковых данных. Один из вариантов такой архитектуры описан в данной работе. В отличие от других решений [1], архитектура, представленная в рамках статьи, не использует искусственные нейронные сети, что делает ее более доступной.

2. Постановка задачи

Определение схожих типов поведения пользователей является актуальной и нетривиальной задачей. Это может использоваться для выявления несанкционированного использования учетных записей пользователей другими пользователями на основании поведенческих

шаблонов. Искомые закономерности имеют вероятностную природу и относятся к слабо коррелированным информационным паттернам.

Действия пользователей преследуют определенные цели. Разные пользователи имеют разные цели использования системы и добиваются своих целей через разные последовательности действий. Анализируя последовательности действий пользователей, представляется возможным идентифицировать подходы к достижению целей, характерные для того или иного пользователя.

Для того, чтобы эффективно решать поставленную задачу, необходим метод, позволяющий определять намерения, стоящие за запросами пользователей. Поскольку для каждой информационной системы существует ограниченный круг приложений, намерения в запросах пользователей можно разделить на некоторый ограниченный круг категорий. Более того, так как информационная система предоставляет ограниченный спектр запросов, одному намерению соответствует лишь ограниченный круг запросов, подчиняющийся некоторым вычислимым законам. Зная эти законы, становится возможным вычислить цель действий пользователя. Таким образом, для выяснения сути запросов, можно использовать алгоритмы классификации.

Решение этой задачи требует построения конвейера обработки данных, способного как определять тип действия пользователя, так и находить закономерности в его действиях. Архитектура подобного конвейера была описана в [2]. Она разделяется на несколько функциональных уровней, решающих задачи различных уровней абстракции.

3. Функциональная схема конвейера интеллектуальной обработки данных

Предложенный подход предполагает определение типа запроса, переданного пользователем, нахождение закономерностей в запросах пользователя и анализ этих закономерностей для определения характерных черт, по которым данного пользователя можно идентифицировать.

Решение описанной задачи производится на базе многоуровневой архитектуры конвейера интеллектуальной обработки данных, представленной в [2] (см. рис.1). Архитектура нацелена на работу с потоковыми данными в условиях информационного шума и позволяет решать задачи различных уровней абстракции. Функционально в ней можно выделить три уровня:

1) Уровень подготовки данных. Основными задачами этого уровня является приведение данных в наиболее удобный для анализа вид, определение особенностей распределения данных в пространстве признаков и фильтрация информационного шума.

2) Уровень обработки данных. Основными задачами данного уровня являются поиск информационных паттернов в данных и поиск зависимостей между появлением этих паттернов. Он оперирует данными, очищенными от информационного шума и выбросов.

3) Уровень анализа результатов обработки данных. Этот уровень проводит заключительные этапы работы системы, включающие в себя анализ найденных закономерностей в появлении паттернов и выделение на их основе полезного знания.

Система обрабатывает категориальные и числовые данные как два отдельных потока данных, применяя соответствующие методы обработки для каждого потока. Исключение составляет уровень анализа результатов, который объединяет оба потока.

В рамках данного исследования мы исходим из предположения, что входящие данные не содержат информационного шума. Тогда нам понадобятся уровень обработки данных и уровень анализа результатов.

В рамках задачи определения схожих типов поведения пользователей, уровень обработки данных призван определить класс действия, совершенного пользователем. Среди таких действий может быть как «нормальный» запрос, так и какой-либо тип кибератаки. Для этого на данном уровне используется адаптивная классификация [3] и поиск ассоциативных правил. Адаптивная классификация позволяет, руководствуясь обновляющейся обучающей выбор-

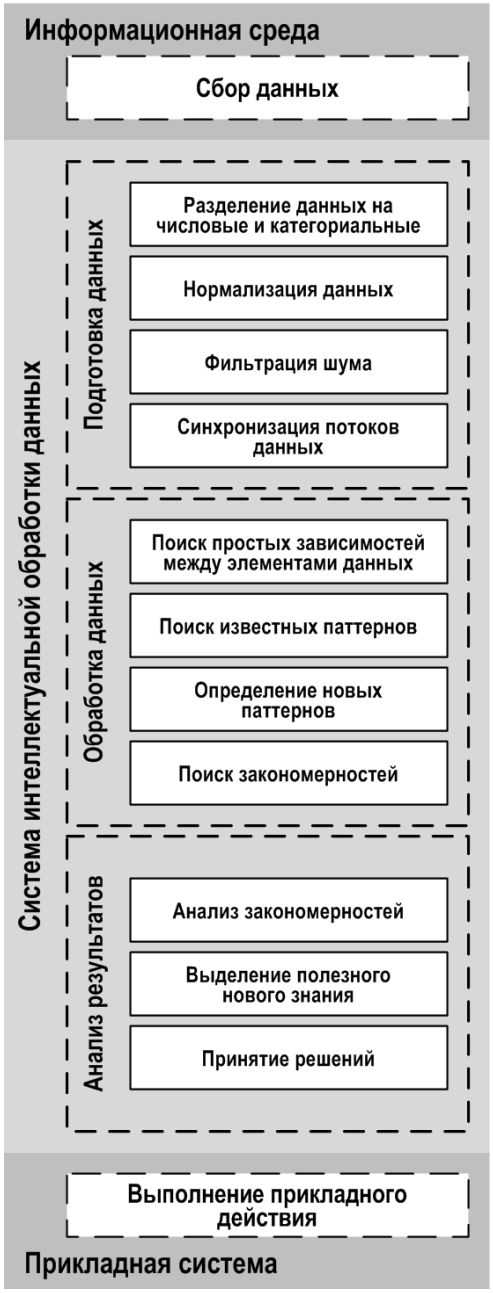


Рис. 1. Функциональная схема конвейера интеллектуальной обработки данных

кой, классифицировать действия пользователя, а поиск ассоциативных правил — определить набор общих закономерностей.

Уровень анализа результатов обработки данных определяет закономерности между действиями пользователя и определяет характерные особенности в поведении пользователей. Для этого на данном уровне используется адаптивная классификация и методы поиска последовательных шаблонов. При помощи метода поиска последовательных шаблонов производится поиск закономерностей, распределенных по времени, в то время как адаптивная классификация позволяет классифицировать обнаруженные закономерности – то есть определить тип поведения пользователя. Кроме того, адаптивная классификация позволяет добавлять новые закономерности в поведении пользователей.

4. Испытание системы интеллектуальной обработки данных на модели данных

Для моделирования прикладной задачи было проведено модульное тестирование каждого уровня системы обработки данных по отдельности. Ключевыми являются уровень обработки данных и уровень анализа результатов обработки.

Модель входящего потока данных строилась на базе набора данных NSL-KDD [4]. Это расширенный вариант набора данных KDD'99, описывающего характерные черты компьютерных сетевых атак. Набор содержит записи, имеющие одновременно категориальные и числовые признаки.

Для упрощения обработки, множество классов, обозначающих разнообразные типы компьютерных атак, было преобразовано в 5 макро-классов, обозначающих 4 типа атак и отдельный класс для «нормальных» или безопасных запросов.

Набор данных NSL-KDD был модифицирован, путем изменения порядка следования записей, для моделирования поведения пользователей информационной системы. Задачей модификации было создать последовательности действий, имеющих некоторый смысл для прикладной системы, при этом не снижая качества набора данных. Например, были сгруппированы попытки получения информации об инфраструктуре рассматриваемой сети (Probe-атаки) с другими типами атак, моделируя поведение злоумышленника, вначале проводящего разведку, а затем пытающегося причинить вред целевой информационной системе. Другим примером является модель пользователя, который по большей части работает с системой в штатном режиме, но время от времени его действия подпадают под определение информационной атаки. Такими действиями являются неправильные и потенциально вредоносные запросы, попытки «атаковать» систему, подбирая собственный пароль и использование программного обеспечения для анализа сетей, которое так же может рассматриваться как проведение Probe-атаки.

4.1. Уровень обработки данных

На наборе данных, имеющих как числовые, так и категориальные признаки, классификаторы числовых и категориальных данных показали различные результаты.

• Точность классификации NSL-KDD у составного адаптивного классификатора, реализующего метод Naive Bayes в режиме числовой классификации, выше, чем аналогичная точность у стандартного классификатора Naive Bayes для 2 наиболее распространенных классов.

• Неопознанные точки данных, появившиеся при классификации, позволяют снизить общее количество ошибок классификации, вызванных недостатком данных или несовершенством алгоритма классификации.

На тестовой модели, классификаторы, относящиеся к уровню обработки данных, показали точность (accuracy) 83–98%. При этом блок категориальной классификации, использовавший дерево решений, стремился охватить все имеющиеся классы. В то же время блок числовой классификации, использовавший алгоритм Gaussian Naive Bayes (GNB), сосредоточился на двух наиболее распространенных классах (DoS, normal), игнорируя менее распространенные классы.

4.2. Уровень анализа результатов

В задаче определения поведенческих шаблонов пользователей, уровень анализа результата берет на себя опознание пользователя по набору выявленных последовательностей. Следует заметить, что длины последовательностей и описывающих их векторов признаков, не одинаковы. В качестве входных данных для этого слоя были взяты метки классов, принадлежащих точкам данных из наборов для уровня обработки данных.

С точки зрения алгоритма поиска последовательных шаблонов данные рассматривались как последовательность предметных наборов из одного предмета — метки класса. Такое представление допустимо для случаев, когда нет событий происходящих одновременно, и позволяет обращать внимание на порядок следования меток классов, так как элементы внутри одного предметного набора сортируются в унифицированном порядке.

В модели были выделены 5 типов поведения:

• Hacker_1 — DoS-атаки. Имитация враждебно настроенного пользователя, пытающегося навредить информационной системе.

• Hacker_2 — Probe-атаки, за которыми следует ряд DoS-атак. Имитация враждебно настроенного пользователя, пытающегося провести разведку внутреннего устройства информационной системы, чтобы нанести максимальный ущерб.

• Hacker_3 — DoS-атаки с несколькими R2L-атаками. Имитация враждебно настроенного пользователя, пытающегося навредить информационной системе и получить доступ к конфиденциальным данным.

• Ordinary_1 — штатное использование системы, иногда сопровождающееся действиями, похожими на кибератаку (DoS).

• Ordinary_2 — штатное использование системы, с использованием средств разработчика, которые попадают под определение кибератаки (DoS, Probe).

В качестве алгоритма классификации для блока адаптивной классификации в рамках уровня анализа результатов были использованы деревья решений, как алгоритм, показывающий хорошие результаты при составлении ансамблевых решений.

Уровень анализа результатов обработки верно оценил классы «Hacker_2», «Ordinary_1» и «Ordinary_2», но отнес элементы класса «Hacker_3» к классу «Hacker_1» из-за того, что различия между классами небольшие, и класс «Hacker_3» представлен малым количеством элементов (0.6 %). Общая точность классификации (accuracy) на данном уровне составила 99.4 %. Таким образом, можно заключить, что представленный конвейер обработки данных справился с поставленной задачей.

Заключение

Представленный в данной работе конвейер обработки данных способен не только решать задачу классификации, но и распознавать закономерности, которые не проявляются при простой обработке входящего потока данных. Задача определения поведенческих шаблонов пользователей информационной системы является актуальной, особенно при использовании методов, не связанных с искусственными нейронными сетями.

Согласно результатам эксперимента, представленное в данной работе решение задачи определения схожих типов поведения пользователей информационной системы является концептуально верным.

Литература

1. Анализ перспективных методов поведенческой биометрии для аутентификации пользователей / В. А. Довгаль, Д. В. Довгаль // Вестник АГУ. – 2017. – Вып. 3.– С. 206.

2. Общая архитектура систем обработки данных с применением технологий машинного обучения для поиска информационных паттернов / Г.В. Митин, А.В. Панов / Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научной конференции, 4-6 декабря 2023 г. Воронеж, 2024. – С. 1427–1433.

3. Адаптивная технология классификации с обратной связью на основе методов машинного обучения / Г. В. Митин, А. В. Панов / Современные наукоемкие технологии. – 2024. – № 1. – С. 55–61.

4. NSL-KDD // Kaggle.com URL: https://www.kaggle.com/datasets/hassan06/nslkdd/data

УДК 004.04

ЗАДАЧИ ИНТЕГРАЦИИ ДАННЫХ РАЗНОРОДНЫХ ИСТОЧНИКОВ И МЕТОДЫ ИХ РЕШЕНИЯ

А. А. Носов, И. Е. Воронина

Воронежский государственный университет

Аннотация. Рассматривается проблема типового решения задач интеграции данных. Приведен анализ методов интеграции данных разнородных источников. Представлены преимущества и недостатки использования подходов ETL, ELT, EAI, EII.
Ключевые слова: интеграция данных, ETL, ELT, EAI, EII, автоматизации, аналитика, бизнес-процесс, бизнес-задача, информационная система.

Введение

Современные промышленные предприятия имеют сложную информационную инфраструктуру, состоящую из множества разнородных источников данных: различные базы данных, электронные таблицы, веб-сервисы, текстовые файлы и т. д. Такое положение на предприятиях складывается в результате «островной» автоматизации отдельных бизнес-процессов. Любая задача, которая связана с принятием решений на основе данных, требует наличия этих данных в единой зоне или для того, чтобы система была наиболее полной, ей необходимы данные из других систем.

Попытки объединить в единое целое несколько разных систем автоматизации в целях воспользоваться их данными оказываются чаще всего неудачными и приходится прибегать к средствам интеграции данных.

Интеграция данных — направление, которое включает в себя следующие процессы:

- объединение или перемещение данных, находящихся в разных источниках;
- предоставление данных пользователям в унифицированном виде.

В качестве примера задач требующих интеграцию данных можно привести следующие проекты:

1. В нефтедобывающих компаниях существует обычно три больших направления: добыча, переработка, продажа. Направление, занимающееся добычей, обычно не знает, что творится в остальных направлениях, но блок логистики (перевозка нефти и трубопроводы) всегда находится в направлении переработки. Между этими направлениями нет передаваемой информации об изменении добычи нефти в блок переработки. Из-за этого возникает логистическая бизнес-проблема: уменьшается резко объем добычи нефти в определенном месте, но вагоны ждут, простаивают, в то время как в другой точке наблюдается резкий рост добычи нефти, но во второе место не успевают дойти вагоны, там заполняются склады. В итоге приходится умышленно снижать добычу нефти только из-за того, что блоки между собой не обмениваются информацией. Данная проблема стала особенно актуальна, когда государство начала контролировать работу нефтяной отрасли и требовать резкого повышения эффективности.

2. Происходит объединение нескольких компаний, контакт-центр становится общим, но информация о продуктах находится в разных ERP-системах. Возникает проблема онлайн и оффлайн синхронизации приложений нескольких систем, содержащих сведения о некоторых продуктах. У специалиста возникает проблема найти нужный продукт в семнадцати приложениях с различным интерфейсом.

Применение интеграции данных направлено на повышение операционной эффективности организации или подразделения, снижение издержек и сокращение расходов, быстрое реагирование на изменения внешней бизнес-среды, соответствие требованиям регуляторов.

В задачи интеграции данных входит:
- формирование чистых консолидированных данных для отчетности и аналитики;
- анализ данных из различных корпоративных систем;
- доступность любых данных из любых систем для бизнес-пользователей;
- минимальное влияние средств отчетности или аналитики на системы источники.

Сложность интеграции данных обусловлена следующими причинами:
- многочисленные «свалки» данных;
- отсутствие единых корпоративных справочников;
- неупорядоченная документация;
- несовместимые, неактуальные, некачественные данные;
- многочисленные инструменты и проектные команды;
- отсутствие анализа зависимостей;
- постоянно меняющаяся и усложняющаяся инфраструктура.

В настоящее время есть множество бизнес-задач по интеграции данных, представленных на рис. 1. А также методы по их реализации: ETL/ELT, EAI, EII, ESB.

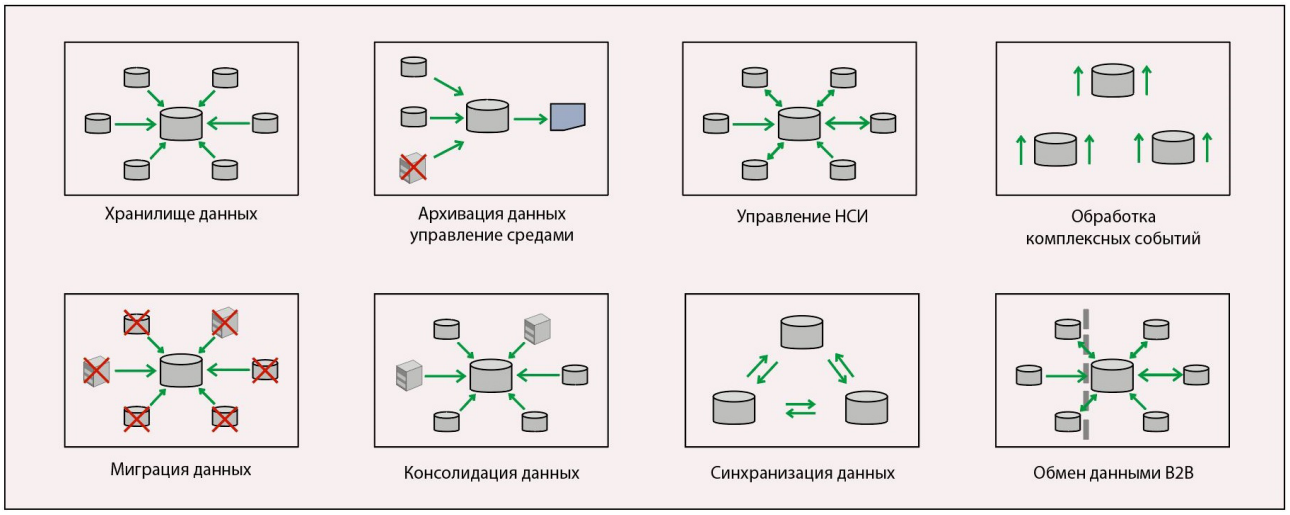


Рис. 1. Задачи интеграции данных

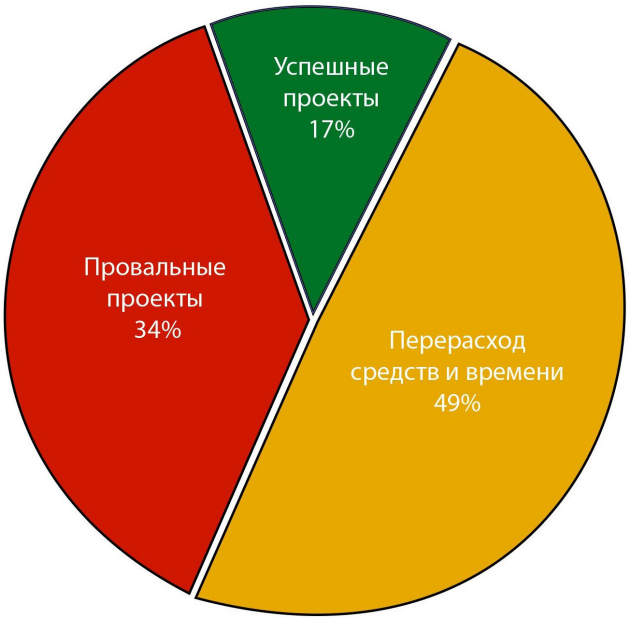


Рис. 2. Успешность проектов миграции и интеграции данных «Standish group»

Проекты по интеграции данных встречаются в настоящее время практически во всех отраслях. Но зачастую в процессе введения интеграции данных на предприятии может возникнуть ситуация, когда применен ошибочный подход построения проектного решения по интеграции данных. В результате этого предприятие теряет прибыль и время. По информации из источника «Standish group» (рис. 2) — 34 % проектов по миграции и интеграции данных являются провальными, 49 % в следствие реализации потребовали перерасход средств и времени, лишь 17 % проектов от общего числа оказывались успешными с точки зрения выделения средств и времени [1].

Следовательно, актуальна разработка проекта, позволяющего производить интеграцию данных наиболее удобно и эффективно для

разных видов бизнес-задач, так как нет типового и достаточно тиражируемого решения. Для этого необходимо проанализировать возможности, различия и конкретные применения того или иного метода интеграции данных из разнородных источников.

1. ETL/ ELT

ETL и ELT — широко используемые способы доставки данных из одного или нескольких источников в централизованную систему для удобства доступа и анализа. Обе этих методики состоят из этапов *extract* (*извлечения*), *transform* (*преобразования*) и *load* (*загрузки*). Разница заключается в последовательности действий, которое для потока интеграции многое меняет.

ETL — метод интеграции предназначенный для пакетного перемещения данных, чаще всего, по расписанию, в больших объемах, со сложными видами трансформации в централизованную систему для удобства доступа и анализа, рисунок 4.

Изначально задумка ETL была в том, чтобы уйти от SQL запросов и сделать процесс интеграции удобным для бизнес-пользователя поэтому инструменты ETL чаще всего удовлетворяют требованиям систем управления реляционными базами данных и/или традиционных хранилищ данных с поддержкой OLAP (online analytical processing, аналитической онлайн-обработки) [2]. Инструменты OLAP и запросы (SQL) требуют, чтобы массивы данных структурировались и стандартизировались при помощи серии преобразований, выполняемых до того, как данные попадут в хранилище.

ELT по сути меняет местами два последних этапа процесса ETL, то есть после извлечения из баз данных данные загружаются напрямую в центральный репозиторий, где происходят все преобразования. Промежуточная база данных отсутствует.

Такая методика стала возможной благодаря современным технологиям, позволяющим хранить и обрабатывать огромные объёмы данных в любом формате. В том числе и благодаря Apache Hadoop — открытому ПО, которое изначально создавалось для непрерывного получения данных из различных источников вне зависимости от их типа. Облачные хранилища данных наподобие Snowflake, Redshift и BigQuery также поддерживают ELT, поскольку они разделяют ресурсы хранения и вычислений, а также обладают высокой масштабируемостью [3].

Из-за того, что подходы построения работы с потоком данных у ETL и ELT различаются (рис. 3), выделяются области применения каждого из них.

ETL используется когда нужно обеспечить соответствие установленным стандартам защиты уязвимых данных клиентов, так как метод позволяет редактировать, шифровать и удалять уязвимые данные перед их передачей в хранилище данных. Следовательно, компаниям проще защищать данные и соблюдать различные стандарты комплаенса.

ELT в настоящее время — молодая методика, но ей отдается предпочтение в случаях необходимости работы с огромными массивами данных для целей бизнес-аналитики и аналитики big-data. Скорость интеграции данных — ключевое преимущество ELT по сравнению с ETL, поскольку целевая система способна выполнять преобразование и загрузку данных парал-

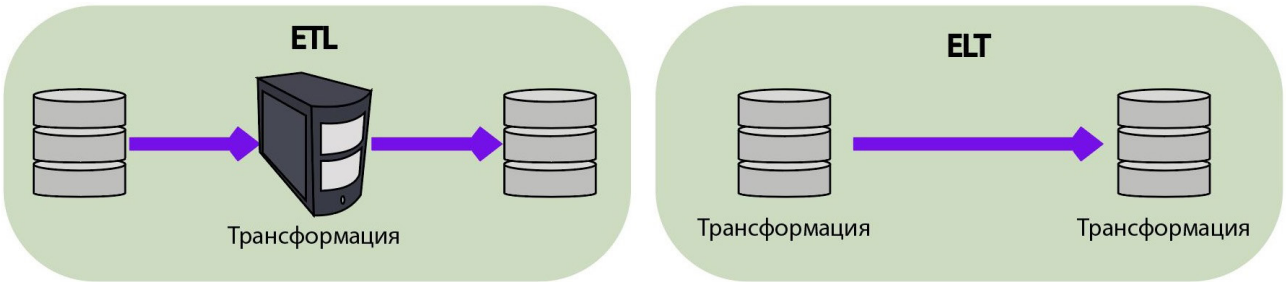


Рис. 3. Различие ETL и ELT

лельно [4]. Это, в свою очередь, позволяет генерировать аналитику практически в реальном времени. ELT можно использовать с облачными хранилищами и «озёрами» данных в отличие от ETL, который не может работать с «озерами» данных, а для работы с облачными хранилищами требуется внедрение промежуточных движков.

2. EAI

EAI, enterprise information integration (интеграция приложений) – метод интеграции предназначенный для работы с динамически изменяемыми данными. Цель EAI – упростить сложную цифровую архитектуру и повысить гибкость бизнеса [5]. Она связывает разрозненные системы для более эффективной совместной работы. Благодаря этой интеграции можно синхронизировать множество систем и инструментов в реальном для быстрого выполнения операционных задач, передавая небольшое количество данных практически без трансформации (рис. 4).

Основная задача данный интеграции по событию передавать данные между системами, в рамках одной транзакции. Для EAI объем данных важен, если одна транзакция будет занимать значительное количество времени, то это может быть критично для системы требующей синхронизации в реальном времени. Также у такого метода есть большое преимущество по сравнению с ETL — это гарантия доставки данных, т. е. в случае, когда приемник информации недоступен по какой-то причине сообщение с транзакцией не потеряется, а будет доставлено, когда приемник информации вновь будет доступен для использования [6].

3. EII

EII — enterprise information integration (интеграция информации) — метод интеграции в режиме реального времени несопоставимых типов данных из многочисленных источников как внутри, так и за пределами организации. Инструменты EII обеспечивают универсальный уровень доступа к данным. Отличительная особенность данного метода — работа по интеграции производится не техническим решением, а, в первую очередь, бизнес-пользователем, т. е. в большинстве своем это разовые операции, инициируемые бизнес-пользователем (рис. 4). Минусы данного вида интеграции в том, что если пользователь запросит большой объем дан-

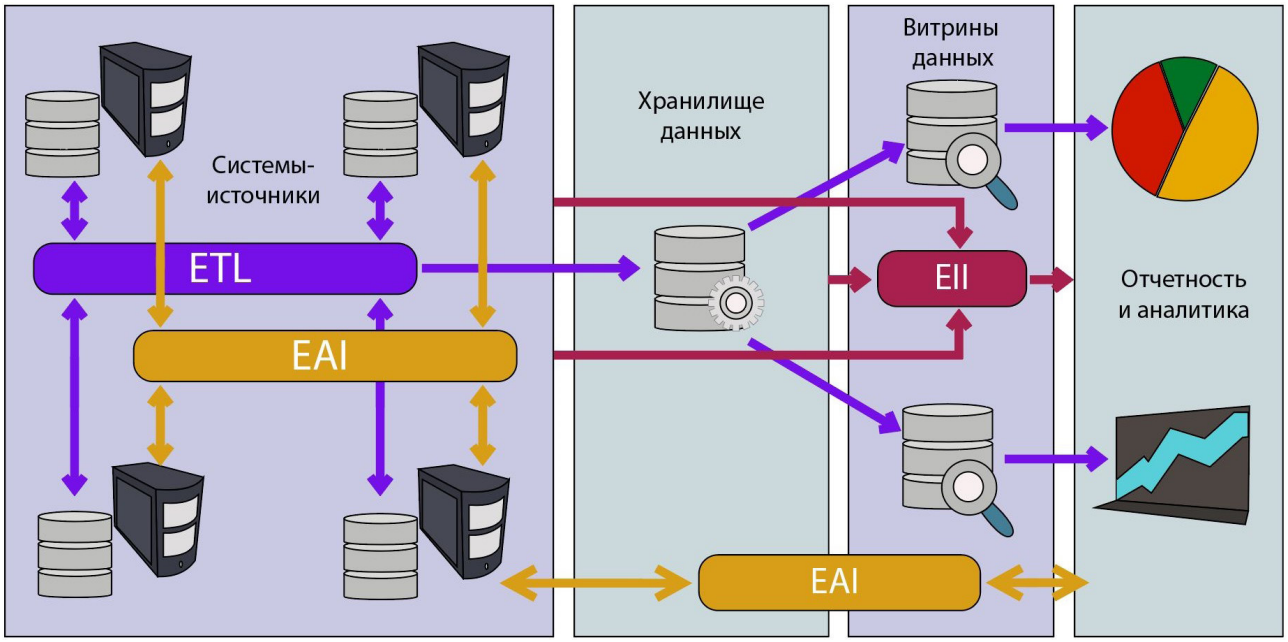


Рис. 4. Единая архитектура интеграции

ных, то это приведет к торможению работы систем источников. Вторая большая проблема — это то, что памяти сервера может не хватить на весь объем запрашиваемых данных ETL [7].

Технология EII использует распределенный запрос для сбора и интеграции информации из различных источников. Обычно такой запрос называют объединенным, или федеративным [8]. В этом случае запросы распределяются по источникам данных, а затем результаты их выполнения присоединяются другу к другу или объединяются. EII несколько отличается от других подходов интеграции. Так, EAI обычно передает сообщения от одного приложения к другому по концентратору или шине между системами. ETL использует физическое перемещение данных из одного местоположения в другое, создавая при этом в складах данных избыточные копии данных. Как правило, эти копируемые данные являются итоговыми данными и в этом случае детальные данные не доступны. В своей основе и EAI, и ETL — это технологии активной доставки. EII же является технологией извлечения информации, при которой объединенный запрос находит данные, необходимые для пользовательского приложения, и вставляет их в представление с пользовательским контекстом.

Заключение

Основные выводы по результатам проведенного анализа можно сформулировать следующим образом: успешность проектного решения по интеграции данных может быть достигнута только за счет грамотного использования всех трех методов ETL/ELT, EAI и EII. Использование метода ETL/ELT предлагается зачастую для создания интегрированного источника данных в рамках одной организации для дальнейшей работы с данными и аналитики. EAI позволит достичь максимальной надежности и скорости синхронизации работы систем с интегрированными данными. Использование метода интеграции данных EII будет предпочтительно для виртуальной интеграции информации между информационными системами, как правило, относящимися к разным организациям, запрещающим физическое копирование данных или предоставляющим ограниченный доступ к данным на платной основе [9]. Таким образом, на нижнем уровне данные интегрируются с помощью ETL, синхронизацию между системами в реальном времени позволяет достичь EAI, а на более высоком уровне интеграция осуществляется с использованием метода EII.

Литература

1. Исследование доли успешности проектов по миграции и интеграции данных : официальный сайт. – URL: https://www.standishgroup.com/sample_research_files/migrateNew.pdf (дата обращения: 24.11.2024).

2. Подход ETL : официальный сайт. – URL: https://aws.amazon.com/ru/what-is/etl/ (дата обращения: 24.11.2024).

3. ETL и ELT ключевые различия : официальный сайт. – URL: https://habr.com/ru/articles/695546/ (дата обращения: 24.11.2024).

4. Основные функции ETL-систем : официальный сайт. – URL: https://habr.com/ru/articles/248231/ (дата обращения: 24.11.2024).

5. Интеграция корпоративных приложений : официальный сайт. – URL: https://aws.amazon.com/ru/what-is/enterprise-application-integration/ (дата обращения: 24.11.2024).

6. Enterprise application integration : официальный сайт. – URL: https://www.tadviser.ru/index.php/Статья:Enterprise_application_integration (дата обращения: 24.11.2024).

7. Интеграция корпоративной информации: новое направление : официальный сайт. – URL: https://iso.ru/ru/press-center/journal/2038.phtml (дата обращения: 24.11.2024).

8. Способы интеграции корпоративных приложений : официальный сайт. – URL: https://studfile.net/preview/9985209/page:5/ (дата обращения: 24.11.2024).

9. Аналитические решения: понимание трех составляющих интеграции – EAI, EII, ETL : официальный сайт. – URL: https://iso.ru/ru/press-center/journal/2061.phtml (дата обращения: 24.11.2024).

УДК 81.32

ИССЛЕДОВАНИЕ ЗАВИСИМОСТИ ПРЕДЕЛЬНОГО СЛОВАРЯ ИНДИВИДА ОТ РАЗМЕРА ЕГО МЕТАКНИГИ

В. А. Оганисян

Воронежский государственный университет

Аннотация. Используя возможности современных технологий и применяя математические методы, лингвисты и другие исследователи получают уникальную возможность оценить идиолект конкретного человека. Чем больше корпус текстов, представляющих язык писателя, тем полнее и богаче информация о зависимости между размером корпуса и богатством словаря индивида. В статье описаны результаты исследования идиолекта В. И. Ленина.

Ключевые слова: лемматизация, корпус текстов, предельный размер словаря, коэффициент лексического разнообразия, идиолект, закон Ципфа, частотный словарь.

Введение

Количественный анализ текстов конкретного человека даёт возможность охарактеризовать его идиолект — индивидуальный вариант общенародного языка, находящий отражение во всем множестве текстов, порождённых индивидом. Чтобы проанализировать язык автора, в качестве предмета исследования выступает «Полное собрание сочинений» В. И. Ленина, состоящее из пятидесяти пяти томов. Из-за такого большого объёма данных было необходимо провести работу в несколько шагов. На каждом из них осуществляется прирост корпуса на примерно равную часть, сопровождаемый определением максимального размера активного словаря В. И. Ленина.

В работе использован метод моделирования и прогнозирования роста словаря индивида, предложенный в статьях А. А. Кретова, И. П. Половинкина и их соавторов [4–9].

Для проведения исследования были использован морфологический анализатор «MyStem», разработанный компанией «Яндекс», и возможности электронных таблиц MS-Excel, в которых производились расчёты и строились графики. С их помощью была проведена лемматизация (приведение слов к их начальным формам), получен частотный словарь автора и получены реальный и предельный размеры словарей индивида.

1. Проведение исследования и получение результатов

Важной характеристикой для этого исследования является *коэффициент лексического разнообразия* (КЛР), в основе которого лежит отношение количества лемм к количеству их употреблений в тексте. Этот показатель представляет собой количественную характеристику текста, отражающую степень богатства словаря при построении текста заданной длины.

Для получения реального и предельного размера словаря В.И. Ленина необходимы суммарные (кумулятивные) значения размера метакниги и словаря (табл. 1).

Таблица 1

Прирост новых слов и покрываемого ими текста

Том	Год	Длина	Слов	КоЛеР	ДлКум	СлКум	КуКоЛеР
Т01	1893–1894	108604	7090	0,0653	108604	7090	0,0653
Т02	1895–1897	104156	7435	0,0714	212760	10124	0,0476
Т03	1896–1900	104507	6499	0,0622	317267	12057	0,0380

T04	1898–1901 апрель	96831	7177	0,0741	414098	13716	0,0331
T05	1901 май – 1901 декабрь	78779	7392	0,0938	492877	15307	0,0311
T06	1902 январь – 1902 август	87138	7031	0,0807	580015	16455	0,0284
T07	1902 сентябрь –1903 сентябрь	67412	6358	0,0943	647427	17260	0,0267
T08	1903 сентябрь – 1904 сентябрь	91709	6419	0,0700	739136	18171	0,0246
T09	1904 июль – 1905 март	73290	6651	0,0907	812426	18995	0,0234
T10	1905 март – 1905 июнь	66348	6001	0,0904	878774	19560	0,0223
T11	1905 июль – 1905 октябрь	83662	6516	0,0779	962436	20190	0,0210
T12	1905 октябрь – 1906 апрель	79341	6515	0,0821	1041777	20759	0,0199
T13	1906 май – 1906 сентябрь	84668	6731	0,0795	1126445	21390	0,0190
T14	1906 сентябрь – 1907 февраль	85065	6416	0,0754	1211510	21930	0,0181
T15	1907 февраль – 1907 июнь	81166	6338	0,0781	1292676	22404	0,0173
T16	1907 июнь – 1908 март	100383	7778	0,0775	1393059	23146	0,0166
T17	1908 март – 1909 июнь	92905	7386	0,0795	1485964	23693	0,0159
T18	1908–1909	88082	5960	0,0677	1574046	24616	0,0156
T19	1909 июнь – 1910 октябрь	87222	6638	0,0761	1661268	25072	0,0151
T20	1910 ноябрь – 1911 ноябрь	87962	7287	0,0828	1749230	25628	0,0147
T21	1911 декабрь – 1912 июль	98711	7589	0,0769	1847941	26151	0,0142
T22	1912 июль – 1913 февраль	71716	6977	0,0973	1919657	26606	0,0139
T23	1913 март – 1913 сентябрь	79352	7534	0,094944	1999009	27154	0,01358
T24	1913 сентябрь – 1914 март	70025	6329	0,090382	2069034	27511	0,0133
T25	1914 март – 1914 июль	85754	6754	0,07876	2154788	27852	0,01293
T26	1914 Июль – 1915 август	73282	6240	0,085151	2228070	28218	0,01266
T27	1915 август – 1916 июнь	85504	6649	0,077762	2313574	28742	0,01242
T28	1915–1916	113221	11533	0,101863	2426795	32440	0,01336
T29	1895–1916	122894	8791	0,071533	2549689	33670	0,01321
T30	1916 июль – 1917 февраль	80682	6232	0,077242	2630371	33944	0,0129
T31	1917 март – 1917 апрель	86941	5805	0,066769	2717312	34197	0,01258
T32	1917 май – 1917 июль	82557	6215	0,075281	2799869	34452	0,01229
T33	1917–1918	50706	4418	0,08713	2850575	34632	0,01214
T34	1917 июль – 1917 октябрь	94706	6816	0,07197	2945281	34911	0,01185
T35	1917 октябрь – 1918 март	72738	6035	0,082969	3018019	35218	0,01167
T36	1918 март – 1918 июль	115722	7039	0,060827	3133741	35541	0,01134
T37	1918 июль –1919 март	116168	7299	0,062831	3249909	35867	0,01104
T38	1919 март – 1919 июнь	91509	6135	0,067043	3341418	36059	0,01079
T39	1919 июнь – 1919 декабрь	96741	6596	0,068182	3438159	36353	0,01057
T40	1919 декабрь – 1920 апрель	74060	5853	0,079031	3512219	36597	0,01042
T41	1920 май – 1920 ноябрь	94486	6656	0,070444	3606705	36846	0,01022
T42	1920 ноябрь – 1921 март	86724	6254	0,072114	3693429	37181	0,01007
T43	1921 март – 1921 июнь	83265	5861	0,07039	3776694	37469	0,00992
T44	1921 июнь – 1922 март	78832	6387	0,08102	3855526	37837	0,00981
T45	1922 март – 1923 март	82006	6495	0,079202	3937532	38203	0,0097
T46	1893–1904	89469	7032	0,078597	4027001	38873	0,00965
T47	1905–1910 ноябрь	52703	5408	0,102613	4079704	39188	0,00961
T48	1910 ноябрь – 1914 июль	58398	6039	0,103411	4138102	39635	0,00958
T49	1914 август – 1917 октябрь	72950	6125	0,083962	4211052	40071	0,00952
T50	1917 октябрь – 1919 июнь	43559	5499	0,126243	4254611	40682	0,00956

T51	1919 июль – 1920 ноябрь	38622	5385	0,139428	4293233	41266	0,00961
T52	1920 ноябрь – 1921 июнь	37747	5036	0,133415	4330980	41726	0,00963
T53	1921 июнь – 1921 ноябрь	43798	5019	0,114594	4374778	42101	0,00962
T54	1921 ноябрь – 1923 март	60697	6180	0,101817	4435475	42496	0,00958
T55	1893–1922	68707	5818	0,084678	4504182	43138	0,00958

В табл. 1: N — текущее значение размера словаря; ΔN — приращение словаря, то есть количество новых уникальных слов при добавлении новых текстов в корпус; M — текущее значение размера корпуса; ΔM — приращение размера корпуса, то есть количество словоупотреблений в добавляемом в корпус тексте; YTTR — текущее значение КЛР

Достижение предельного размера словаря активной лексики В. И. Ленина. достигается при приращении словаря близком к нулю, когда КЛР в процессе увеличения корпуса стремится к нулю, но не принимает нулевого значения, поскольку размер словаря — величина положительная. В исследовании используется линия тренда — логарифмическая зависимость. На каждом этапе были получены логарифмические функции, которые было необходимо приравнять нулю. Таким образом, был прогнозно оценён предельный размер словаря писателя.

Исследование первоначально проводилось в пять шагов, каждый из которых включал добавление равного количества текста автора (по одиннадцать томов). Впоследствии, для более детального анализа, метакнига (корпус текстов) была разделена на меньшие подкорпусы, состоящие из пяти томов.

Таблица 2

Полученные результаты на разных шагах исследования

Тома	Реальный словарь (РС)	Предельный словарь (ПС)	Среднее значение (СЗ)	Объем текста, при котором достигается ПС
1–5т.	15 307	29 161 – 29 475	29 318	1 833 503
1–10т.	19 560	32 637 – 33 918	33 277	2 433 886
ШАГ 1. 11 Т.	20 190	28 795 – 31 067	29 931	1 914 563
1–15т.	22 404	31 335 – 36 595	33 965	3 883 402
1–20т.	25 628	33 470 – 34 484	33 977	3 230 784
ШАГ 2. 22 Т.	26 606	40 026 – 43 655	41 840	5 560 786
1–25т.	27 852	34 462 – 36 214	35 338	3 330 889
1–30т.	33 944	40 505 – 42 991	41 748	4 562 281
ШАГ 3. 33 Т.	34 632	45 170 – 46 630	45 900	5 788 759
1–35т.	35 218	40 113 – 49 530	44 821	4 257 896
1–40т.	36 597	43 415 – 50 179	46 797	6 841 117
ШАГ 4. 44 Т.	37 837	47 373 – 48 300	47 836	6 107 328
1–45т.	38 203	42 394 – 45 184	43 789	4 802 347
ШАГ 5. 55 Т.	43 138	48 706 – 49 020	48 863	6 367 176

В результате, оценка предельного размера словаря В. И. Ленина преимущественно постепенно растёт по мере увеличения размера метакниги.

Заключение

Таким образом, наибольшее значение предельного словаря В. И. Ленина наблюдается на последнем шаге исследования, который проводился на основе корпуса из пятидесяти томов «Полного собрания сочинений» (1893-1923). На этом шаге предельный размер словаря активной лексики индивида находится в интервале *48 706 – 49 020*, т. е. равен *48 863* словам.

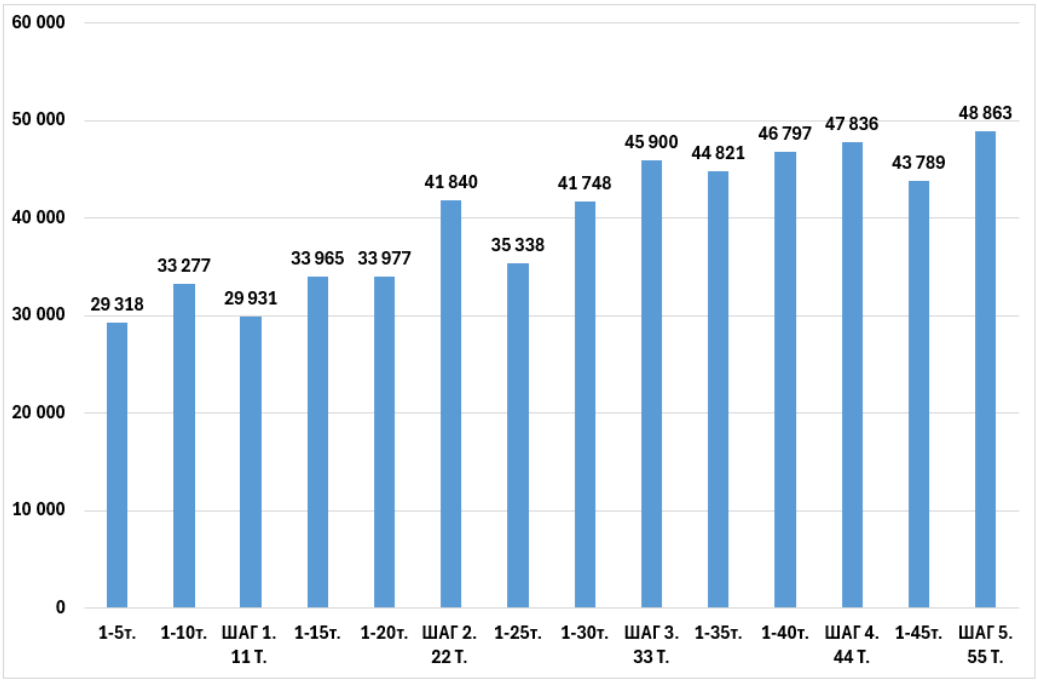


Рис. 1. Динамика роста значения предельного словаря активной лексики В. И. Ленина

Литература

1. *Алексеев П. М.* К основам статистической лексикографии. – В кн. Проблема слова и словосочетания. – ЛГИИ, 1980. – С. 95–105.

2. *Алексеев П. М.* Частотные словари и приемы их составления. Статистика речи. – Л. : Наука, 1968. – С. 61–63.

3. *Арапов М. В., Шрейдер Ю. А.* Закон Ципфа и принцип диссимметрии системы // Семиотика и информатика. – 1978. – Вып. 10. – С. 74–95.

4. *Кретов А. А.* Лексическое богатство словаря В. В. Набокова / А. А. Кретов, И. П. Половинкин, Н. А. Касымова, М. В. Половинкина // Электронный научный журнал «Квантитативная филология». – Смоленск : Смоленский Центр квантитативной филологии, 2021. – № 1. – С. 39–48. DOI 10.35785/0000-0000-2021-1-39-48.

5. *Кретов А. А.* О предельном размере словаря и фрактальной размерности метакниги М.Е. Салтыкова-Щедрина / А. А. Кретов, М. В. Половинкина, И. П. Половинкин, Н. А. Касымова // Информатика: проблемы, методы, технологии: сборник материалов XXII международной научно-методической конференции / под редакцией Д. Н. Борисова; Воронеж, Воронежский государственный университет, 10-12 февраля 2022 г. – Воронеж : «ВЭЛБОРН», 2022. – С. 1146–1154.

6. *Кретов А. А., Половинкин И. П., Ломец М. В.* Абсолютное и относительное «богатство словаря» на примере произведений Л. Н. Толстого // Математика и междисциплинарные исследования – 2020. – С. 200–203.

7. *Кретов А. А., Половинкина М. В., Половинкин И. П., Ломец М. В.* О моделировании изменений языка // Современные методы теории функций и смежные проблемы. – 2021. – С. 173–174.

8. *Кретов А. А., Половинкина М. В., Половинкин И. П., Ломец М. В.* О некоторых количественных характеристиках фрактальности в языке // Информатика: проблемы, методы, технологии. – 2020. – С. 1627–1634.

9. *Кретов А. А., Ломец М. В., Половинкин И. П.* Возможный алгоритм вычисления предельного размера словаря писателя // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2021. – С. 133–145.

10. *Папп Ф.* Количественный анализ словарной структуры некоторых русских текстов // Вопросы языкознания. – 1961. – № 6. – С. 93–100.

11. *Папп Ф.* О машинной обработке одноязычных словарей // Научно-техническая информация. – 1969. – № 3, Сер. 2. – С. 53–56.

УДК 004.056

ИНФОРМАЦИОННО-ПСИХОЛОГИЧЕСКОЕ ВОЗДЕЙСТВИЕ НА РАБОТНИКОВ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

Ю. А. Паршенкова

МИРЭА – Российский технологический университет

Аннотация. В нынешних реалиях критическая информационная инфраструктура играет ключевую роль в обеспечении функционирования различных сфер деятельности. Её безопасность и устойчивость являются важными аспектами национальной безопасности. В связи с этим возникает необходимость проведения тестирования обслуживающего персонала этой инфраструктуры с использованием специальных информационно-психологических воздействий (ИПВ). Это актуальное и важное направление аудита ИБ, поскольку человек является ключевым звеном системы КИИ, но при этом он уязвим для информационно-психологического воздействия, которое может привести к нарушению политики ИБ.

Ключевые слова: критическая информационная инфраструктура, информационная безопасность, политика информационной безопасности, объекты КИИ, информационно-психологическое воздействие, аудит ИБ, лицо, принимающее решение, тестирование ИБ.

Введение

Информационно-психологическое воздействие — это процесс воздействия на разум и чувства человека или группы людей с целью изменения их восприятия реальности, поведенческих установок и действий. Такое воздействие может быть направлено на индивидуальное, групповое, массовое и общественное сознание и иметь как положительный, так и отрицательный эффект [1].

1. Общее представление об информационно-психологическом воздействии.

Рассмотрим варианты информационно-психологического воздействия на работников КИИ.

Воздействие информационно-психологических средств на сотрудников объектов критической информационной инфраструктуры осуществляется с целью:

– жертву вынуждают нарушить политику безопасности предприятия;

– злоумышленник вынуждает жертву нарушить конфиденциальность/ целостность/ доступность рабочей информации;

– нарушить социально-психологические процессы привычного функционирования коллектива объекта КИИ;

– нарушить работу объектов критической информационной инфраструктуры, подорвав доверие к лицам, принимающим решения, и создав условия, при которых отдельные люди и группы будут отказываться выполнять приказы и распоряжения вышестоящего руководства, склоняя их к саботажу [2].

Объектами информационно-психологического воздействия могут быть:

– Личность: конкретные представители органов управления, правопорядка и безопасности, работники организаций, учреждений и предприятий, чья деятельность влияет на объекты КИИ.

– Система формирования и функционирования духовной сферы, общественного сознания и общественного мнения: образование, подготовка кадров, распространение социально значимой информации и социокультурных ценностей.

– Социальные группы и объединения людей: политические, профессиональные, национально-этнические, демографические, религиозные группы и другие.

– Органы власти, государственного и военного управления.

– Общественные и политические организации, общественно-политические движения и объединения граждан.

– Силовые министерства и ведомства.

– Государство и общество, население страны в целом как социально-историческая общность людей.

Субъектами информационно-психологического воздействия являются, прежде всего, силы информационных операций враждебных стран, а также люди и группы, занимающиеся диссидентской и подрывной деятельностью или сотрудничающие со спецслужбами других государств [3]. В зависимости от поставленных задач, воздействие осуществляется в следующих направлениях:

– воздействие на работников с целью побуждения их к некоторым действиям, влияющим на работу предприятия;

– изменения восприятия получаемой информации, формируя искаженное представление человека о происходящих процессах;

– воздействие на переживания и внутренние эмоции людей;

– создание условия для сотрудничества или конфликта между людьми.

Основные принципы разработки и реализации ИПВ:

– подготовка ИПВ всегда начинается заранее. Обязательно учитываются индивидуальные особенности объекта, подвергающегося обработке. Воздействие не афишируется;

– планирование и реализация ИПВ происходит с учётом найденных недостатков. Причем, речь идет как и о недостатках в политике безопасности предприятия, так и о уязвимостях психологического состоянии отдельных личностей и групп, наиболее подверженных к воздействию;

– разные ИПВ должны иметь корреляцию между собой для достижения общей цели;

– силы и средства ИПВ применяются массово, комплексно и разнообразно.

Мероприятия ИПВ направлены на внедрение определённых идей, убеждений или лозунгов в сознание отдельных людей и групп, чтобы вызвать недоверие к руководству, осознание неблагоприятного положения, угрозы жизни и благополучию близких, а также ввести в заблуждение и склонить к сотрудничеству. При разработке и проведении ИПВ злоумышленники используют следующие основные принципы:

– квалифицированное руководство — чёткое представление руководителей о целях и задачах операции;

– правдоподобие — действия, восприятие и объяснения, вызывающие доверие у целевой аудитории;

– доступность — учёт психологических особенностей, культуры, образа жизни, социальных взаимоотношений целевой аудитории, лингвистических, исторических, религиозных, природных и других объективных факторов;

– диалог — ИПВ в форме обмена мнениями способствует взаимопониманию и установлению доверительных отношений с объектом воздействия;

– масштабность — отсутствие временных и пространственных ограничений на ИПВ;

– согласованность — согласованные и интегрированные действия на всех уровнях иерархии по единому замыслу и плану;

– целенаправленность — ориентация всей совокупности ИПВ на получение конкретного результата;

– непрерывность — непрерывный процесс ИПВ, требующий постоянной обратной связи между планированием и действиями, а также анализом и оценкой результатов.

В процессе проведения отдельного ИПВ можно выделить три этапа:

– операциональный, когда субъект воздействует на объект;

– процессуальный, когда объект принимает или не принимает это воздействие;

– заключительный, когда проявляются реакции объекта на перестройку его психики.

Перестройка психики под воздействием ИПВ может быть разной по масштабу и продолжительности. ИПВ может осуществляться с использованием разных методов и средств, многие из которых, постоянно развиваясь и совершенствуясь, превратились в сложные технологии воздействия на психику людей. К примеру, наиболее распространенной технологией на сегодняшний день является воздействие на сознание людей посредством телевизионной техники и сети Интернет [4].

Психология как наука решает следующие задачи в рамках подготовки и применения ИПВ:

– указывает на особенности человеческой и групповой психики, которые целесообразно подвергнуть воздействию;

– предоставляет эффективные методы оценки психологического состояния отдельных лиц и групп;

– даёт рекомендации специалистам, использующим ИПВ, по планированию операций;

– разрабатывает критерии и методы оценки эффективности ИПВ для людей.

Психологи западных стран основываются на достижениях различных психологических школ при создании научной базы ИПВ. Основные положения, на которые они опираются, включают:

– решающую роль бессознательного в определении поведения человека и функционировании механизмов психологической защиты;

– способы преодоления психологической защиты (психоанализ);

– рефлекторное закрепление восприятий, переживаний и действий (бихевиоризм, нейролингвистическое программирование).

Когнитивная психология подчёркивает роль ментальных схем в восприятии человеком окружающего мира и информации, а гуманистическая психология изучает структуру и динамику человеческих потребностей. Психология помогает организаторам ИПВ определить уязвимые места в психологическом состоянии человека, а также научно обосновать тактику психологического воздействия на него. Используются политические, национальные, социальные и религиозные разногласия, распространение слухов и дезинформации для достижения злоумышленником поставленных целей.

Заключение

Подводя итоги статьи, можно сделать следующие краткие выводы:

– Для оценки уровня защищённости объектов КИИ в организационной и психологической сферах можно использовать тестирование отдельных лиц и персонала КИИ с помощью специальных средств и методов ИПВ. Эти средства и методы, а также сценарии их применения должны соответствовать аналогичным средствам, методам и сценариям потенциального противника [5].

– Классификация средств и методов специальных ИПВ для тестирования объектов КИИ включает информационные, лингвистические, психотронные, психофизические, психотропные и соматопсихологические методы.

– Выявлять уязвимости в организационной и психологической сферах. Разрабатывать предложения по повышению уровня информационно-психологической безопасности для обеспечения стабильности объектов КИИ в условиях информационного противоборства.

– Человек в системе КИИ является ключевым элементом так как к нему стекаются информационные потоки и он принимает окончательные решения. Однако человек также является сла-

бым звеном, подверженным воздействию ИПВ, которые могут привести к нарушению политики информационной безопасности. Аудит ИБ объектов КИИ должен включать проверку психологической устойчивости персонала с использованием специальных средств и методов ИПВ.

Литература

1. *Баришполец В. А.* Информационно- психологическая безопасность: основные положения // Информационные технологии. – 2003. – Т. 3, № 2. – С. 69–104.

2. *Монахов М. Ю., Полянский Д. А., Монахов Ю. М., Семенова И. И.* Концепция управления процессом обеспечения достоверности информации в ИТКС в условиях информационного противодействия // Фундаментальные исследования. – 2014. – № 9. – С. 2397–2402.

3. *Иванова Н. В., Коробулина О. Ю.* Аудит информационной безопасности. – СПб. : ПГУПС, 2011. – 57 с.

4. *Макаренко С. И., Смирнов Г. Е.* Методика обоснования тестовых информационно-технических воздействий, обеспечивающих рациональную полноту аудита защищенности объекта критической информационной инфраструктуры // Вопросы кибербезопасности. – 2021. – № 6 (46). – С. 12–25. – DOI: 10.21681/2311-3456-2021-6-12-25

5. *Максимова Е. А.* Инфраструктурный деструктивизм субъектов критической информационной инфраструктуры [монография] / Е. А. Максимова // Москва – Волгоград : Волгоградский государственный университет, 2021. – 181 с. – EDN ZZTOKE

УДК 004

МЕТОДЫ ОПТИМИЗАЦИИ SQL-ЗАПРОСОВ К БАЗАМ ДАННЫХ

А. С. Свиридов

Воронежский государственный университет

Аннотация. В научной статье рассматриваются методы оптимизации запросов к базам данных. В ходе работы будут определены методы оптимизации, порядок выполнения запросов, проанализировано влияние разных запросов на систему, дано понимание, как оценивается оптимизация и как работает план выполнения запросов. В качестве примера рассматривается упрощённая модель хранения документов. Рассмотрены следующие технологии: СУБД PostgreSQL, ERwin Data Modeler — компьютерная программа для проектирования и документирования баз данных.
Ключевые слова: база данных, PostgreSQL, ERwin Data Modeler, оптимизация, SQL-запрос, join, ER-диаграмма, план запросов, индкесы, DML.

Введение

Существуют приложения, которые для корректной своей работы опираются на данные, полученные от действий пользователей или внешних ресурсов. Эти данные сохраняются в базе данных. Базы данных, лежащие в основе таких приложений, подвергаются высокой нагрузке, что может привести к снижению производительности, задержкам и недоступности системы. Оптимизация запросов к базе данных является критически важным аспектом обеспечения стабильной работы и высокой производительности приложения. По мере роста объема информации и пользователей приложения должны гарантировать быструю и качественную работу, так как возрастает нагрузка как на базу данных, так и на приложение в целом. В этой статье рассмотрим планы выполнения запросов, варианты оптимизации самих запросов на примере модели хранения документов, загруженные через приложение.

1. Определение сферы решаемой задачи

В качестве примера для демонстрации оптимизации запросов выбрана модель хранения документов. Эта модель является актуальной, так как многие государственные организации (например, налоговая служба), запрашивают необходимые документы у других частных компаний, а электронное представление документов облегчит работу.

При загрузке документа данные должны сохраняться в базу данных. Документ принадлежит к определённому типу, организации, важно учитывать период документа и информацию о пользователе, который загрузил его. Сами данные относятся к определённому разделу. Например, к декларации по налогу на имущество.

В качестве запроса для оптимизации выступает выборка данных конкретных документов с типом «IMUD_DECLARATION_509», которые загружены в периоде «2022, 1 квартал».

Для того, чтобы приступить к реализации, необходимо спроектировать саму модель данных, ориентируясь на нормальные формы [1]. В качестве программы проектирования выбран ERwin Data Modeler. ERwin Data Modeler — это программное обеспечение для моделирования баз данных, разработанное компанией Micro Focus. На рис. 1 представлена ER-диаграмма «Хранение документов».

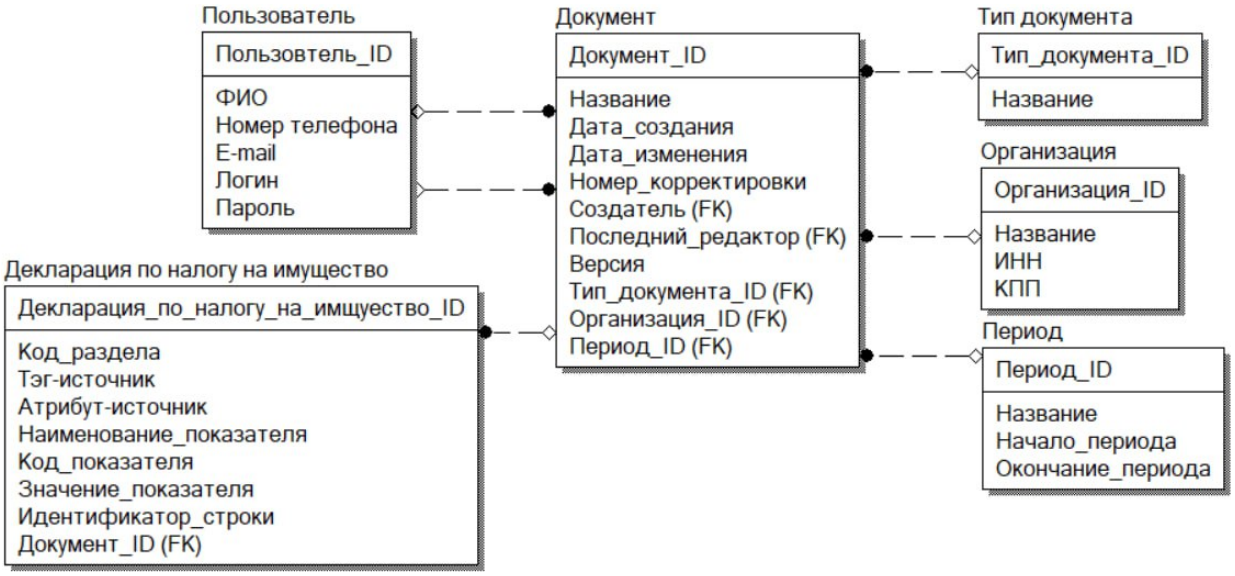


Рис. 1. ER-диаграмма «Хранение документов»

2. База данных

Для хранения данных, к которым будут написаны запросы, был выбран PostgreSQL — объектно-реляционная база данных [2]. Эта СУБД представляет все данные в виде объектов, имеет иерархию ролей. PostgreSQL имеет следующие преимущества:

- поддержка баз данных неограниченного размера;
- мощные и надёжные механизмы транзакций и репликации;
- лёгкая расширяемость.

Ориентируясь на перечисленные преимущества, PostgresSQL больше подходит для выбранной темы, чем другие СУБД. Например, MySQL, который недостаточно надёжен при работе с данными.

3. Оптимизация запросов

Для извлечения данных из базы данных используется select. Select — оператор запроса в языке SQL, который относится к подмножеству DML, возвращающий набор данных из базы данных [3].

Запрос для выборка данных конкретных документов с типом «IMUD_DECLARATION_509», которые загружены в периоде «2022, 1 квартал» выглядит следующим образом:

```
select *
from dm_imud_decl did
where did.doc_id in (select distinct d.doc_id
                     from documents d,
                          document_types dt,
                          tax_periods tp
                     where d.doc_type_id = dt.doc_type_id
                       and dt.doc_type_code = 'IMUD_DECLARATION_509'
                       and tp.tax_period_name = '2022, 1 квартал');
```

В этом запросе данные документа при заданных условиях находятся при помощи подзапроса. Формируются идентификаторы документов, по которым ищем нужные данные, проверяя, входят они по полю doc_id в сформированный список или нет. План выполнения запроса представлен на рис. 2.

QUERY PLAN
Hash Join (cost=38.47..1931.17 rows=65 width=7264)
Hash Cond: (did.doc_id = d.doc_id)
-> Seq Scan on dm_imud_decl did (cost=0.00..1887.69 rows=1869 width=7264)
-> Hash (cost=38.38..38.38 rows=7 width=8)
-> Unique (cost=38.28..38.31 rows=7 width=8)
-> Sort (cost=38.28..38.30 rows=7 width=8)
Sort Key: d.doc_id
-> Nested Loop (cost=10.14..38.18 rows=7 width=8)
-> Seq Scan on tax_periods tp (cost=0.00..1.16 rows=1 width=0)
Filter: ((tax_period_name)::text = '2022, 1 квартал'::text)
-> Hash Join (cost=10.14..36.95 rows=7 width=8)
Hash Cond: (d.doc_type_id = dt.doc_type_id)
-> Seq Scan on documents d (cost=0.00..24.63 rows=563 width=16)
-> Hash (cost=10.12..10.12 rows=1 width=8)
-> Seq Scan on document_types dt (cost=0.00..10.12 rows=1 width=8)
Filter: ((doc_type_code)::text = 'IMUD_DECLARATION_509'::text)

Рис. 2. План выполнения запроса

План выполнения запроса можно оценить по стоимости выполнения, за которое отвечает значение cost. В этом примере происходит полное сканирование таблиц, что при больших данных приведёт к более низкому выполнению по времени. В плане выполнения запроса встречается sort и unique. Эти операции могут быть избыточны для выборки данных конкретных документов. Можно оптимизировать запрос, используя оператор объединения таблиц и добавив индексы на таблицы, что позволит ускорить выполнение. Индексы — это специальные структуры данных, которые значительно ускоряют поиск данных в таблицах [4].

Оптимизированный запрос выглядит следующим образом:

```
select did.*
from udl.dm_imud_decl did
         join wdl.documents d on did.doc_id = d.doc_id
         join wdl.document_types dt on d.doc_type_id = dt.doc_type_id
         join wdl.tax_periods tp on d.period_id = tp.tax_period_id
where dt.doc_type_code = ‘IMUD_DECLARATION_509’
  and tp.tax_period_name = ‘2022, 1 квартал’;
```

В оптимизированном запросе не используется подзапрос. Вместо него фигурирует join — оператор для внутреннего соединения таблиц. После объединения выполняется оператор where, который позволяет применить условия фильтрации, затем выполняется конкретная выборка данных. На рис. 3 представлен план выполнения запроса.

По стоимости выполнения оптимизированный запрос значительно лучше первого, хотя есть вложенные циклы, которые потребуют дальнейшей оптимизации с увеличением объёма данных. При этом использование индексов и избавление от подзапроса помогло ускорить запрос. Применились на практике следующие методы оптимизации:

- ограничение использования подзапросов;
- использование индексов;
- извлечение только тех данных, которые нужны;
- ограничение использования сортировок.

QUERY PLAN
Nested Loop (cost=8.60..275.75 rows=31 width=190)
-> Nested Loop (cost=8.31..34.89 rows=1 width=8)
-> Hash Join (cost=8.17..34.32 rows=3 width=16)
Hash Cond: (d.doc_type_id = dt.doc_type_id)
-> Seq Scan on documents d (cost=0.00..24.63 rows=563 width=24)
-> Hash (cost=8.16..8.16 rows=1 width=8)
-> Index Scan using uk_document_types on document_types dt (cost=0.14..8.16 rows=1 width=8)
Index Cond: ((doc_type_code)::text = 'IMUD_DECLARATION_509'::text)
-> Index Scan using tax_periods_u1 on tax_periods tp (cost=0.14..0.18 rows=1 width=8)
Index Cond: (tax_period_id = d.period_id)
Filter: ((tax_period_name)::text = '2022, 1 квартал'::text)
-> Index Scan using ix_dm_imud_decl on dm_imud_decl did (cost=0.29..158.68 rows=8218 width=190)
Index Cond: (doc_id = d.doc_id)

Рис. 3. План выполнения оптимизированного запроса

Заключение

В данной статье были сравнены планы выполнения запросов, варианты оптимизации самих запросов на примере модели хранения документов.

Оптимизация запросов к базам данных — сложная задача, требующая комплексного подхода. Анализ производительности, оптимизация схемы, SQL-запросов — все эти аспекты должны быть учтены для достижения наилучших результатов в условиях постоянно возрастающей нагрузки на современные приложения. Постоянный мониторинг производительности и профилирование запросов, а также гибкое применение описанных методов позволят поддерживать стабильную и эффективную работу приложения.

Литература

1. Руководство по проектированию реляционных баз данных. Режим доступа: свободный. https://habr.com/ru/post/193136. (Дата обращения 23.11.2024)

2. PostgreSQL : официальный сайт. – URL: https://www.postgresql.org/ (дата обращения: 23.11.2024).

3. *Грофф Д.* SQL: Полное руководство / Д. Грофф, П. Вайнберг. – Москва : Вильямс, 2015. – 960 с.

4. PostgreSQL : Документация: 15: 1.2. Основы архитектуры : Компания Postgres Professional. – URL: https://postgrespro.ru/docs/postgresql/15/tutorial-arch (дата обращения: 24.11.2024).

УДК 65.016.7

РАЗРАБОТКА АЛГОРИТМА ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ УПРАВЛЕНЧЕСКИХ РЕШЕНИЙ В МАТРИЧНОЙ ОРГАНИЗАЦИОННОЙ СИСТЕМЕ НЕПРЕРЫВНОЙ ОЦЕНКИ ОБУЧАЮЩИХСЯ

В. С. Тарасов

МИРЭА – Российский технологический университет

Аннотация. В данной статье рассматривается интеллектуальная поддержка принятия управленческих решений, а также предложен алгоритм интеллектуальной поддержки принятия управленческих решений в матричной организационной системе оценки обучающихся. Используется математическая модель для определения «рабочей точки», которая показывает уровень сформированности набора компетенций у обучающихся. Определяется, какие факторы влияют на уровень сформированности компетенций. Используется уравнение линеаризации для определения степени влияния факторов на изменение «рабочей точки». На основе полученных данных о степени влияния факторов на уровень сформированности компетенций, принимаются управленческие решения для оптимизации процесса обучения. Данный алгоритм позволяет выявить проблемы в образовательном процессе и грамотно подойти к решению данных проблем.
Ключевые слова: алгоритм, интеллектуальная поддержка, оценка, организационная система, управленческие решения, математические компетенции.

Введение

На сегодняшний день управление образовательным процессом требует от руководящего состава оперативности и грамотных управленческих решений, которые без матричной организационной системы — в недостаточной степени эффективны. В связи с чем, руководители в сфере образования сталкиваются с необходимостью принимать множество решений, опираясь на результаты обработки больших объемов данных. В виду чего, возникает крайняя необходимость в разработке систем интеллектуальной поддержки процесса принятия управленческих решений на базе организационных систем.

1. Поддержка принятия управленческих решений в образовательной сфере

Вопроса о принятии управленческих решений в образовательной сфере на сегодняшний день, в той или иной степени, опирается на возможности анализа диагностических показателей. И, для эффективного управления профессиональной подготовки и обоснования компетентных управленческих решений, необходима формулировка вопросов по определению уровня и алгоритма интеллектуальной поддержки в процессе принятия управленческих решений.

В связи с чем, становится актуальной разработка механизма, которая предоставляет собой локализацию управленческих решений в профессиональной подготовке [2]. Отметим, что разработка алгоритма управления в матричной организационной системе, которая представляет собой метод двойного управления, занимает важное место в решении управленческих задач. Однако, несмотря на достижения математических теорий, остаются нерешённые вопросы, с которыми приходится сталкиваться при управлении решений. Стоит отметить, что на практике управленческих решений возникает множество классических задач, методы математических решений которых, основываются на предположениях, не соответствующих современным особенностям математических моделей. К примеру, высокий уровень неопределенности может затруднять применение стандартных математических моделей. В связи с чем, необходимы

новые, более совершенные подходы и методы, учитывающие непрерывную оценку при матричной организационной системе, которая подразумевает под собой использование оценок, исходя их выполненных заданий, форм фиксирования оценок, педагогических приёмов и механизмов анализа полученных результатов.

Стоит отметить, что алгоритмы интеллектуальной поддержки управленческих решений в матричной организационной системе, изначально были направлены на возможность осуществления обратной связи с целью выявления уровня навыков учащихся. В связи с чем, уровень обоснования коррекционных методов образовательного процесса, с точки зрения математических решений, должен отвечать сформированности компетенций, отвечающих за формирование знаний у обучающихся [3]. При этом, математическая компетенция не может быть сформулирована на одном заданном уровне, однако её, в зависимости от характера, можно разделить на группы:

– в зависимости от сложности математических теорий;
– в зависимости от способности демонстрации к абстракции и формальных теорий;
– в зависимости от областей математики и других предметных областей;
– в зависимости от уровня сложности и математических утверждений.

Таким образом, стоит выделить методы и критерии определения трех уровней математических компетенций (пороговый, продвинутый, высокий), которые служат критериями отбора заданий, как предоставлено в табл. 1 [1].

Таблица 1

Критерии определения уровней математических компетенций

Уровень	Критерий определения	Критерий отбора
I пороговый	Знать и выявлять ключевые определения и проблемы математики в рамках учебной информации, уметь находить и репродуцировать необходимую информацию	Задачи, которые можно распознавать и применять
II продвинутый	Уметь сопоставлять и преподносить изученные материалы и утверждения, а также анализировать и синтезировать полученную информацию	Задачи, которые выходят за рамки традиционного обучения
III высокий	Знать и определять проблематику информации; уметь анализировать и интерпретировать полученные результаты математики, а также вести обсуждения на любую тему	Задачи, для умения выбирать математический инструментарий и осуществлять разработку алгоритма действий

2. Определение влияния показателей успеваемости на уровень компетенций

Оценка является исходной позицией для определения уровня сформированности набора компетенций, отвечающих за формирование знаний X_3, со всей совокупностью промежуточных результатов вычисления с целью создания новых методов в предметной области. Отметим, что состояние уровня сформированности компетенций $\dot{X}$ описывается «рабочей точкой», которое представляет собой вычислительное значение оценки обучающихся [4].

$$\dot{X} = f(\dot{X}_1, \dot{X}_2, \dot{X}_n). \quad (1)$$

Таким образом, согласно решению математического уравнения можно определить уровень сформированности набора компетенций между i-ми компетенциями в j-й группе с учетом использования обратной функции, как указано в формуле 2:

$$X_j + \Delta X_j \to \varphi\pi - 1 \to E + E_j. \quad (2)$$

Изменение X^* в «рабочей точке» следует описать уравнением ниже:

$$\Delta X = \sum_{i=1}^{n} K_i \Delta x_i \quad (3)$$

где значения линеаризации $\frac{\partial X}{\partial X_i} X^*, i \in \overline{1, I}$ определяются методом в областях ε_t^i на основе оценки к вариациям отдельных критериев успеваемости.

На основе оценивания уровня сформированности компетенций в матричной организационной системе обучающихся было выявлено, что на уровень сформированности воздействуют два показателя: уровень сформированности компетенций и управленческая деятельность. Таким образом, процедуру определения коэффициентов линеаризации k_4, k_5 в области линеаризации $\varepsilon_4, \varepsilon_5$, где

$$k_4 = tga_4 = 0,57,\ \Delta X_5 \in [0;0,3],\ k_5 = tga_5 = 0,83,\ \Delta X_6 \in [0;0,3],$$

следует представить следующим образом [5]:

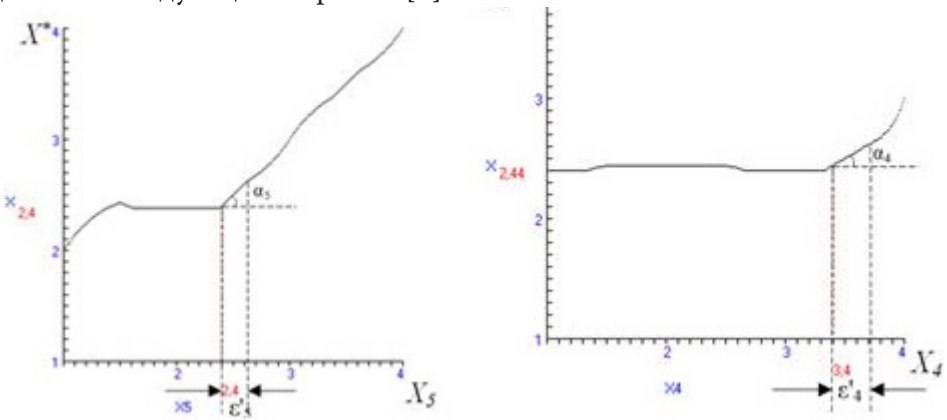


Рис. 1. Процедура определения коэффициентов линеаризации k_4, k_5

Таким образом, уравнение линеаризации выглядит следующим образом:

$$\Delta X = k_4 \Delta X_4 + k_5 \Delta X_5,\ \Delta X^{\max} = 0,57 * 0,3 + 0,83 * 0,3 \quad (4)$$

Таким образом, управленческие решения в матричной организационной системе оценки обучающихся должны охватывать процессы формирования компетенций и процессы профессиональной подготовки, учитывая при этом, принципы управления (измерения состояния величины, задания желаемого значения, получения обратной связи) с применением результатов измерений, основанных на уровне оценки обучающихся и уровне управленческих решений. Предложенные инструментальные средства реализуют компетентностный подход в области образовательных управленческих решений, соблюдая принципы управления, с целью выполнения задач в образовательном процессе.

Таким образом, алгоритм интеллектуальной поддержки принятия управленческих решений в матричной организационной системе непрерывной оценки обучающихся состоит из 3 этапов, где:

– на первом этапе формируется статистическая выборка обучающихся и их оценочная успеваемость;

– на втором этапе выполняется корреляционный анализ показателей;

– на третьем этапе строится когнитивная модель (рис. 2), где определяются концепты и аспекты процесса обучения, которые выявлены в когнитивной модели как нуждающиеся в корректировке или оптимизации и, на основе которой выполняется поиск соответствующих управленческих решений.

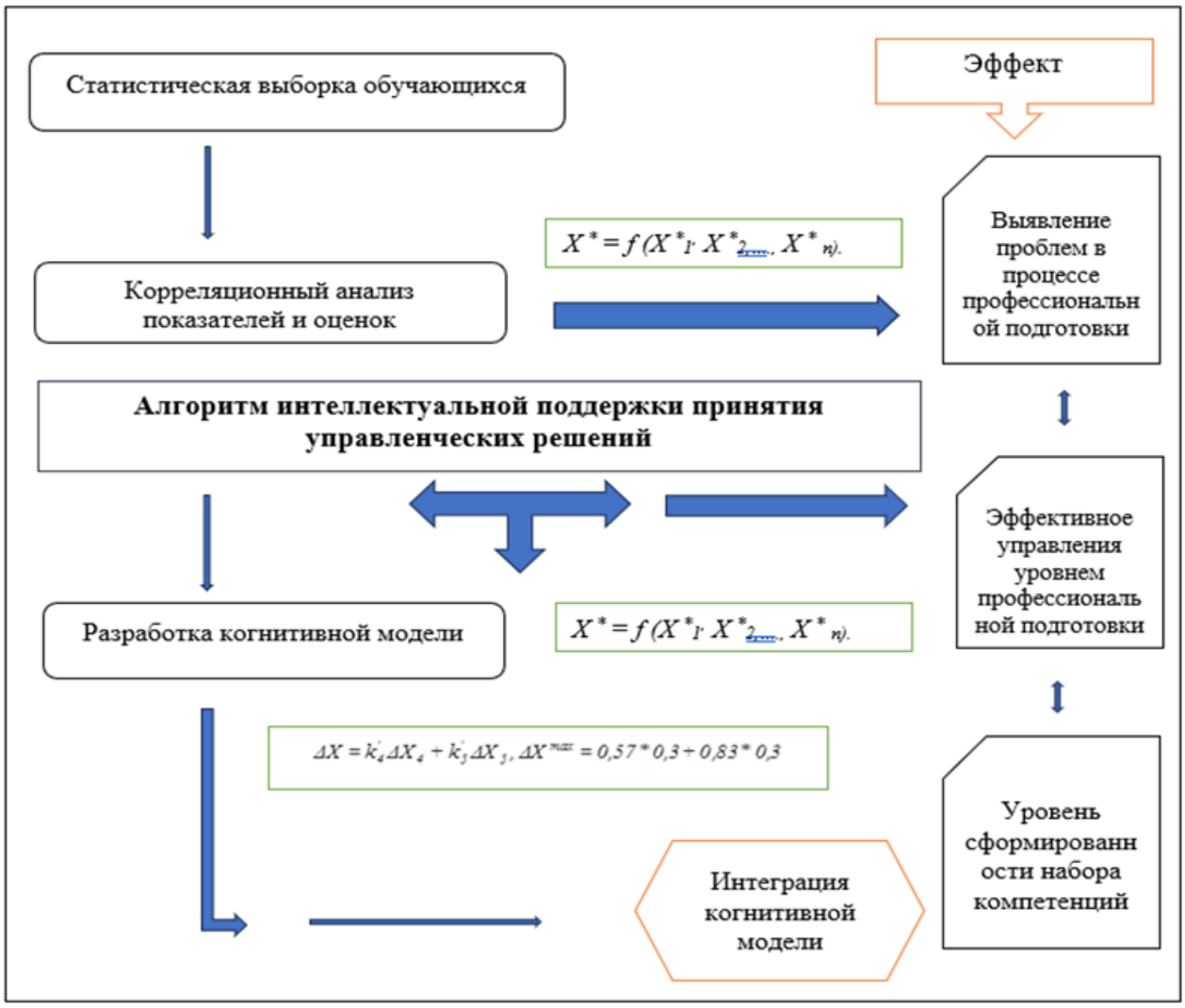


Рис. 2. Алгоритм интеллектуальной поддержки принятия управленческих решений

Таким образом, особенности предлагаемого алгоритма заключаются в том, что при построении когнитивной модели применяются как статистические, так и интеллектуальные методы, которые позволяют рассчитать ресурсы, обеспечивающие достижение максимального интегрального значения.

Заключение

Таким образом, данный алгоритм, возможно, применять для повышения эффективности процесса совершенствования образовательного процесса, а также при внедрении интерактивного метода обучения и т. д. Разработанный алгоритм интеллектуальной поддержки принятия управленческих решений в матричной организационной системе непрерывной оценки обучающихся позволяет выявить проблемы образовательного процесса с целью осуществления адресного воздействия при помощи методов двойного управления и математических решений в рамках эффективности удельного веса по освоению дисциплины.

Литература

1. *Анисова Т. Л.* Математические компетенции, уровни оценки / Т. Л. Анисов // Международный журнал экспериментального образования. – 2015. – № 11-4. – С. 497.

2. *Алексеев А. О.* Алгоритмические основы комплексного оценивания объектов / А. О. Алексеев // Фундаментальные исследования. – 2014. – № 3. – С. 474.

3. *Бурков В. Н.* Введение в теорию управления организационными системами: учебник / под ред. Д. А. Новикова. – М. : Либроком, 2009. – 264 с.

4. *Харитонов В. А.* Функциональные возможности комплексного оценивания с топологической интерпретацией матриц / В. А. Харитонов // Управление большими системами. – М. : Изд-во ИПУ РАН, 2007. – Вып. 18.

5. *Харитонов В. А.* Интеллектуальные технологии обоснования инновационных решений: монография / под науч. ред. В. А. Харитонова. – Пермь : Изд-во Перм. гос. техн. ун-та, 2010. – 363 с.

УДК 004.657

АНАЛИЗ РАСЧЁТА ПРОСРОЧЕННОЙ ЗАДОЛЖЕННОСТИ В КОРПОРАТИВНОМ ХРАНИЛИЩЕ ДАННЫХ БАНКА

П. Е. Толстых

Воронежский государственный университет

Аннотация. В статье проведён обзор и анализ двух алгоритмов расчёта просроченной задолженности в корпоративном хранилище данных банка: по методу *FIFO* (в первую очередь происходит погашение самой ранней просроченной задолженности) и по методу *LIFO* (в первую очередь происходит погашение последней просроченной задолженности).
Ключевые слова: корпоративное хранилище данных, просроченная задолженность, *FIFO*, *LIFO*, сторнирование.

Введение

Банковская сфера характеризуется большим объёмом данных, которые необходимы для создания отчётов [1]. Для быстрого анализа информации, формирования отчётности и корректности рабочих процессов возникает необходимость единого корпоративного хранилища данных (далее — КХД).

В КХД не только хранится информация о клиентах, заключённых договорах, операциях, транзакциях и других важных сущностях банковского процесса, но и происходит расчёт многих показателей банка, анализ которых помогает бизнес-пользователям принимать взвешенные решения. Одним из таких показателей является просроченная задолженность.

Просроченная задолженность — не погашенная в срок задолженность по основному долгу и/или плановым процентам за пользование ссудой, а также иным платежам по кредитному договору [2].

В КХД банка используются следующие основные алгоритмы расчёта просроченной задолженности: *FIFO* (*First In First Out* — «первым вошёл — первым вышел») и *LIFO* (*Last In First Out* — «последним вошёл — первым вышел»). Результаты расчётов доступны бизнес-пользователям и хранятся в объектах витринного слоя.

1. Требования к витрине, содержащей данные о просроченной задолженности

1.1. Функциональное описание

Витрина должна содержать исторические данные по просроченной задолженности по договору. Объект должен рассчитываться ежедневно.

В витрине учитывается просроченная задолженность по следующим типам:

- основной долг («*PRN*»);
- проценты («*INT*»);
- страховая плата («*INS*»);
- комиссии («*COM*»).

Информация должна быть сгруппирована в разрезе следующих измерений:

1) идентификатор договора;
2) тип просроченной задолженности;
3) дата начала просроченной задолженности.

История возникновения и погашения просроченной задолженности формируется по операциям, связанным с переносом договора на пророченную задолженность, и платежам клиента.

В случае, если по операции был выполнен процесс сторнирования [3] — отмена или корректировка записи об операции при ошибке — сторно-операция также должна быть учтена.

1.2. Атрибутный состав

Для предоставления отчётности обязательно наличие следующих атрибутов:

- идентификатор договора;
- тип просроченной задолженности;
- дата начала действия просроченной задолженности — дата окончания платёжного периода, в течение которого на счёт клиента не поступила минимально необходимая сумма, совпадает с датой операции, означающей перевод договора в состояние просроченной задолженности;
- дата окончания действия просроченной задолженности — дата совершения операции, суммы которой достаточно для погашения просроченного платежа;
- количество дней просроченной задолженности;
- уровень просроченной задолженности.

2. Алгоритм расчёта просроченной задолженности

Основные этапы расчёта просроченной задолженности:

1. Выделение из общего перечня операций, совершённых по договору, тех, которые непосредственно влияют на расчёт просроченной задолженности.

2. Определение «верно» сторнированных операций — система-источник в явном виде передала сторнирующую операцию с тем же типом и суммой.

3. Определение «плохо» сторнированных операций — у операции присутствует признак сторнирования, однако система-источник по каким-либо причинам не передала в КХД в явном виде сторнирующую операцию. В этом случае производится поиск сторнированных операций методом ближайшей операции: сторнирующей считается операция, ближайшая по дате к сторнированной операции и совпадающая по сумме операции.

4. Определение дат начала и окончания просроченной задолженности: даты операций с типом «*OVD*» (просроченная задолженность) становятся датами начала просроченной задолженности, даты операций с типом «*PAY*» (платёж) — датами окончания просроченной задолженности в случае, если суммы платежа достаточно для погашения текущей задолженности.

5. Пересчёт количества дней и уровня просроченной задолженности в зависимости от операций по договору.

Алгоритм выполнения п. 4, 5 зависит от метода, по которому происходит расчёт просроченной задолженности.

2.1. FIFO

Алгоритм *FIFO* — метод учёта просроченной задолженности, при котором внесённый платеж погашает первый просроченный платеж.

Дата начала непрерывной просроченной задолженности может меняться по мере того, как происходит её погашение.

Пример расчёта просроченной задолженности по методу *FIFO*:

1. Дата окончания платёжного периода по договору — 10.09.2024. На счёт клиента не поступила сумма, необходимая для выполнения обязательства по платежу. В системе-источнике появилась информация о начале просроченной задолженности по основному долгу. После расчёта в КХД строка с данной информацией появилась в витрине (табл. 1) с заполненной датой начала просроченной задолженности, но с незаполненной датой окончания.

Таблица 1

Состояние витрины после первого выхода договора на просроченную задолженность (FIFO)

Тип	Дата начала	Дата окончания	Уровень	Кол-во дней
PRN	10.09.2024	NULL	1	1

2. Дата окончания следующего платёжного периода по договору — 10.10.2024. За период с 10.09.2024 по 10.10.2024 на счёт клиента также не поступило ни одного платежа, который мог бы погасить просроченную задолженность. В системе-источнике сформировалась ещё одна операция по переносу договора на просроченную задолженность, которая появилась в витрине с уровнем 2 (табл. 2). Помимо этого, клиент не оплатил комиссию, что также отразилось в системе-источнике и витрине. При этом у предыдущей просроченной задолженности изменилось количество дней (меняется каждый день при расчете). Датой начала непрерывной просроченной задолженности считается 10.09.2024.

Таблица 2

Состояние витрины после повторного выхода договора на просроченную задолженность (FIFO)

Тип	Дата начала	Дата окончания	Уровень	Кол-во дней
PRN	10.09.2024	NULL	1	30
PRN	10.10.2024	NULL	2	1
COM	10.10.2024	NULL	1	1

3. На следующий день клиент внёс сумму по основному долгу, достаточную для погашения одной из просроченных задолженностей. В системе-источнике появилась запись с типом платёж и была передана в КХД. У самой ранней просроченной задолженности (от 10.09.2024) появилась дата окончания. Теперь датой начала просроченной задолженности по договору считается 10.10.2024 (табл. 3).

Таблица 3

Состояние витрины с информацией о просроченной задолженности после погашения (FIFO)

Тип	Дата начала	Дата окончания	Уровень	Кол-во дней
PRN	10.09.2024	11.10.2024	1	31
PRN	10.10.2024	NULL	2	2
COM	10.10.2024	NULL	1	2

Для реализации данного алгоритма в КХД используются аналитические (оконные) функции. Они позволяют вычислять различные показатели на основе группы строк, возвращая при этом результат из нескольких записей для каждой группы [4].

Использованы следующие функции:

• *SUM* — возвращает арифметическую сумму всех выбранных значений поля [5], используется для вычисления сумм просроченных задолженностей и платежей;

• *LEAD* и *LAG* — функции смещения, возвращающие значение выражения, вычисленного для следующей или предыдущей строки результирующего набора соответственно [6], используются для определения дат начала и окончания просроченной задолженности;

• *ROW_NUMBER* — возвращает номер строки в текущей выборке [7], используется для определения уровня просроченной задолженности.

2.2. LIFO

Алгоритм *LIFO* — метод учёта просроченной задолженности, при котором внесённый платеж погашает последний просроченный платеж.

Дата начала непрерывной просроченной задолженности не может меняться по мере того, как происходит её погашение.

Пример расчёта просроченной задолженности по методу *LIFO*:

1. Дата окончания платёжного периода по договору — 10.09.2024. На счёт клиента не поступила сумма, необходимая для выполнения обязательства по платежу. В системе-источнике появилась информация о начале просроченной задолженности по основному долгу. После расчёта в КХД эта строка появилась в витрине с заполненной датой начала просроченной задолженности, но с незаполненной датой окончания. Также в витрине появилась запись с типом «*OVD*», означающая непрерывную просроченную задолженность (табл. 4).

Таблица 4

Состояние витрины после первого выхода договора на просроченную задолженность (LIFO)

Тип	Дата начала	Дата окончания	Уровень	Кол-во дней
OVD	10.09.2024	NULL	1	1
PRN	10.09.2024	NULL	1	1

2. Дата окончания следующего платёжного периода по договору — 10.10.2024. За период с 10.09.2024 по 10.10.2024 на счёт клиента также не поступило ни одного платежа, который мог бы погасить просроченную задолженность. В системе-источнике сформировалась ещё одна операция по переносу договора на просроченную задолженность, которая отразилась в витрине после расчёта с уровнем 2 (табл. 5). При этом уровень у непрерывной просроченной задолженности не изменился, он всегда остаётся равным единице. Помимо этого, клиент не оплатил комиссию, что также отразилось сначала в системе-источнике, а затем и в витрине. При этом у предыдущей просроченной задолженности изменилось количество дней (меняется каждый день при расчете). Датой начала непрерывной просроченной задолженности считается 10.09.2024.

Таблица 5

Состояние витрины после повторного выхода договора на просроченную задолженность (LIFO)

Тип	Дата начала	Дата окончания	Уровень	Кол-во дней
OVD	10.09.2024	NULL	1	30
PRN	10.09.2024	NULL	1	30
PRN	10.10.2024	NULL	2	1
COM	10.10.2024	NULL	1	1

3. На следующий день клиент внёс сумму по основному долгу, достаточную для погашения одной из просроченных задолженностей. В системе-источнике появилась запись с типом платёж и была передана в КХД. У самой поздней просроченной задолженности (от 10.09.2024) появилась дата окончания. При этом дата начала непрерывной просроченной задолженности по договору не изменилась (табл. 6).

4. Через несколько дней клиент внёс два платежа, суммы которых достаточно для погашения двух просроченных задолженностей: по основному долгу и по комиссиям. После расчёта витрины в ней появились даты окончания просроченных задолженностей. Так как больше у данного договора нет открытых просроченных задолженностей, дата окончания появилась также и у строки с непрерывной просроченной задолженностью (табл. 7).

Таблица 6

Состояние витрины с информацией о просроченной задолженности после первого погашения (LIFO)

Тип	Дата начала	Дата окончания	Уровень	Кол-во дней
OVD	10.09.2024	NULL	1	31
PRN	10.09.2024	NULL	1	31
PRN	10.10.2024	11.10.2024	2	2
COM	10.10.2024	NULL	1	2

Таблица 7

Состояние витрины с информацией о просроченной задолженности после полного погашения (LIFO)

Тип	Дата начала	Дата окончания	Уровень	Кол-во дней
OVD	10.09.2024	15.10.2024	1	34
PRN	10.09.2024	15.10.2024	1	34
PRN	10.10.2024	11.10.2024	2	2
COM	10.10.2024	15.10.2024	1	6

Для реализации данного алгоритма в КХД используется PL/SQL. Это язык программирования, который предоставляет процедурные расширения SQL и позволяет реализовывать сложную бизнес-логику с использованием условных операторов, циклических конструкций, процедур, функций и других элементов кода [8].

Для корректного расчета просроченной задолженности используются несколько переменных с пользовательским комбинированным типом данных — запись [9].

Запись содержит следующие поля:

- номер просроченной задолженности;
- номер договора;
- тип просроченной задолженности;
- дата окончания просроченной задолженности;
- сумма.

В цикле с курсором (набором строк, полученным при помощи оператора SELECT и связанным с ним указателем текущей строки [9]) по всем операциям происходит формирование массива сумм.

Для каждой операции с типом «OVD» (просроченная задолженность) происходит добавление нового значения в массив сумм. Для каждой последующей строки для каждого значения массива рассчитывается сумма (для операций с типом «OVD» учитывается отрицательное значение суммы, для операций с типом «PAY» — положительное).

Этот процесс продолжается, пока текущая сумма не станет неотрицательной (т. е. просроченная задолженность закроется). После этого происходит запись даты закрытия просроченной задолженности.

Как только заканчивается обработка всех операций по одному типу просроченной задолженности или договору, полученные данные из записи переносятся в целевую таблицу. Расчёт продолжается для всех оставшихся типов просроченных задолженностей и договоров.

Заключение

Для быстрой и корректной работы банковских процессов часто на стороне КХД происходит расчёт многих показателей, одним из которых является просроченная задолженность. Это сложный вычисляемый показатель, который непосредственно влияет на деятельность банка, позволяет оценить кредитную историю клиента, его платежеспособность и принять решение по выдаче кредита.

Расчёт просроченной задолженности может происходить по двум алгоритмам: FIFO (сначала погашаются более ранние просроченные задолженности) или LIFO (сначала погашаются более поздние просроченные задолженности).

Эти методы являются алгоритмами высокой сложности, т. к. требуют корректного соотношения между операциями перевода на просроченную задолженность и платежами, а также построения истории, поэтому для их реализации в КХД используются расширенные возможности SQL — аналитические функции и процедурное расширение PL/SQL.

Литература

1. Хранилище данных для банков. – URL: https://datafinder.ru/solutions/hranilishche-dannyh-dlya-bankov?

2. Просроченная задолженность. – URL: https://www.banki.ru/wikibank/prosrochennaya_zadoljennost/

3. Что такое сторно? Проводки в бухгалтерии. – URL: https://assistentus.ru/buhuchet/storno/

4. Analytic Functions. – https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/Analytic-Functions.html

5. СУБД: язык SQL в примерах и задачах / И. Ф. Астахова [и др.]. – Москва : ФИЗМАТЛИТ, 2009. – 38 с.

6. Функции LAG и LEAD. – http://www.sql-tutorial.ru/ru/book_lag_and_lead_functions.

7. Учебник Oracle SQL, PL/SQL | Аналитические функции: – URL: http://orasql.ru/sql/standfunc/analytics

8. *Фейерштейн С.* Oracle PL/SQL. Для профессионалов/ С. Фейерштейн, Б. Прибыл. – 6-е изд. – СПб. : Питер, 2015. – 26 с.

9. *Домников А. С.* Введение в Oracle / А. С. Домников, В. И. Комаров. – Москва : МГТУ имени Н. Э. Баумана, 2005. – 44 с.

УДК 004.056

АРХИТЕКТУРА СППР ДЛЯ ОЦЕНКИ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

М. А. Утенкова, А. А. Комков, Е. А. Максимова

МИРЭА – Российский технологический университет

Аннотация. В ходе работы было исследована история развития архитектуры СППР. Выделены особенности каждого типа: СППР, ориентированные на модели; СППР, ориентированные на данные; СППР, ориентированные на документы; СППР, ориентированные на знания; СППР, ориентированные на коммуникацию; СППР на основе сети Интернет. Проведена аналитическая работа по выявлению более подходящей архитектуры СППР для оценки рисков информационной безопасности. Приведена типовая архитектура СППР для оценки рисков информационной безопасности.

Ключевые слова: система поддержки принятия решений, информационная безопасность, оценка рисков, архитектура СППР, принятие решений, количественные метрики, Model-driven DSS, Data-driven DSS, Communication-driven DSS, Document-driven DSS, Knowledge-driven DSS, Web-based DSS.

Введение

Системы поддержки принятия решений являются важной частью процесса принятия решений в современных компаниях. СППР могут применяться в бизнес-процессах организаций различных отраслей, в том числе относящихся к субъектам критической информационной инфраструктуры (см. рис. 1) [1, 6]. За счет цифровизации процессов и развития обработки данных при помощи искусственного интеллекта, обостряется вопрос защиты используемой информации. Во всех представленных отраслях СППР могут быть использованы и для решения отдельных задач информационной безопасности (далее — ИБ), в частности — оценки рисков.

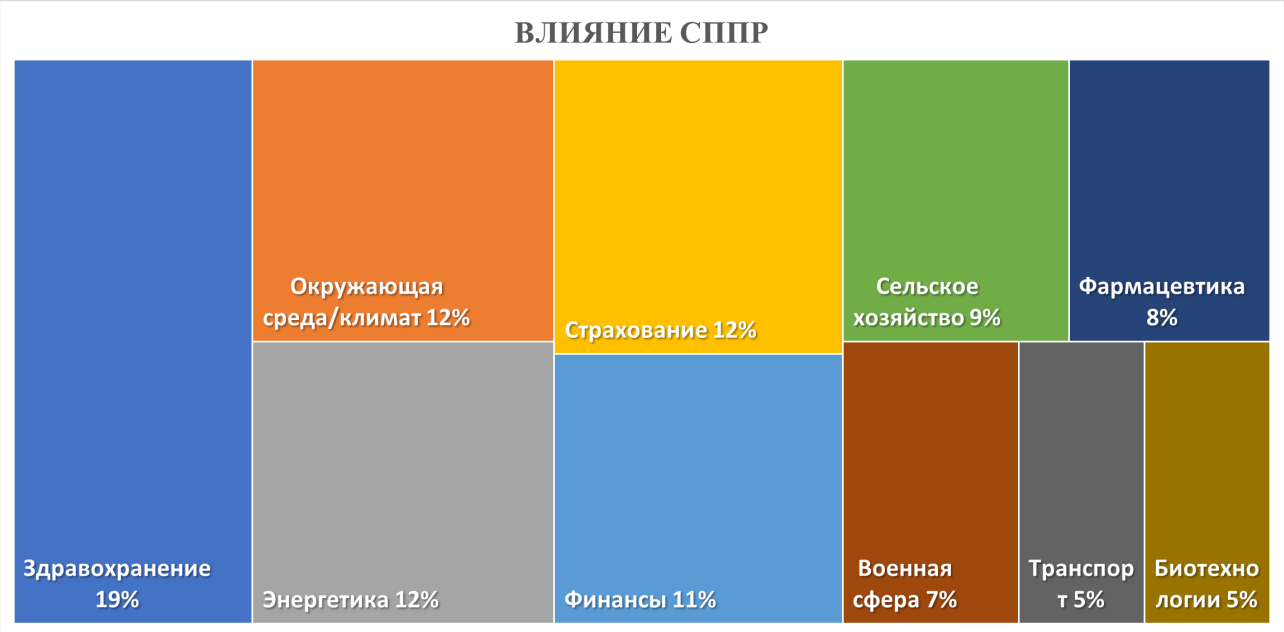


Рис. 1. Соотношение популярности использования СППР в различных отраслях

На сегодняшний день существует множество различных архитектур СППР и их реализаций. Каждая архитектура наилучшим образом подходит для решения определенных задач. В рамках данного исследования для решения проблемы выбора архитектуры СППР для оценки рисков ИБ необходимо проанализировать и сравнить различные архитектуры СППР и выбрать наилучшую, которая поможет поддерживать высокий уровень защищенности на протяжении всего процесса эксплуатирования системы.

1. Основные этапы развития СППР

Для принятия решения о выборе архитектуры систем поддержки принятия решений, следует сделать шаг назад и посмотреть на весь проделанный путь СППР. Хоть история СППР не столь велика, но уже сейчас можно выделить основные исторические периоды, в которых СППР заметно менялось (см. рис. 2) [7].

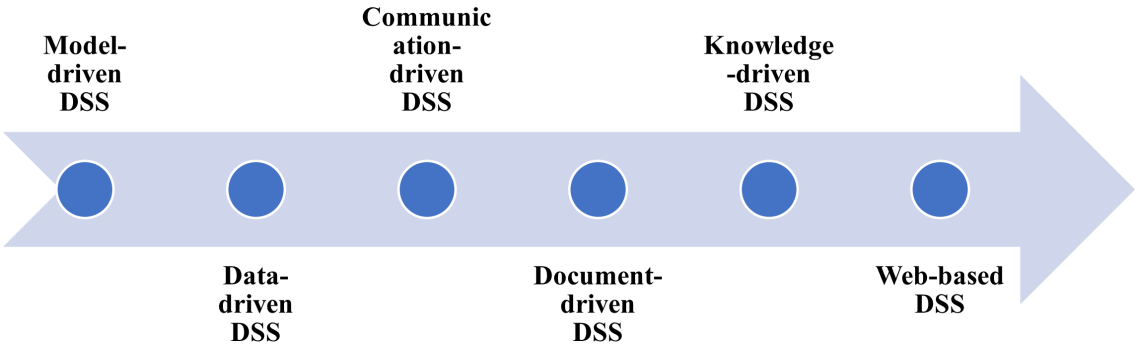


Рис. 2. Основные этапы развития СППР

Начиная историческую справку, стоит отметить СППР на основе моделей, которые были основаны на простых количественных моделях: статистические данные, финансовые данные или расписания, а также их взаимосвязях [2]. Используя простые инструменты анализа и сложную статистику, такие СППР служат для взаимодействия с пользователем на не столь больших объемах данных в сравнении с другими типами СППР.

Следующий тип, который следует отметить являются системы поддержки принятия решений, ориентированные на данные, которые делают акцент на запросы к хранилищам данных в целях нахождения конкретных ответов для конкретных целей, опираясь на данные временных рядов.

Отдельно стоит выделить системы поддержки принятия решений, ориентированные на коммуникацию, так как в отличие от разобранных выше типов, она поддерживает совместную работу и коммуникацию. Основной задачей данных СППР становится обеспечение удобной совместной работы пользователей. Самые распространенные технологии, использующиеся в этом типе СППР являются клиент-серверная технология и веб-технология: программы для обмена мгновенными сообщениями, онлайн системы совместной работы.

Для работы с большими объемами информации в неструктурированном виде используются системы поддержки принятия решений, ориентированные на документы. Одной из целей таких СППР может быть осуществление поиска в документах по определённому набору ключей или поисковых запросов. За информацию могут быть взяты факты и цифры, протоколы совещаний, каталоги, деловые переписки и любая другая документируемая информация в компании.

Для взаимодействия искусственного интеллекта и когнитивных способностей человека используются системы поддержки принятия решений, основанные на знаниях. Это обширная категория, охватывающая широкий спектр систем, предназначенных для использования в ор-

ганизации. Перед ними ставится цель предоставления рекомендаций по управлению или выбора продукции и услуг. Особое внимание уделяется обработки фактов и экспертных оценок.

Переходя к самой сложной системе по своей структуре — веб-СППР, которые работают на основе всемирной сети интернет. Благодаря осуществлению вычислений в облаке данные СППР предоставляют эффективные решения, которые могут привести к значительным сокращениям расходов. За счет доступности всемирной аудитории эти СППР не ограничены в масштабируемости.

2. Выявление наиболее подходящего вида СППР для оценки рисков ИБ

Из представленных видов СППР, появлявшихся на каждом этапе их развития, в меньшей степени для оценки рисков ИБ подходят СППР, ориентированные на документы. Оценка рисков опирается на количественные метрики, которые сложно извлечь из неструктурированных данных. Тем не менее в перспективе развитие технологий распознавания естественно-языковых текстов данная проблема может быть решена следующими поколениями нейросетей.

В классическом представлении риск ИБ вычисляется как сумма произведений стоимости ущерба при реализации некоторой угрозы на вероятность ее реализации. Эти данные могут быть предоставлены экспертами в качестве модели, в рамках которой СППР производит вычисление.

Однако рост масштаба информационных систем, которые требуется защищать, а также увеличение баз данных угроз и уязвимостей, которые необходимо учитывать, не всегда позволяет обработать вручную такие объемы данных. Поэтому на передний план выходят СППР, ориентированные на данные. В хранилище данных содержится информация обо всех актуальных угрозах и уязвимостях, с которой СППР в дальнейшем работает. Эти данные могут быть получены от SIEM-систем или других средств защиты информации, способных осуществлять сканирование информационных систем на наличие уязвимостей.

Также важно отметить, что в процессе оценки рисков ИБ должны участвовать специалисты как непосредственно по защите информации, так и финансового направления организации, так как только они могут знать стоимость ущерба в случае нарушения работы отдельных сегментов информационной инфраструктуры. Набирающие популярность СППР основанные на безлюдных подходах, не подходят [3]. Поэтому СППР должна поддерживать возможность совместной работы, то есть быть ориентированной на коммуникации.

В спектр задач СППР может входить также построение прогнозных моделей, которые могут быть использованы как бизнесе в целом [4], так и в информационной безопасности [8–9].

В перспективе развития СППР именно для решения задач по оценки рисков ИБ может быть добавлено применение технологий искусственного интеллекта при обработке данных, а также использование общедоступных баз данных, расположенных в сети Интернет (например, БДУ ФСТЭК России или база данных сертифицированных средств защиты информации с указанием закрываемых ими угроз и их стоимостью).

3. Технические аспекты архитектуры СППР

СППР могут быть классифицированы по функционалу [5]. Пассивные СППР не дают рекомендаций для пользователя, а только аккумулируют и представляют в удобном для анализа виде входные данные. В отличие от них существуют также активные СППР, которые для лица принимающего решения предлагают рекомендованные варианты решения задачи. Существуют также комплексные (они же совместные) СППР, которые объединяют оба способа работы СППР. В рамках оценки рисков ИБ лучше всего подойдет либо пассивная, либо комплексная СППР.

Уже была обоснована необходимость многопользовательского режима работы СППР. Для его реализации может быть использована клиент-серверная архитектура СППР. В качестве клиентов выступает АРМ каждого пользователя, а сервер является отдельным ЭВМ, на котором происходят все аналитические вычисления. При этом не стоит забывать о необходимости наличия хранилища данных, представляющего собой базу данных, в которой средства защиты информации аккумулируют информацию об угрозах и уязвимостях, а специалисты финансового направления организации — данные о возможном ущербе в случае их реализации. На рис. 3 представлена типовая архитектура СППР, подходящая под задачи оценки рисков ИБ.

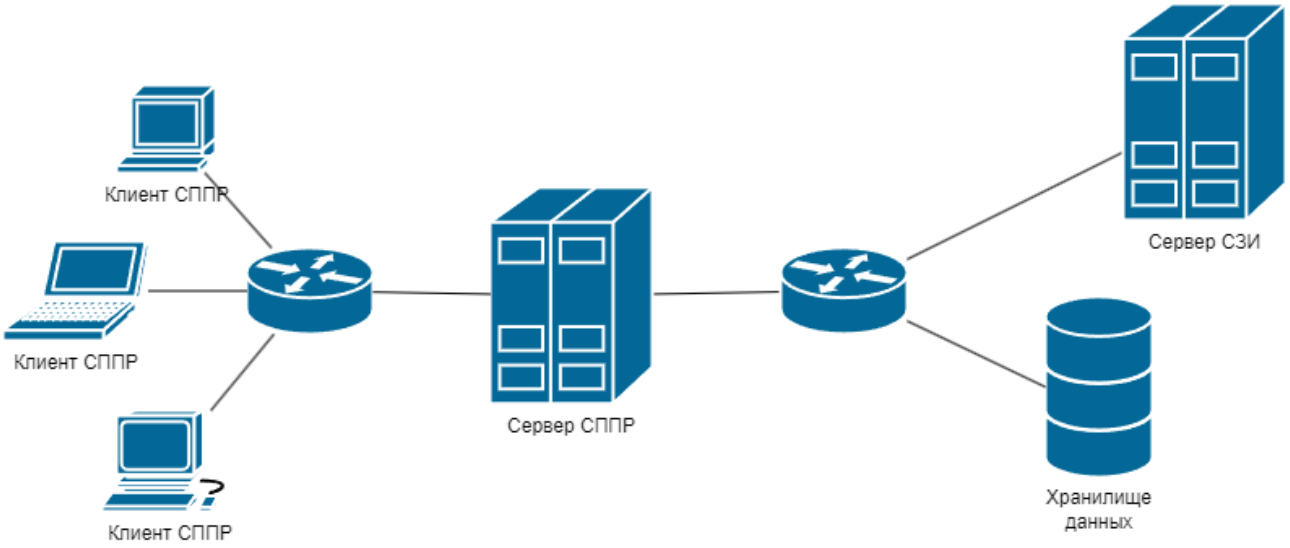


Рис. 3. Типовая архитектура СППР для оценки рисков ИБ

Заключение

В ходе исследования были рассмотрены основные этапы развития видов СППР и их архитектуры. Исходя из особенностей каждого типа СППР, были выведены преимущества и недостатки относительно применимости для оценки рисков ИБ. Наиболее перспективным вектором развития СППР для решения поставленной задачи выделяется WEB-DSS, который также может включать в себя улучшенные алгоритмы анализа данных и технологии искусственного интеллекта. Непосредственный доступ к базам данных сети интернет позволит своевременно актуализировать информацию об угрозах и уязвимостях, что в свою очередь выведет оценку рисков ИБ на новый качественный уровень.

Литература

1. Федеральный закон № 187-ФЗ от 26.07.2017 «О безопасности критической информационной инфраструктуры Российской Федерации».

2. *Антонов В. В., Конев К. А.* Усовершенствование ситуационной методологии разработки систем поддержки принятия решений для предприятий // Онтология проектирования. – 2022. – №4 (46). – URL: https://cyberleninka.ru/article/n/usovershenstvovanie-situatsionnoy-metodologii-razrabotki-sistem-podderzhki-prinyatiya-resheniy-dlya-predpriyatiy (дата обращения: 07.11.2024).

3. *Афанасьев М. А.* Рентабельность функционирования СППР с подсистемой принятия решений безлюдными технологиями // Научные труды Вольного экономического общества России. – 2023. – № 3. – URL: https://cyberleninka.ru/article/n/rentabelnost-funktsionirovaniya-sppr-s-podsistemoy-prinyatiya-resheniy-bezlyudnymi-tehnologiyami (дата обращения: 07.11.2024).

4. *Янушкина Ю. И., Комков А. А.* Анализ зарубежного опыта обеспечения устойчивости национальной валюты // Сборник научных трудов «Цифровизация науки и образования: перспективы создания единой системы знаний». Материалы Международных научно-практических мероприятий Общества Науки и Творчества (г. Казань) за октябрь 2024 года.

5. *Овчинников В. В., Станкевич С. А., Никольский С. Н.* Архитектура и таксономия систем поддержки принятия решений // Прикаспийский журнал: управление и высокие технологии. – 2018. – № 3 (43). – URL: https://cyberleninka.ru/article/n/arhitektura-i-taksonomiya-sistem-podderzhki-prinyatiya-resheniy (дата обращения: 07.11.2024).

6. Clinical Decision Support Systems. PSNet [internet]. Rockville (MD): Agency for Healthcare Research and Quality, US Department of Health and Human Services. – 2019. – URL: https://psnet.ahrq.gov/primer/clinical-decision-support-systems

7. *Power D. J.* A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web, http://DSSResources.COM/history/dsshistory.html, version 4.0, March 10, 2007.

8. *Utenkova M. A., Maksimova E. A.* Proactive modeling of limited access data leaks at critical information infrastructure facilities (using the example of the transport industry) // Scientific and analytical journal «Vestnik Saint-Petersburg university of State fire service of EMERCOM of Russia». – 2024. – № 2. – P. 91–104. DOI: https://doi.org/10.61260/2218-130X-2024-2-91-104

9. *Utenkova M., Maksimova E.* Forecasting the Development of Information Security Events in the Context of Information Warfare. In: Lapina M., Raza Z., Tchernykh A., Sajid M., Zolotarev V., Babenko M. (eds) AISMA-2024: International Workshop on Advanced Information Security Management and Applications. AISMA 2024. Lecture Notes in Networks and Systems. – 2024. – Vol. 863. Springer, Cham. https://doi.org/10.1007/978-3-031-72171-7_32

УДК 004:658

СОСТАВЛЯЮЩИЕ ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКОЙ КОМПЕТЕННОСТИ СОТРУДНИКОВ ОВД РОССИИ

П. В. Шаляпин[1], Л. В. Россихина[2]

[1]Оперативно-поисковое бюро МВД России
[2]Академия Управления МВД России

Аннотация. В статье исследуются основные составляющие информационно-технологической компетенции сотрудников органов внутренних дел (ОВД) России. Рассмотрены ключевые элементы этой компетенции, включая знание основ информатики, владение специализированными программами, навыки работы с большими данными и аналитическими инструментами, а также готовность к непрерывному обучению. Особое внимание уделено применению математических моделей и методов анализа данных в деятельности ОВД. Предложены рекомендации по совершенствованию информационно-технологической подготовки сотрудников ОВД.
Ключевые слова: информационно-технологическая компетенция, органы внутренних дел, математика в ОВД, прогнозирование преступлений, анализ больших данных, кибербезопасность, профессиональное развитие.

Введение

Современный мир характеризуется стремительным развитием информационных технологий, которые оказывают значительное влияние на все сферы жизни общества, включая обеспечение общественной безопасности и правопорядка [1]. Органы внутренних дел (ОВД) Российской Федерации играют важнейшую роль в поддержании стабильности и защите прав граждан, и для эффективного выполнения своих функций сотрудники этих структур должны обладать высоким уровнем информационно-технологической компетенции. В данном исследовании мы рассмотрим ключевые составляющие этой компетенции, проанализируем существующие подходы к ее формированию и развитию, а также предложим пути совершенствования процесса обучения и профессиональной подготовки сотрудников ОВД.

1. Основные компоненты информационно-технологической компетенции

Информационно-технологическая компетенция сотрудников ОВД включает в себя следующие основные компоненты:

Знание основ информатики и вычислительных систем: включает базовые понятия о принципах работы компьютерных систем, архитектуры компьютеров, операционных систем, сетевых технологий и баз данных. Этот компонент необходим для понимания того, как функционируют современные информационные системы, используемые в правоохранительных органах.

Владение специализированными программами и приложениями: сотрудники ОВД должны уметь эффективно использовать программное обеспечение, предназначенное для решения задач, связанных с управлением информацией, расследованием преступлений, мониторингом общественного порядка и другими аспектами их профессиональной деятельности. К таким программам относятся системы автоматизированного документооборота, базы данных криминальной статистики, аналитические инструменты и средства защиты информации.

Навыки работы с большими данными и аналитическими инструментами: важнейшим элементом современной информационно-технологической компетентности является умение

обрабатывать и анализировать большие массивы данных, получаемых из различных источников. Сюда входят навыки работы с системами бизнес-аналитики (BI), средствами визуализации данных, а также знание основ машинного обучения и искусственного интеллекта.

Понимание вопросов кибербезопасности и защиты информации: учитывая рост числа киберугроз, сотрудникам ОВД необходимо владеть основами информационной безопасности, знать принципы защиты конфиденциальной информации, а также уметь выявлять и предотвращать попытки несанкционированного доступа к данным.

Готовность к непрерывному обучению и профессиональному росту: информационные технологии постоянно эволюционируют, и сотрудники ОВД обязаны поддерживать свою квалификацию на высоком уровне, осваивая новые инструменты и методики работы с информацией.

2. Математическое моделирование и анализ данных в ОВД

Одним из важнейших аспектов информационно-технологической компетенции является умение применять математические модели и методы анализа данных для решения практических задач [2]. Рассмотрим некоторые примеры таких подходов.

2.1. Прогнозирование преступной активности

Прогностический анализ преступной активности позволяет сотрудникам ОВД планировать свои ресурсы и меры профилактики наиболее эффективным образом. Одним из популярных методов прогнозирования является метод временных рядов [5], который основывается на анализе исторических данных о преступлениях. Рассмотрим простую модель авторегрессионного процесса первого порядка (AR(1)):

$$y_t = \phi y_{t-1} + \epsilon_t,$$

где y_t — число преступлений в период t, ϕ — коэффициент автокорреляции, ϵ_t — случайная ошибка.

Эта модель позволяет предсказывать будущее значение ряда на основе предыдущих значений и помогает выявить тенденции и закономерности в динамике преступной активности [6].

2.2. Анализ социальных сетей и связей между подозреваемыми

Анализ социальных сетей используется для выявления взаимосвязей между участниками преступных группировок и потенциальными сообщниками. Методология социального графового анализа позволяет строить модели взаимодействия между людьми и выявлять ключевые фигуры в сети [7]. Например, центральность узла (degree centrality) определяется формулой:

$$C_D(v) = \frac{\deg(v)}{N-1},$$

где $\deg(v)$ — степень вершины v, N — общее количество вершин в графе.

Этот показатель позволяет определить, насколько важна та или иная фигура в социальной сети, и может использоваться для планирования оперативных мероприятий.

2.3. Моделирование оптимальных маршрутов патрулирования

Оптимальное распределение ресурсов патрульной службы является важной задачей для обеспечения безопасности на улицах городов. Для решения этой задачи могут применяться методы линейного программирования [5]. Рассмотрим классическую транспортную задачу:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij}.$$

При ограничениях:

$$\sum_{i=1}^{m} x_{ij} = \mathrm{d}_j, j = 1,\ldots,n,$$

$$\sum_{j=1}^{n} x_{ij} = s_i, i = 1,\ldots,m,$$

$$x_{ij} \geq 0, \ \forall \mathrm{i}, j,$$

где c_{ij} — стоимость перемещения из пункта i в пункт j, d_j — спрос в пункте j, s_i — предложение в пункте i, x_{ij} — объем перевозки из i-го пункта в j-й.

Эта задача позволяет найти оптимальный маршрут патрулирования, минимизируя затраты на передвижение и обеспечивая максимальное покрытие территории.

3. Совершенствование информационно-технологической компетенции

Для повышения уровня информационно-технологической компетентности сотрудников ОВД необходимо разрабатывать и внедрять комплексные образовательные программы, включающие теоретические и практические занятия. Важно также учитывать индивидуальные особенности обучающихся, предлагая персонализированные траектории развития [3,4]. Ниже приведены рекомендации по совершенствованию процесса обучения:

Интерактивные учебные курсы: использование интерактивных платформ и симуляторов позволит сотрудникам лучше усваивать материал и практиковаться в решении реальных задач.

Обучение на рабочем месте: организация стажировок и тренингов непосредственно на рабочих местах поможет сотрудникам быстрее адаптироваться к новым технологиям и инструментам.

Регулярные обновления учебных материалов: постоянное обновление учебных программ и материалов в соответствии с последними достижениями в области информационных технологий обеспечит актуальность знаний и навыков сотрудников.

Взаимодействие с научным сообществом: сотрудничество с университетами и исследовательскими центрами позволит привлекать экспертов для проведения лекций и семинаров, а также участвовать в совместных научных проектах.

Заключение

Информационно-технологическая компетенция сотрудников ОВД является ключевым фактором повышения эффективности их работы и обеспечения безопасности граждан. Развитие этой компетенции должно включать систематическое обучение, внедрение передовых технологий и постоянное повышение квалификации персонала. Комплексный подход к образованию и подготовке кадров, основанный на использовании современных образовательных технологий и математических методов анализа данных, позволит значительно улучшить качество работы правоохранительных органов и повысить уровень общественной безопасности.

Литература

1. *Аверьянова Т. В.* Информационная безопасность в органах внутренних дел : учебное пособие / Т. В. Аверьянова. – Москва : Юнити-Дана, 2017. – 223 с.

2. *Данько Т. П.* Современные информационные технологии в управлении : учеб. для вузов / Т. П. Данько. – Минск : Амалфея, 2020. – 416 с.

3. *Козлов В. Н.* Информационное право : учеб. пос. для студентов юрид. фак. / В. Н. Козлов. – Ростов-на-Дону : Феникс, 2021. – 352 с.

4. *Лебедев С. А.* Философия науки : учеб. для магистрантов и аспирантов / С. А. Лебедев. – М. : Академический проект, 2020. – 608 с.

5. *Молчанов А. Ю.* Основы теории информации : учеб.пособие / А. Ю. Молчанов. – Санкт-Петербург : Питер, 2019. – 480 с.

6. *Пароджанов В. Д.* Как улучшить работу ума : алгоритмы без программистов – просто и доступно / В. Д. Пароджанов. – Дубна : Феникс+, 2018. – 224 с.

7. *Сергиенко А. Б.* Цифровая обработка сигналов : учеб. пособия для вузов / А. Б. Сергиенко. – СПБ. : Питер, 2021. – 604 с.

8. *Трайнев В. А.* Информационные коммуникационные педагогические технологии (обобщение и рекомендации) : учеб.-метод. пособие / В. А. Трайнев. – 2-е изд., испр. и доп. – Москва : Дашков и Ко, 2020. – 272 с.

УДК 004.42

РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ УПРАВЛЕНИЯ ИГРОВЫМИ СОБЫТИЯМИ

К. Н. Шестаков, В. О. Кушев

Пермский государственный национальный исследовательский университет

Аннотация. В работе представлено моделирование информационной системы, предназначенной для управления игровыми событиями. В рамках исследования разработаны диаграммы прецедентов, описывающие взаимодействие трёх типов акторов. Для каждого актора выделены основные функциональные возможности, такие как регистрация, авторизация, создание и поиск игровых событий. Подробно описаны ключевые процессы, включая персонализацию поиска событий и обработку жалоб, с использованием диаграмм активности в нотации BPMN. Кроме того, построена ER-диаграмма, которая демонстрирует основные сущности системы. Представленные модели обеспечивают структурированный подход к моделированию системы, способствуя её дальнейшей реализации и улучшению.

Ключевые слова: информационная система, диаграммы прецедентов, диаграммы активности, UML, BPMN, ER-диаграмма, акторы, игровые события, модерация, управление пользователями, проектирование базы данных.

Введение

Современные информационные системы играют ключевую роль в организации и управлении различными видами активности пользователей, особенно в области цифровых платформ, где активно взаимодействуют различные категории участников. Одной из таких областей являются игровые платформы, которые предоставляют пользователям возможность создавать, находить и участвовать в игровых событиях. Однако разработка подобных систем требует тщательного проектирования архитектуры, чтобы учесть разнообразие функциональных требований, типов пользователей и их взаимодействий.

Целью работы является разработка концептуальной модели системы, обеспечивающей интуитивное и продуктивное взаимодействие пользователей с платформой для управления игровыми событиями.

Проектирование

Начать моделирование системы стоит с создания диаграмм прецедентов. Таким образом будут обозначены основные варианты использования системы пользователями. Диаграммы разработаны используя нотацию UML. Правила и подходы моделирования информационной системы применялись в соответствии с учебно-методическим пособием [1].

Для удобства чтения будут спроектированы несколько диаграмм прецедентов для разных типов акторов. Акторы — это внешние объекты, взаимодействующие с системой.

Выделим три актора для разрабатываемой ИС:

– гость. Незарегистрированный пользователь, взаимодействующий с системой;

– авторизованный пользователь. Пользователь прошедший процесс авторизации;

– модератор. Пользователь имеющие особые права по модерации контента системы.

Отметим, что все акторы связаны отношением наследования. Так гость находится на верхушке иерархии и модератор его последний наследник.

Спроектируем диаграмму прецедентов для гостя и авторизированного пользователя (рис. 1).

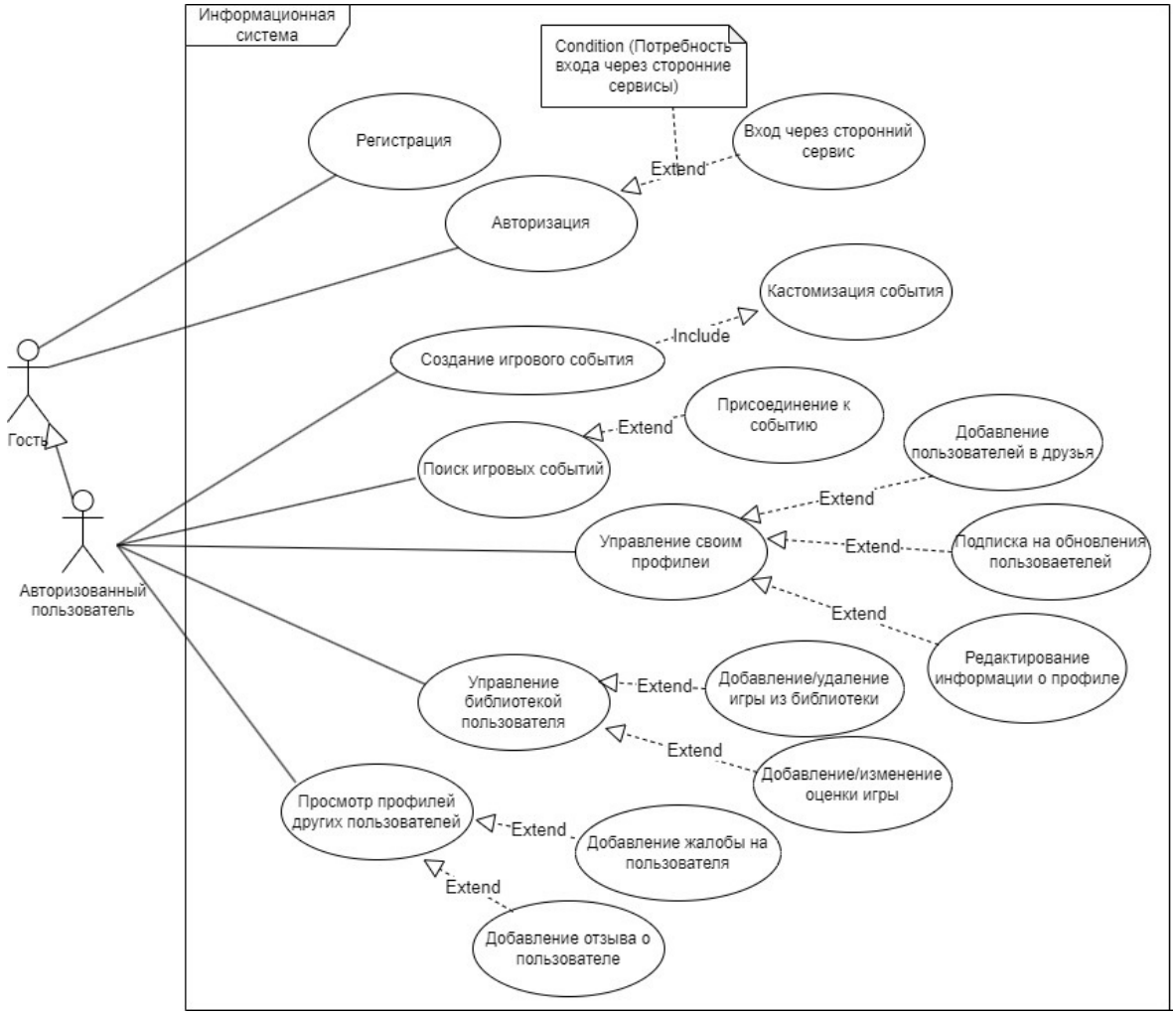


Рис. 1. Диаграмма прецедентов для Гость и Авторизированный пользователь

Основные прецеденты:

– регистрация (регистрация в системе для пользователя гостя);

– авторизация (вход пользователей в систему. Расширяется дополнительным прецедентом вход через сторонний сервис);

– создание игрового события (создание и публикация игрового события. Включает в себя прецедент кастомизация события, который отвечает за персональную настройку текущего события);

– поиск игровых событий (поиск игровых событий по заданным критериям фильтрация. Расширяется прецедентом присоединения к событию);

– управление своим профилем (управление профилем пользователя. Расширяется такими прецеденты как добавление пользователей в друзья, подписка на обновления пользователей, редактирование информации о профиле);

– управление библиотекой пользователя (управление игровой библиотекой пользователя. Расширяется прецедентами добавление/удаление игры из библиотеки, добавление/изменение оценки игры);

– просмотр профилей других пользователей (изучение игровых профилей других пользователей. Расширяется прецедентами добавление жалобы на пользователей, добавление отзыва о пользователе).

Остаётся обозначить прецеденты для актора модератора (рис. 2).

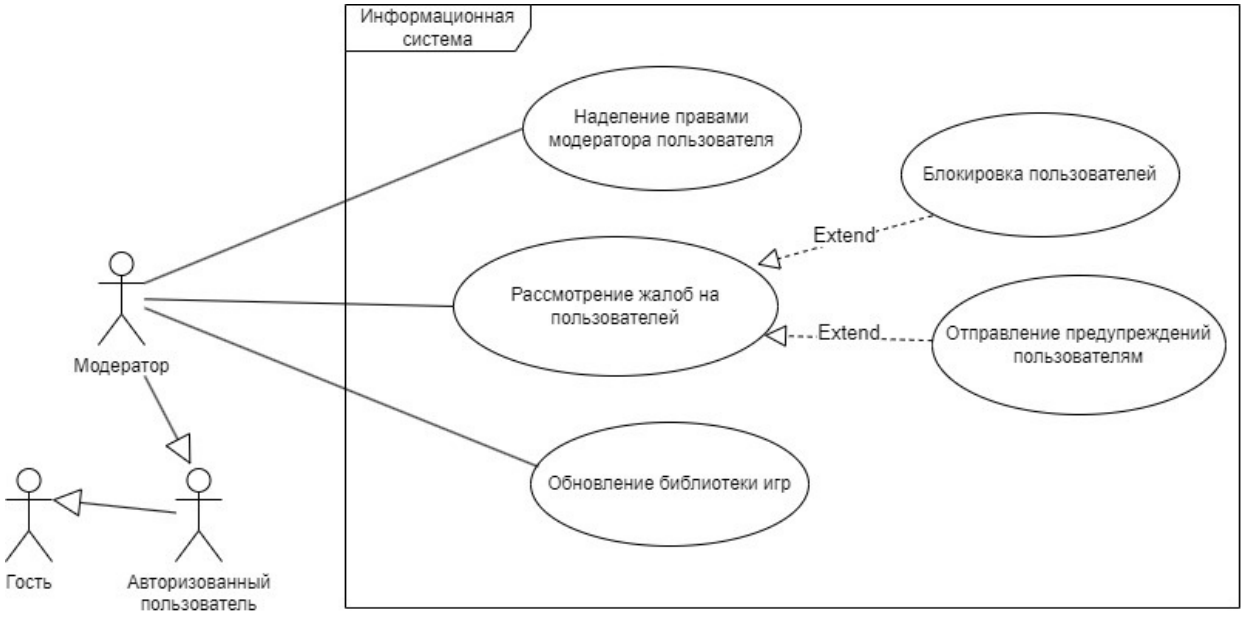


Рис. 2. Диаграмма прецедентов для Модератора

Основные прецеденты:

– наделение правами модератора пользователя (обновление прав зарегистрированного пользователя до модератора);

– рассмотрение жалоб на пользователей (анализ жалоб на пользователей. Расширяется блокировкой пользователей на период времени, и возможным отправлением предупреждений пользователям);

– обновление библиотеки игр (актуализация общей библиотеки игр).

Для лучшего представления и понимание будущих процессов системы перейдём к созданию диаграмм активностей.

Проектирование диаграмм активности произведено с помощью нотации BPMN. Диаграмма активности позволяет более наглядно смоделировать случаи использования. Начнём с описания процесса создания игрового события (см. рис. 3). Нужно учесть, что создание игрового события будет доступно только авторизированным пользователям.

После моделирования создания игрового события, логично следует перейти к разработке диаграммы поиска событий (рис. 4). Для удобства чтения диаграмм процесс «Персонализация фильтра поиска игровых событий» был выделен как подпроцесс и описан на другой диаграмме (см. рис. 5).

Рассмотрев диаграмму фильтра поиска игровых событий, можно убедиться, что разные пользователи смогут настроить поиск игровых событий под свои потребности.

В информационных системах, где пользователи взаимодействуют между собой, никак не обойтись без хотя бы какой-нибудь системы модерации. В статье «How Collegiate Players Define, Experience and Cope with Toxicity» [2] авторы приходят к выводу, что токсичное поведение игроков в мультиплеерных играх нередкое явление. Более того, такое поведение может плохо сказываться на результативности игроков в соревновательных играх. Таким образом необходим механизм контроля недобросовестных пользователей. Используя диаграмму активности, смоделируем процесс рассмотрения жалоб модератором (рис. 6).

От описания процессов перейдём к проектированию сущностей системы.

Отобразим на ER-диаграмме основные сущности в контексте системы (рис. 7).

Было выделено 8 основных сущностей в системе. На диаграмме можно рассмотреть существующие связи между сущностями.

Разработанная диаграмма при разработке базы данных послужит опорной структурой. Также диаграмма будет использоваться при разработке системных классов.

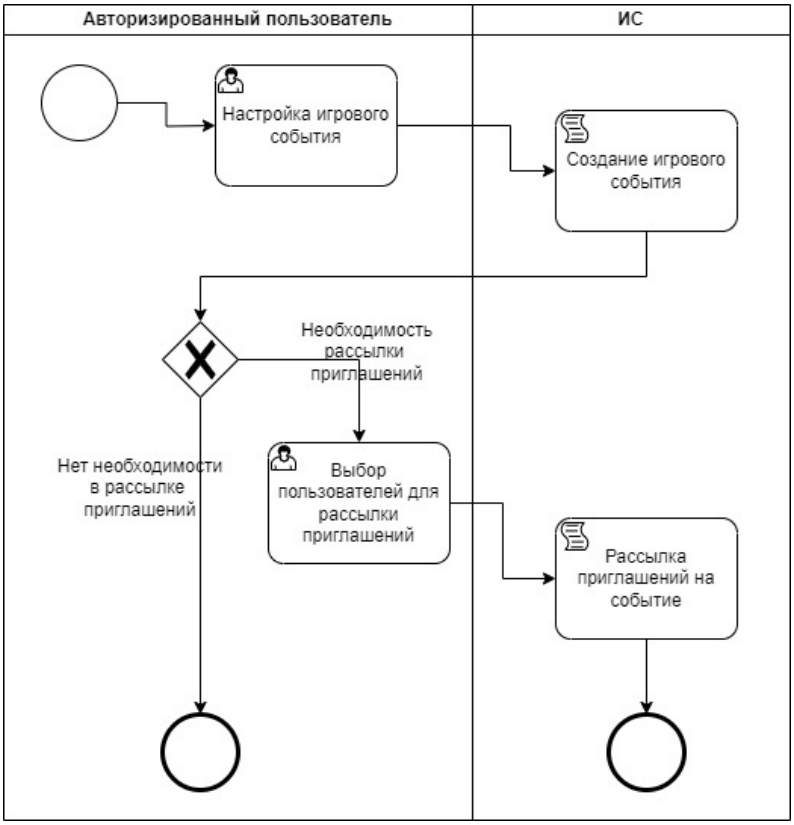


Рис. 3. Диаграмма активности создание игрового события

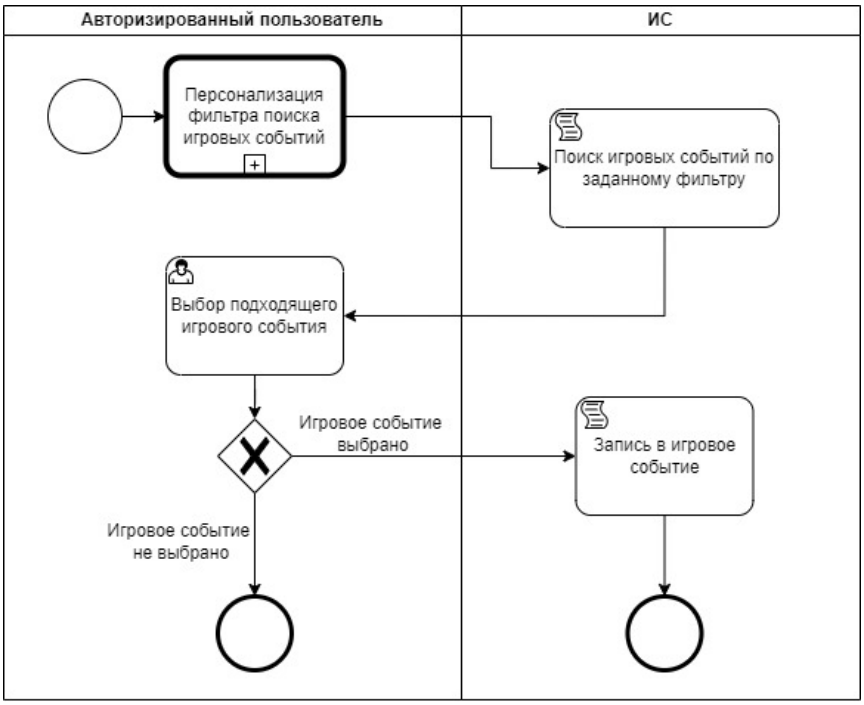


Рис. 4. Диаграмма активности поиск игровых событий

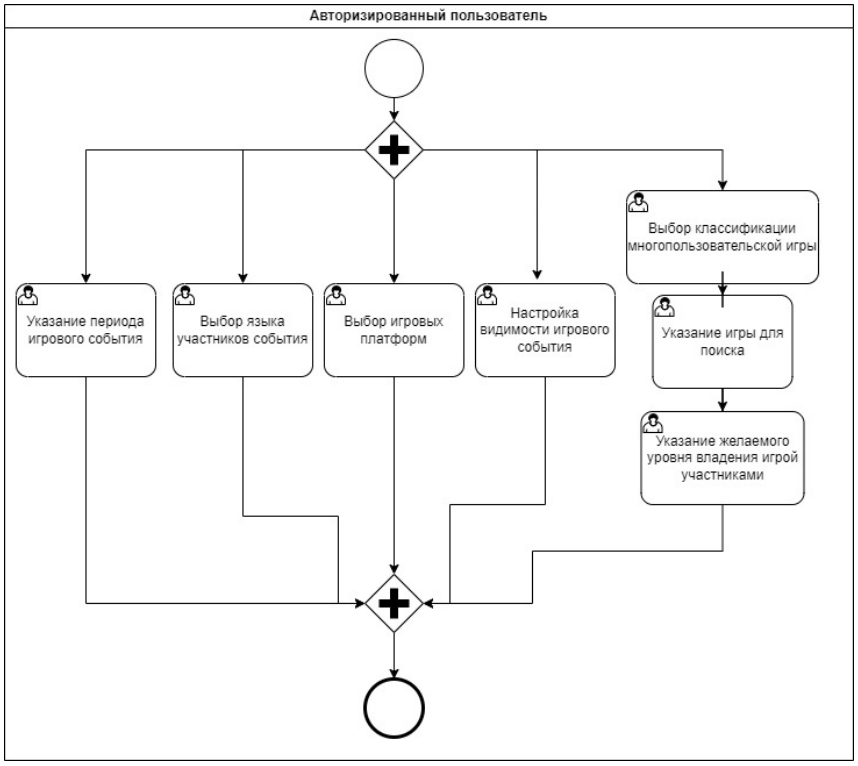


Рис. 5. Диаграмма активности подпроцесс фильтра поиска игровых событий

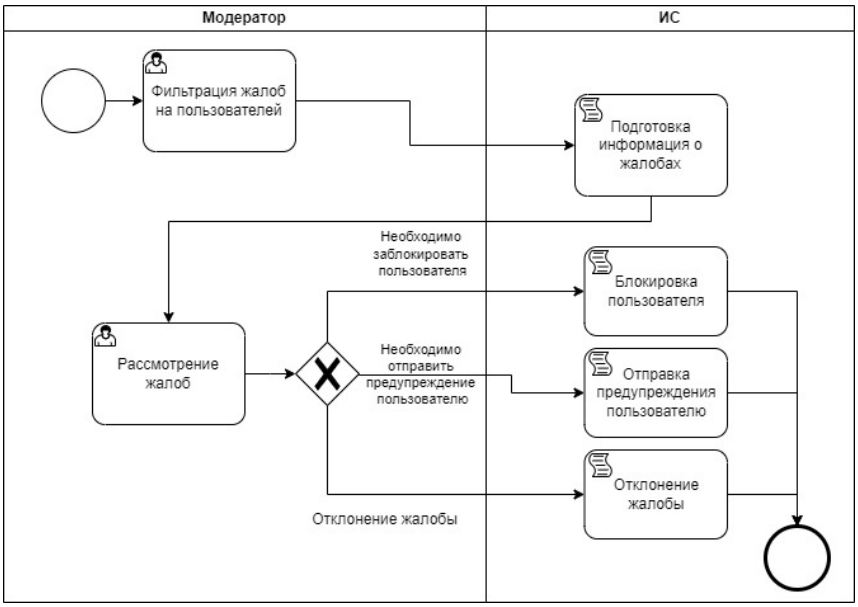


Рис. 6. Диаграмма активности рассмотрение жалоб

Заключение

В представленной работе был отображён комплексный подход к проектированию информационной системы для управления игровыми событиями. Разработанные диаграммы прецедентов, активности и ER-диаграмма позволяют структурировать процессы взаимодействия пользователей с системой, учесть разнообразные требования и оптимизировать архитектуру на этапе разработки.

Рассмотренные в работе модели демонстрируют, как различные типы пользователей — гости, авторизованные пользователи и модераторы — взаимодействуют с функционалом системы. Диаграммы прецедентов помогают выделить ключевые сценарии использования, такие

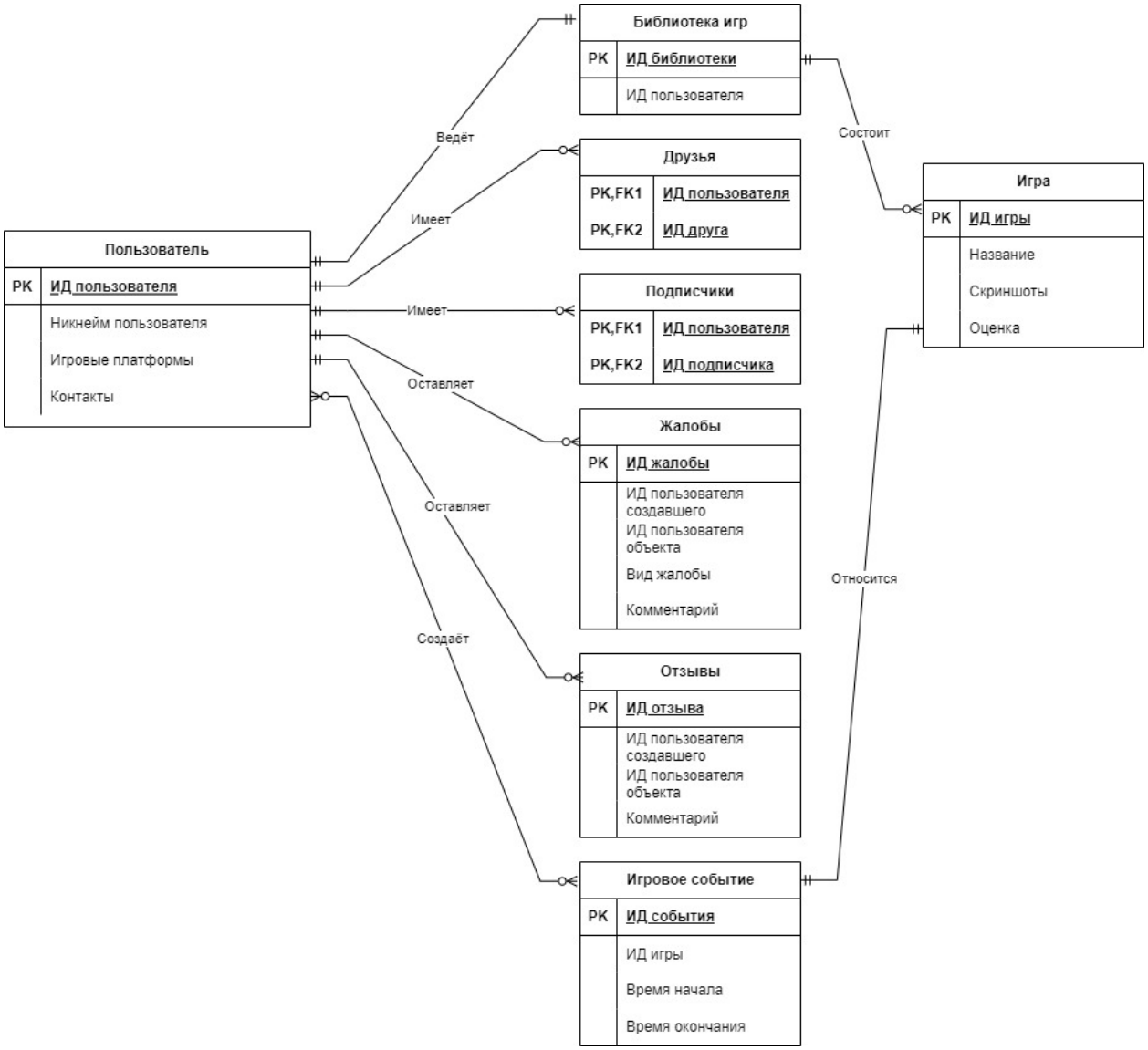


Рис. 7. ER–диаграмма

как регистрация, поиск и создание игровых событий, управление профилем и библиотекой игр, а также процессы модерации. Использование диаграмм активности в нотации BPMN позволило более детально смоделировать основные процессы, такие как персонализация поиска событий и обработка жалоб. Построенная ER-диаграмма служит основой для проектирования базы данных, обеспечивая её соответствие функциональным требованиям системы.

Литература

1. Основы проектирования и реализации информационных систем: учебно-методическое пособие / Н. Н. Дацун, А. В. Отинов, Т. С. Гашева, Д. И. Власов; Пермский государственный национальный исследовательский университет. –– Пермь, 2021. – 132 с.

2. See No Evil, Hear No Evil, Speak No Evil: How Collegiate Players Define, Experience and Cope with Toxicity / Selen Türkay, Jessica Formosa, Sonam Adinolf, Robert Cuthbert, Roger Altizer [Электронный ресурс] // CHI ‘20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems – URL: https://dl.acm.org/doi/10.1145/3313831.3376191 (Дата обращения 19.11.2024)