2023

77 РЕГИОНАЛЬНАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ СТУДЕНТОВ, АСПИРАНТОВ И МОЛОДЫХ УЧЕНЫХ

СТУДЕНЧЕСКАЯ ВЕСНА

СБОРНИК НАУЧНЫХ СТАТЕЙ

ТОМ 1

VESNA.SUT.RU

СПбГУТ)))

УДК 061.3(082)
ББК 74.58

77-я региональная научно-техническая конференция студентов, аспирантов и молодых ученых «Студенческая весна – 2023» : сб. науч. ст. / Под ред. Р. В. Киричка; сост. А. А. Нестеров, А. А. Задорожняя. Т. 1. СПб. : СПбГУТ, 2023. 264 с.

В научных статьях участников конференции исследуются состояние и перспективы развития мирового и отечественного уровня IT и телекоммуникаций. Предназначено студентам, аспирантам и специалистам отрасли связи.
Издание изготовлено оргкомитетом конференции.

Корректура и верстка Е. М. Аникевич.
Подписано в печать 01.11.2023.
Вышло в свет 30.11.2023. Формат 60×90 1/8.
Уст. печ. л. 16,5. Заказ № 103-ИТТ-2023.
пр. Большевиков, д. 22, корп. 1
Россия, Санкт-Петербург, 193232

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ

Председатель
Абилов А. В. – кандидат технических наук, доцент, первый проректор – проректор по учебной работе

Заместитель председатель
Нестеров А. А. – начальник управления организации научной работы и подготовки научных кадров

Ответственный секретарь
Задорожняя А. А. – начальник отдела организации научной работы студентов

Члены организационного комитета
Касаткин Ф. Ю. – проректор по цифровой трансформации
Григорян Г. Т. – начальник управления маркетинга и рекламы
Васильева Л. А. – начальник учебно-методического управления
Пономарева Е. Ю. – начальник редакционно-издательского отдела
Аникевич Е. А. – начальник отдела организации научно-исследовательской работы и интеллектуальной собственности

ПРОГРАММНЫЙ КОМИТЕТ

Председатель
Киричек Р. В. – доктор технических наук, доцент, ректор

Заместитель председателя
Макаренко С. И. – доктор технических наук, доцент, проректор по научной работе

Ответственный секретарь
Елагин В. С. – кандидат технических наук, доцент, доцент кафедры инфокоммуникационных систем

Члены программного комитета
Кирик Д. И. – кандидат технических наук, доцент, декан факультета радиотехнологий связи
Окунева Д.В. – кандидат технических наук, декан факультета инфокоммуникационных сетей и систем
Зикратов И. А. – доктор технических наук, профессор, декан факультета информационных систем и технологий
Владыко А. Г. – кандидат технических наук, доцент, декан факультета фундаментальной подготовки
Сотников А. Д. – доктор технических наук, профессор, декан факультета цифровой экономики, управления и бизнес-информатики
Гирш В. А. – начальник военного учебного центра
Шутман Д. В. – кандидат политических наук, доцент, декан гуманитарного факультета
Кривошеев А. В. – руководитель центра перспективных проектов и разработок

Неисключительные права на все материалы, опубликованные в данном издании, принадлежат СПбГУТ. Все материалы, авторские права, которые принадлежат СПбГУТ, могут быть воспроизведены при наличии письменного разрешения от СПбГУТ. Ссылка на первоисточник обязательна. По вопросам приобретения неисключительных прав и использования сборника обращайтесь по тел. (812) 312-83-79, e-mail: sno@sut.ru. Тип компьютера, процессор, сопроцессор, частота: Pentium IV и выше / аналогичное; оперативная память (RAM): 256 Мб и выше; необходимо на винчестере: не менее 64 Мб; ОС MacOS, Windows (XP, Vista, 7, 8, 10) / аналогичное; видеосистема встроенная; дополнительное ПО: Adobe Reader версия от 7.X или аналогичное. Защита от незаконного распространения: реализуется встроенными средствами Adobe Acrobat.

© СПбГУТ, 2023

Инфокоммуникационные сети и системы

6 **Амолов Т. Х., Батышев И. Р., Демин Д. Д., Евсиков В. А., Фомичев Е. О.** Блокчейн и AI: прототипы взаимодействия

10 **Антипина С. О., Потемкина Ю. Ф., Строило А. Ю.** Обзор алгоритмов распознавания рукописных подписей

15 **Баранов С. Н., Смирнов С. В., Тарасова Д. В., Целоусов Г. А., Штейнберг И. В.** Ассемблер: вечная основа или устаревший подход?

19 **Белов М. С.** Динамические эффекты в усилителях EDFA при резком изменении входной мощности

22 **Бен Гедрия М., Кальницкий И. А., Мартынова Е. В., Силина А. А.** Искусственный интеллект для транзакций: варианты применения в современных условиях

26 **Бреднева О. В., Земцов А. А., Пестряева Е. С., Ремденок П. А.** Исследование преимуществ технологии Thunderbolt перед существующими технологиями высокоскоростного подключения устройств

29 **Брусиловский А. С., Волосников К. Д., Добродеев Г. С., Никель Н. Д., Щербаков А. Д.** Особенности 3D принтеров и формирования моделей для них

33 **Бугрова Е. С., Романюк Е. О., Шестакова В. А.** Отслеживание атаки типа отказ в обслуживании и их симуляция на облачной инфраструктуре OpenStack с использованием системы мониторинга Grafana

39 **Важник М. М., Галкина В. Д., Райхерт В. Е., Рыбкина Е. В., Хмиль Д. И.** Ассемблер: «за» или «против»

44 **Васильев Н. С., Клименко В. П.** Сравнительный анализ амплитудной и фазовой бинарной модуляции цифровых оптических сигналов

49 **Воронцов А. А., Шемякин С. Н.** Использование идеального шифра в канале обратной связи в квантовой криптографии

54 **Восьмаков Д. И., Сергеев Н. В., Якубова О. А.** Разработка системы мониторинга городской инфраструктуры

59 **Дворцов А. С., Киселев Д. В., Мамлеев А. И.** Side Chain – перспективы и варианты применения

62 **Дубовиков Л. О., Серебряный И. Е., Юклутов Д. А.** Применение инерциальной навигации в системах LBS

67 **Ефременко К. С., Кузьмин А. С.** Обзор существующих концепций распределенных вычислений

71 **Зайцев И. А., Катеринчук Л. Е., Назарук В. С., Смирнов А. В., Шилов М. А.** Машинное обучение – ключевые алгоритмы, современные инструменты и вариативность применения

76 **Зуев Д. П., Шарифов Р. Г.** Разработка программы распознавания подлинности рукописной подписи

81 **Зырянова К. А., Ким И. А., Черненко С. В.** Исследование актуальности медицинских приложений

86 **Кокин Ю. И., Мартынов М. В., Тарасов И. С., Суменков М. С.** Haskell – потенциал и влияние на современное программирование

90 **Комар М. В.** Применение волоконно-оптических технологий в системах экологического мониторинга промышленных предприятий

95 **Краснорудский И. В., Николаев Р. М., Сумин А. С.** Машинное обучение – ключевые алгоритмы, современные инструменты и вариативность применения

99 **Кузин Д. И.** Голосовые помощники: вопросы безопасности, конфиденциальности и пути решения

104 **Махмутхажиев У. У., Соколов Е. А., Шейде Г. С.** Разработка iOS-приложения: влияние QA на конечный продукт

108 **Медведев И. Н.** Использование оптического BER-тестера для определения качества протяженной оптической линии связи

112 **Платонов А. Е.** Угрозы информационной безопасности, связанные с незаконным оборотом SIM-карт

117 **Сединкин И. Б.** Искусственный интеллект и кибербезопасность: взгляд на будущее борьбы с новыми видами угроз

122 **Сиваков В. В.** Исследование причин замедления темпов роста вычислительной мощности микропроцессоров

126 **Третьякова А. С.** Вопросы нарушения законодательства РФ в ГИС ЖКХ
130 **Фраз А. В.** Применение спектроанализаторов в системах связи с технологией DWDM

Информационные системы и технологии

134 **Баженов С. Г., Ефимова А. В., Павлюченко С. А., Шумских Л. А.** Крипто-преступления и технические методы их пресечения: развитие технологий для борьбы с киберугрозами
137 **Байзигитов М. М., Пенькова-Печерская С. П., Соловьев Д. М., Одинцова Ю. В.** Введение в стартап и разработка его MVP
142 **Баряев А. А., Курносов А. А., Михеев Р. С., Петров Д. А., Скоба Е. И.** Применение искусственного интеллекта в задачах теории графов
147 **Боклаженко А. А., Григорьев Е. А., Ильин В. В., Шаров Е. А.** Исследование методов процедурного разбиения плоскости
151 **Верещагина И. Г.** Информационные системы управления логистикой на предприятии
156 **Внуков И. А., Малдыбаев Р. А., Жаранова А. О.,Матвеев П. В.** Реализация основных компонентов сервиса для сопровождения образовательного туризма
162 **Восьмаков Д. И., Уласик В. В.** Прототип микроприложения хранения и обработки информации о научных публикациях
164 **Гайбалиев Э. Э., Жаранова А. О.** Проектирование информационной системы для поиска музыки по ритму
168 **Гончаров С. В., Широпатин О. И.** Создание моделей и исследование топологических операций в PostGIS
172 **Горбунов И. И., Емельянов К. А., Кириков П. В., Ленкин М. А., Межуев Р. А.** Java: современное применение и ключевые направления при разработке
177 **Грибовский А. А.** Разработка программного обеспечения для создания пользовательского интерфейса на базе технологии Blazor WebAssembly
180 **Доманцевич Н. С., Проскуряк В. А., Саидов А. А.** 3D-принтер: перспективы развития и ключевые технические принципы
184 **Ершова Н. С., Ноянов Р. С.** Телеграм-боты, как средство автоматизации элементов образовательного процесса
187 **Жаранова А. О., Кужлев М. Ю.** Проектирование информационной системы автоматизированного дистанционного мониторинга здоровья пациента
192 **Жаранова А. О., Сагиданов М. Д.** Проектирование информационной системы панели управления IPTV
196 **Жарова Ю. Е., Самолетова Д. В., Федоров А. П.** Машинное обучение. Сравнение двух основных подходов к обучению нейронных сетей
200 **Изюмова М. А., Жаранова А. О.** Проектирование веб-расширения по формированию персональной подборки книг
206 **Ильин И. В., Перебейнос Е. И.** Исследование современной архитектуры кроссплатформенных мобильных приложений на React Native
209 **Кильдяев И. В., Мельникова Д. М.** Принцип построения самоорганизующейся mesh-сети с использованием технологии Yggdrasil
213 **Молодцова Ю. С., Муратов М. Е.** Сравнительны анализ различных методов шифрования для безопасной передачи данных
220 **Жаранова А. О., Нерман К. А.** Проектирование информационной системы трекера задач с системой вознаграждения
225 **Рыбалко А. В.** Разработка аппаратно-программного устройства контроля состояния сотрудника аварийно-спасательной службы
231 **Жаранова А. О., Солдатов А. А.** Проектирование информационной системы корпоративного видеочата
236 **Толоконникова Е. П.** Разработка метода контроля времени, проводимого в социальных сетях
239 **Шакурова А. А.** Разработка модели парсинга данных
245 **Ширин А. И.** Разработка программного средства для системы сопровождения транспортной логистики масштаба мегаполиса

249 **Шохина М. А.** Разработка блока управления на базе ПЛИС для интеллектуальной системы освещения помещений предприятий
255 **Шульгин Ф. Э.** Методы оптимизации графика движения транспорта
260 **Жаранова А. О., Яремко Д. И.** Проектирование информационной системы по аренде квартир

УДК 004.41

Т. Х. Амолов, И. Р. Батышев, Д. Д. Демин, В. А. Евсиков,
Е. О. Фомичев (студенты гр. ИКПИ-11, СПбГУТ)

БЛОКЧЕЙН И AI: ПРОТОТИПЫ ВЗАИМОДЕЙСТВИЯ

Данная работа посвящена исследованию прототипов взаимодействия блокчейна и искусственного интеллекта. В работе рассмотрены возможности блокчейн-технологий для обеспечения безопасности данных и транзакций, а также способы использования искусственного интеллекта для улучшения процессов обработки и анализа данных. Описаны примеры применения этих технологий в различных отраслях. В работе также рассматривается использование блокчейна и AI для расчета вероятности уплаты кредита, рассмотрение преимуществ. Исследованы преимущества и недостатки данной системы.

блокчейн, искусственный интеллект, смарт-контракт, скоринг, кредит.

Блокчейн – это децентрализованная и распределенная база данных, состоящая из цепочки блоков, где каждый блок содержит набор транзакций или информацию. Каждый блок связан с предыдущим блоком с использованием криптографических методов, образуя непрерывную цепочку блоков.

Машинное обучение – это подраздел искусственного интеллекта, который занимается разработкой алгоритмов и моделей, позволяющих компьютерным системам автоматически обучаться и улучшать свою производительность на основе опыта и данных, без явного программного указания.

Взаимодействие блокчейна и искусственного интеллекта дает возможность создать децентрализованные автономные систем. Блокчейн обеспечивает прозрачность и безопасность данных, а искусственный интеллект предоставляет возможность создать интеллектуальные алгоритмы и модели, способные принимать решения и выполнять действия на основе этих данных.

Теоретические примеры использования блокчейна и AI

Голосование на основе блокчейна с использованием AI – в некоторых проектах применяется комбинация AI и блокчейна для обеспечения безопасного и прозрачного голосования. Искусственный интеллект может использоваться для проверки подлинности и подтверждения идентичности избирателей, а блокчейн записывает результаты голосования, обеспечивая надежность и невозможность подделки данных.

Управление умными контрактами с помощью AI – умные контракты, основанные на блокчейне, могут быть усилены с помощью искусственного интеллекта для автоматизации и более интеллектуального выполнения

условий контрактов. AI может использоваться для анализа больших объемов данных, автоматического регистрации и обработки событий, а также для предсказания результатов и принятия решений в контексте умных контрактов.

Анализ данных блокчейна с использованием AI – блокчейн хранит огромное количество данных, и искусственный интеллект может быть применен для анализа этих данных и извлечения ценной информации. AI может помочь обнаружить аномалии, выявить паттерны, предсказать тенденции и оптимизировать процессы на основе данных блокчейна.

Идентификация и аутентификация с использованием AI и блокчейна – AI может быть использован для биометрической идентификации, а блокчейн может обеспечить безопасное хранение и передачу данных об идентификации. Это может применяться в различных областях, таких как финансы, медицина и кибербезопасность, где важно обеспечить достоверность и безопасность личных данных.

Уже реализованные прототипы взаимодействия блокчейна и AI в разных сферах:

здравоохранение и медицина. Блокчейн и AI могут совместно применяться в здравоохранении для обеспечения безопасного и эффективного обмена медицинскими данными. Блокчейн может использоваться для хранения и обмена медицинских записей, а AI может помочь в анализе этих данных для диагностики, прогнозирования заболеваний и персонализации лечения.

финансовые услуги. В финансовой сфере блокчейн может использоваться для улучшения прозрачности и безопасности транзакций. AI может помочь в автоматизации и оптимизации процессов, например, в анализе кредитного риска, обнаружении мошенничества, автоматическом трейдинге и предсказании финансовых рынков.

снабжение и управление цепочкой поставок. Блокчейн может использоваться для создания прозрачных и безопасных цепочек поставок. AI в этом контексте может помочь в автоматическом анализе данных о поставках, прогнозировании спроса, оптимизации инвентаризации и управлении рисками. Комбинирование блокчейна и AI позволяет эффективно отслеживать и анализировать данные в реальном времени, улучшая прозрачность и эффективность в цепочке поставок.

В финансовой сфере существует большое количество проектов, в которых можно реализовать и блокчейн и искусственный интеллект, например, расчет вероятности выплаты кредита у заемщика.

Скоринг – это система оценки заемщика, с помощью которой банки и крупные микрофинансовые организации (МФО) могут предсказать, насколько аккуратно человек будет выплачивать кредит. В основе прогноза лежат математические расчеты и статистика.

Банки могут запрашивать информацию о человеке в государственных органах.

Обычно банки используют одновременно несколько систем скоринга. Выбор системы, подходящей для конкретного заемщика зависит от категории заемщика и вида кредита.

Оценив все параметры, банк получает вероятность, с которой заемщик выплатит кредит (скоринговый балл).

Для получения наиболее точной вероятности уплаты кредита, банк должен использовать как компьютерные модели, так и оценку квалифицированного персонала этого банка. Человек может оценивать дополнительные параметры, которые не прописаны в компьютерной модели.

Использование блокчейна в качестве базы данных, которая хранит информацию нужную для расчета скорингового балла, гарантирует одновременно безопасность и прозрачность, данные о заемщике будет знать только банк, при этом кредиторы получат точную информацию о своем клиенте.

Использование искусственного интеллекта для расчета вероятности уплаты кредита, обеспечивает эффективную обработку и анализ большого объема данных и позволяет выносить точный вердикт без участия работников банка.

Информация о клиентах будет хранится в смарт-контракте в виде массива структур. Каждая структура имеет поля, соответствующие параметрам нужным для оценки заемщика.

Смарт-контракт должен иметь модификатор, добавляющий новых владельцев, чтоб банки могли иметь доступ к private полям структур.

Также в смарт-контракт должны быть функции, которые принимают информацию о клиенте и сохраняют их в блокчейн, также которые шифрующуют и расшифровывают данные клиента.

Библиотека web3.js для Ethereum предоставляют набор инструментов для взаимодействия с блокчейном через API. С ее помощью будут отправлены данные из нейронной сети на блокчейн в качестве транзакции, а затем извлекать эти данные через API для обработки и использования в нейронной сети.

Для расчета скорингового балла можно взять уже готовый алгоритм машинного обучения LightGBM (*Light Gradient Boosting Machine*).

Алгоритм создает дерево решений. На каждой итерации построенное дерево решений делает предположение, на фоне этого предположения считается ошибка, с помощью этой ошибки на следующей итерации строится новое дерево, используя данные предыдущего. В конечном итоге, с помощью такого рода градиентного спуска с каждой итерацией падает процент ошибки. На соревновании люди смогли достичь точности в 0,805.

Обучается программа на наборе данных из 300+ тыс. записей историй кредитов с 122 категориями. Набор данных используется от объединенных организаций Home Credit Group.

В ходе исследования были выявлены следующие наиболее важные признаки при вынесении вердикта (рис. 1). Первое, второе и третье место заняли внешние источники данных, о которых Home Credit не распространяются. Дальше идут по убывающей возраст, размер кредита, ежегодная плата по кредиту, стоимость товара (на который оформляется кредит), сколько дней прошло с момента поступления на работу до подачи заявки на кредит, сколько дней прошло с момента смены паспорта до подачи заявки на кредит, дни со смены прописки, дни со смены номера телефона, пол (женский), образование (высшее), зарплата, тип кредита (наличными), возраст автомобиля, не имеет машину, население места проживания клиента, семейное положение (в браке).

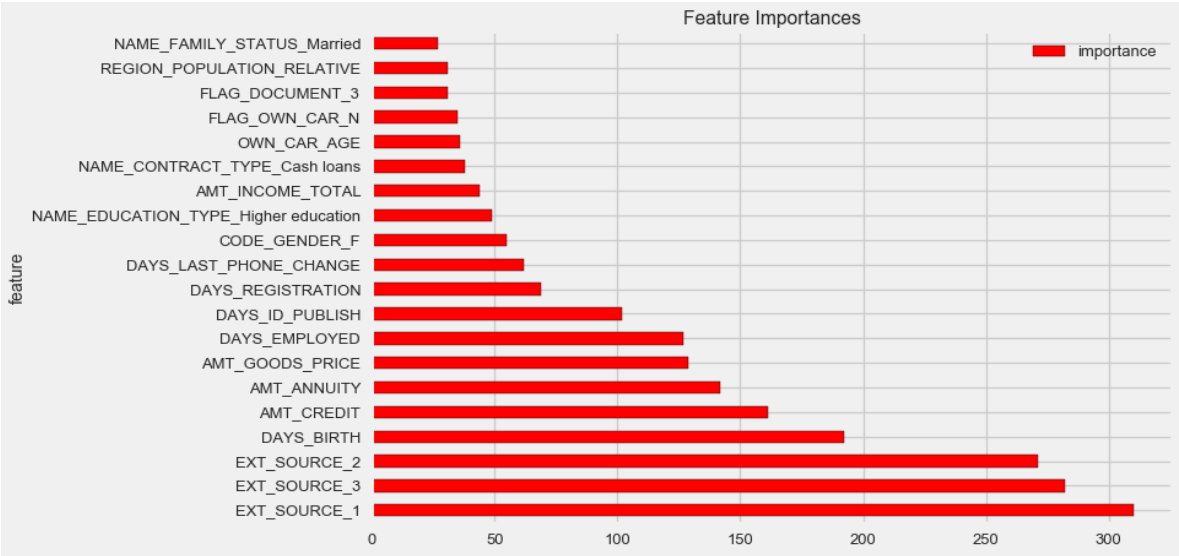


Рис. 1. Наиболее важные параметры при вынесении вердикта

Положительные качества данной теоретической системы:

- Возможность проведения процесса без сотрудников.
- Высокая скорость анализа данных.
- Прозрачная и защищенная система хранения данных в блокчейн системе.
- Высокая точность прогноза.
- Высокий потенциал развития.
- Не требует высоких вычислительных мощностей.

Отрицательные качества данной теоретической системы:

- Высокая сложность при введении системы в эксплуатацию.
- Задержка при получении информации из источников.
- Не в состоянии полностью заменить сотрудника банка

Заключение

Таким образом данный проект в настоящее время не будет пользоваться большой популярностью, несмотря на высокую точность, LightGBM

не сможет полностью заменить сотрудника банка, а в использовании блокчейна нет острой необходимости, так как на данный момент банк использует компьютерные модели, которые полностью удовлетворяют как клиентов, так и сотрудников банка. Хотя в целом AI и технология блокчейн в большинстве своем могут хорошо сказаться на отрасли, в которой будет использоваться их прототип взаимодействия.

Список используемых источников

1. Чуб Д. В. Использование технологии блокчейн при заключении и администрировании договора синдицированного кредита // Актуальные проблемы российского права. 2021. № 11. URL: https://www.elibrary.ru/item.asp?id=47290114 (дата обращения 10.05.2023).

2. Кудряшова Т. А., Федосова Т. В., Шульгина Е. А. Модель децентрализованной кредитной системы на основе блокчейн-технологии // Вопросы инновационной экономики. 2021. № 3. URL: https://www.elibrary.ru/item.asp?id=46618232 (дата обращения 13.05.2023).

3. Кальницкая И. В., Максимочкина О. В. Актуальные вопросы применения смарт-контракта как цифрового финансового инструмента // Фундаментальные исследования. 2022. № 5. URL: https://www.elibrary.ru/item.asp?id=48616539 (дата обращения 10.05.2023).

4. Городецкая О. Ю., Гобарева Я. Л. Ключевые тренды применения искусственного интеллекта в банковской сфере // Финансовые рынки и банки. 2022. № 12. URL: https://cyberleninka.ru/article/n/klyuchevye-trendy-primeneniya-iskusstvennogo-intellekta-v-bankovskoy-sfere (дата обращения 10.05.2023).

5. Морозова Г. В., Жогина К. А. Внедрение IT-технологий в процесс формирования кредитного рейтинга заёмщика // Контентус. 2021. № 2. URL: https://cyberleninka.ru/article/n/vnedrenie-it-tehnologiy-v-protsess-formirovaniya-kreditnogo-reytinga-zayomschika (дата обращения 13.05.2023).

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.01-004.93

С. О. Антипина, Ю. Ф. Потемкина, А. Ю. Строило
(студенты гр. ИКТЗ-16, СПбГУТ)

ОБЗОР АЛГОРИТМОВ РАСПОЗНАВАНИЯ РУКОПИСНЫХ ПОДПИСЕЙ

Подпись представляет собой один из самых распространенных методов для подтверждения документов. В настоящей работе представлен обзор алгоритмов распознавания рукописных подписей, который служит базой для дальнейшей разработки соответствующей программы. Представленный обзор направлен на подбор подходящего

алгоритма и служит предварительным теоретическим исследований для последующих практических работ.

подпись, верификация, подлинность, распознавание.

Существуют различные способы идентификации и последующей верификации подписей. Цель настоящей работы – исследование непосредственно алгоритмов распознания и дальнейший выбор подходящего алгоритма для последующей практической реализации.

Актуальность разработок обусловлена существующей необходимостью подтверждения подлинности рукописной подписи, несмотря на тенденцию к использованию электронных цифровых подписей (ЭЦП). Однако использование ЭЦП, в том числе квалифицированных, в настоящий момент не является общедоступным методом заверения документов.

В рамках обзора выявлен ряд открытых источников по данной тематике. Так, в статье [1] авторы исследуют возможность идентификации сигнатуры на основе ее фрактальной размерности Минковского. Подпись, рассматриваемая как траектория кончика пера, может быть описана с помощью уравнений Ланжевена, а ее фрактальная размерность Минковского может быть вычислена. На основе данного параметра можно различить подлинные и фальсифицированные подписи, что обеспечивает надежную идентификацию разных авторов. Предложенный подход, в сочетании с методами машинного обучения, представляет собой мощный инструмент для идентификации и верификации подписей и других форм обозначений [2].

В представленной научной статье [3] описывается разработка алгоритма, использующего искусственные нейронные сети (ИНС) для точного распознавания рукописных подписей на изображениях. Для оценки эффективности такого подхода, использовались сверточная и классическая ИНС. Первая сеть адаптирована для обработки двумерных поверхностей и обладает высокой степенью устойчивости к пространственным искажениям входных данных. Данная ИНС состоит из нескольких слоев, где первый слой представлен плоскостью, размеры которой соответствуют размерам входного изображения. По мере продвижения к последующим слоям, количество плоскостей увеличивается, а их размеры уменьшаются. Последние слои содержат обычные нейроны, которые выполняют окончательную классификацию признаков, выявленных в предыдущих слоях.

В исследованиях, проведенных в работе [3], рассматриваются различные конфигурации сверточных ИНС с учетом таких параметров, как количество слоев, количество плоскостей в слоях, количество связей между слоями и функции активации нейронов. В результате эксперимента было установлено, что для эффективного решения поставленной задачи наиболее предпочтительным оказывается использование сверточной ИНС без подвыборочных слоев. Это связано с тем, что такие ИНС могут потерять мелкие, но значимые характеристики рукописных подписей.

В статье [3], в процессе разработки классической ИНС, было проведено исследование различных конфигураций сети, учитывающих количество слоев, количество нейронов в каждом слое и количество связей между нейронами. Функция активации, примененная в данном исследовании, является гиперболическим тангенсом. Классическая ИНС, как описывается, представляет собой многослойный персептрон с архитектурой, состоящей из входного слоя, скрытого слоя и выходного слоя.

В статье [4] рассматриваются методы проверки подписи. Изображение с отсканированной собственноручной подписью, которое имеет уровень серого, показывает, что некоторые пиксели могут быть формами, написанными с высоким давлением и выглядеть более темными зонами [5]. Адаптивная сегментация интенсивности чернил учитывает влияние различных пишущих инструментов и позволяет проверять различные датчики пера. Таким образом, анализ текстуры следа чернил становится новым подходом для определения характера личного написания и улучшения процедуры проверки подлинности рукописной подписи. Биометрические системы, использующие верификацию подписи в сочетании с текстурным анализом, могут распознавать информацию о распределении чернильных пикселей, которая отражает личные характеристики подписывающего. Так возможно прийти к выводу, что распределение можно рассматривать как комбинацию информации о текстуре и форме подписи. Первый метод, который рассматривают в упомянутой статье – метод матрицы встречаемости серого уровня, который позволяет оценить плотность вероятности второго порядка, основываясь на одном изображении и предполагая, что плотность вероятности зависит исключительно от взаимного расположения.

Следующая мера, которая применяется к серому уровню – оператор локального бинарного паттерна (англ. *Local Binary Patterns, LBP*). LBP –метод, используемый для определения местоположения пикселя на изображении, который основан на анализе восьми ближайших пикселей вокруг центрального пикселя. Основные свойства LBP: инвариантность к монотонным изменениям уровня серого и вычислительная простота, позволяющая анализировать изображения в условиях реального времени.

Также в открытых исследованиях используется текстурный анализ для проверки подписи, при этом существует два варианта такой проверки: удаление фона путем постеризации или смещением гистограммы в сторону нуля.

В исследовании [5] обсуждаются перспективные методы для решения задач распознавания подписей – ИНС, которые способствуют автоматизации при проверке подписей, а также создание Telegram-бота, который обеспечивает предварительную обработку данных для последующей реализации проверки с помощью ИНС.

В ходе исследования открытых источников в рамках настоящего обзора также выявлены различные способы подделки подписи – существуют визуальные характеристики, которые позволяют отличить оригинальную подпись от копии или подделки, что открывает возможности для применения технологий компьютерного зрения, основанных на алгоритмах машинного обучения для автоматической проверки подлинности подписей.

Также в ходе данной работы выявлено, что машинное обучение представляет три типа: обучение с учителем, обучение без учителя и обучение с подкреплением. Основная цель обучения с учителем состоит в предсказании закона, согласно которому каждому входному значению x ставится в соответствие выходное значение y, на основе предоставленного набора помеченных пар данных. Обучение без учителя подразумевает наличие входных данных и поиск неких закономерностей в них. Данная задача менее определена, поскольку заранее неизвестно, какие именно закономерности следует искать, и не существует метрики для оценки результатов работы алгоритма. В рамках задачи проверки подлинности подписи наиболее применимым видится обучение с учителем.

Кроме того, при проведении данного исследования рассмотрены методы ближайших соседей и методы k-ближайших соседей. Первый метод представляет собой наиболее простой метод классификации. В процессе тестирования указанных выше методов выяснено, что метод ближайших соседей заключается в том, чтобы присвоить тестовому изображению класс самого ближайшего соседа, а метод k-ближайших соседей присваивает тестовому изображению класс, который наиболее часто встречается среди k-ближайших соседей.

В результате настоящего исследования выявлено, что метод k-ближайших соседей не является оптимальным для работы с изображениями, так как расстояние между двумя изображениями не отражает их семантическую близость эффективно. Более подходящим решением является использование параметрического подхода, в котором каждому изображению x применяется функция $f(x, W, b)$. Данная функция выдает набор оценок вероятности принадлежности изображения x к каждому из классов, на которые обучена модель. Процесс обучения модели заключается в определении оптимального набора параметров W для данной задачи. Для оценки качества разных значений W используется функция потерь. Алгоритм, который ищет наилучшее значение W, называется алгоритмом оптимизации.

В рамках подготовки обзора проведено тестирование нейросетевой модели ConvNet для работы с изображениями, поскольку согласно данным из открытых источников результат работы этой модели достигает точности 92,7 %. Для решения задачи определения подлинности подписи была использована модель ConvNet, известная как VGG19, в ее конфигурации E.

Данная конфигурация показывает отличные результаты в задаче классификации на датасете ImageNet. Чтобы сэкономить вычислительную мощность и время обучения, применена технология трансферного обучения.

При анализе различных глубоких ИНС, которые были обучены на фотографиях, замечено, что первые слои модели обрабатывают одни и те же особенности изображений. В то же время, последние слои ИНС специфичны для каждой отдельной задачи. Данное наблюдение привело к выводу, что необязательно обучать каждую новую модель с самого начала, можно использовать уже обученные первые слои моделей.

Таким образом, настоящий обзор является предварительным – проведено первичное исследование, которое может быть использовано в качестве теоретической базы для дальнейших разработок. В рамках практической реализации, представленной в отдельной статье, выполнена разработка программы автоматического распознавания и верификации рукописной подписи на основе размерности Минковского, поскольку опередено, что при дальнейших исследованиях и разработках, в том числе в перспективе с применением методов машинного обучения, данный алгоритм может показать высокие результаты. Также следует отметить, что в рамках дальнейших работ планируется более детальное рассмотрение и применение представленных в настоящей работе алгоритмов.

Список используемых источников

1. Статья портала ITMO news. Как фрактальная размерность Минковского помогает определить подлинность подписи человека. 2019. URL: https://news.itmo.ru/ru/science/photonics/news/8529/ (дата обращения 10.05.2023).

2. Rudyi S., Vovk T., Rozhdestvensky Y. Signature identification by Minkowski dimension // Chaos. 2019. № 29 (5).

3. Друки А. А., Милешина М. А. Алгоритмы распознавания рукописных подписей на основе нейронных сетей // Фундаментальные исследования. 2014. № 11 (9). С. 1906–1910.

4. Ахунджанов У. Ю., Старовойтов В. В. Предварительная обработка изображений рукописных подписей для последующего распознавания // Системный анализ и прикладная информатика. 2022. № 2. С. 4–9.

5. Ложкин Л. Д., Анкина К. П. Проверка подписи на основе информации об уровне серого с использованием нейронной сети // StudNet. 2021. № 5. 116. URL: https://stud.net.ru/proverka-podpisi-na-osnove-informacii-ob-urovne-serogo-s-ispolzovaniem-nejronnoj-seti/

6. Левинский А. Т. Алгоритмы верификации рукописных подписей на основе нейронных сетей // StudNet. 2020. № 3. С. 395–402.

Статья представлена кандидатом технических наук, доцентом Зарубиным А. А.

УДК 004.42

С. Н. Баранов, С. В. Смирнов, Д. В. Тарасова, Г. А. Целоусов, И. В. Штейнберг (студенты гр. ИКПИ-15, СПбГУТ)

АССЕМБЛЕР: ВЕЧНАЯ ОСНОВА ИЛИ УСТАРЕВШИЙ ПОДХОД?

Работа посвящена изучению аспектов использования языка программирования низкого уровня – ассемблера – в качестве инструмента разработки современного программного обеспечения. Целью работы явилось определение актуальности ассемблера и исследование области его применения. В статье приводятся результаты опроса о способах применения ассемблера программистами, на основе которых выявляются тенденции применения языка для решения практических задач разными поколениями разработчиков. Также проводится анализ востребованности специалистов в области низкоуровневой разработки на рынке труда. Рассматриваются достоинства и недостатки использования ассемблера в сравнении с высокоуровневыми языками программирования.

ассемблер, разработка ПО, низкоуровневый язык программирования, процессор.

Язык ассемблера – средство для представления команд процессора в доступном для чтения виде [1]. Язык ассемблера принято считать языком программирования низкого уровня.

Ассемблером называется программа-транслятор, преобразующая код на языке ассемблера в машинный код.

Цель работы заключается в выяснении роли ассемблера в современной разработке. Гипотеза исследования – ассемблер по-прежнему является фундаментальным средством разработки ПО. В связи с этим были поставлены и решены следующие задачи:

– изучить современные области применения ассемблера;
– выявить преимущества использования ассемблера;
– выявить частоту использования языка ассемблера разработчиками;
– выявить степень востребованности специалистов в сфере низкоуровневой разработки на рынке труда;

Выделяют следующие современные области применения ассемблера:

1. Разработка операционных систем.
2. Разработка драйверов [2, 3].
3. Программирование периферийных устройств [2].
4. Написание антивирусов.
5. Программирование микроконтроллеров [4, 5].
6. Разработка программного обеспечения для критически важной инфраструктуры (АЭС, электростанции, дамбы, правительственные и силовые

структуры и ведомства), где на законодательном уровне запрещается использовать ПО сторонних разработчиков из-за возможности наличия недекларированных возможностей.

7. Создание визуального контента. Например, создание оконных приложений, а также компьютерное искусство, заключающееся в создании видеороликов, ограниченных определенными объемами исполняемого файла (64 kb, 4 kb, 1 kb) [4].

Язык ассемблера обладает следующими преимуществами:

1. Полный контроль и большая гибкость, чем при использовании других языков программирования [4].

2. Работа с процессором и памятью напрямую. Существуют платформы с ограниченными ресурсами, где оптимизация кода очень актуальна. В условиях ограниченности ресурсов (памяти/времени) ассемблер является оптимальным решением.

3. Возможность внедрения ассемблерного кода в другие процессы (с целью обнаружения уязвимостей при внедрении кода), создание антиотладочных приемов [5].

4. Принцип «Одна строка – одна инструкция процессора», который позволяет оптимальнее использовать ресурсы процессора по сравнению с языками высокого уровня [4].

Однако можно выявить следующие недостатки:

1. Чрезмерная громоздкость и большой объем исходного кода при решении масштабных задач [1].

2. Необходимость писать код под конкретную платформу [4].

Для определения популярности ассемблера среди молодых специалистов было проведено анкетирование, в котором приняли участие 112 человек, в том числе студенты технических направлений СПбГУТ и других вузов [6], с целью выяснить масштабы и области использования ассемблера.

В ходе тестирования были получены следующие данные: 85 % респондентов – люди в возрасте до 25 лет. Остальные 15 % старше 25 лет (рис. 1).

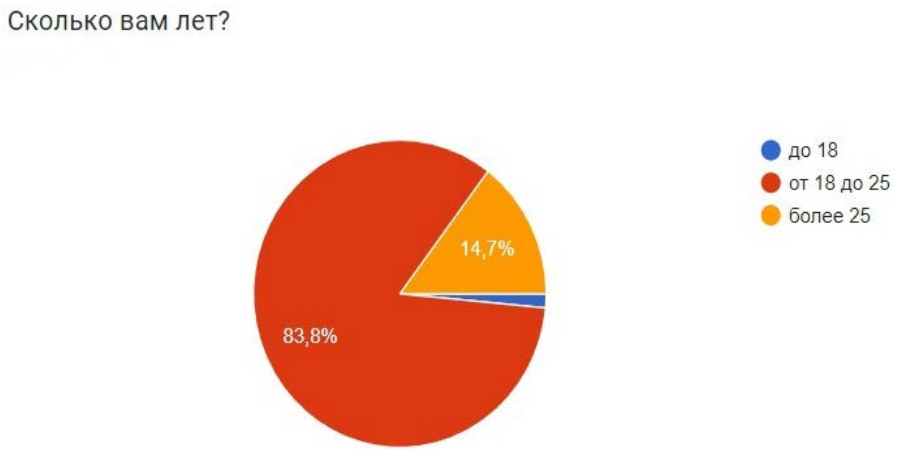


Рис. 1. Результаты опроса. Вопрос 1: Сколько вам лет?

Также большинство респондентов получают или имеют образование, связанное с разработкой ПО. Далее опрошенными будем называть именно их, т. к. для остальных опрос завершается.

Чуть меньше половины опрошенных (42 %) считают ассемблер актуальным средством разработки ПО и 88 % сталкивались с программированием на языке ассемблера (рис. 2). При этом доля респондентов, писавших код на языке ассемблера в рамках учебной программы, составляет 98 % (рис. 3).

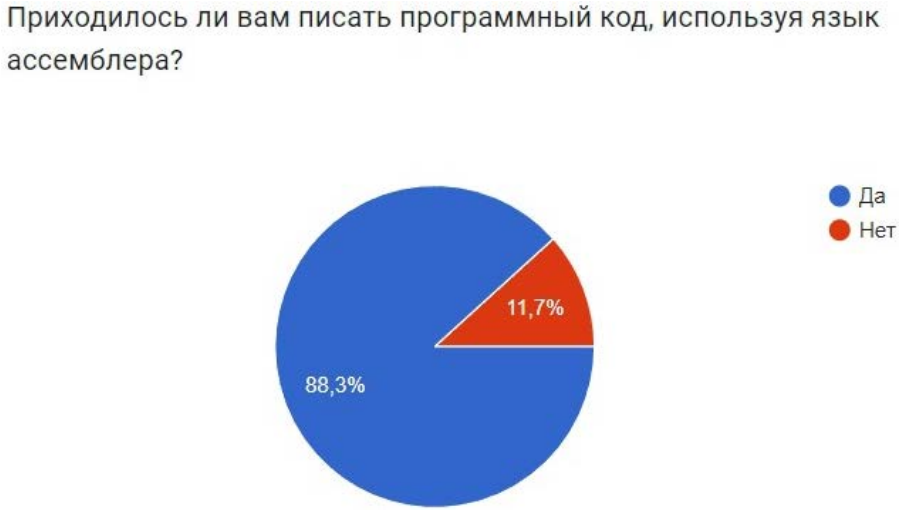


Рис. 2. Результаты опроса. Вопрос 4:
Приходилось ли вам писать программный код, используя язык ассемблера?

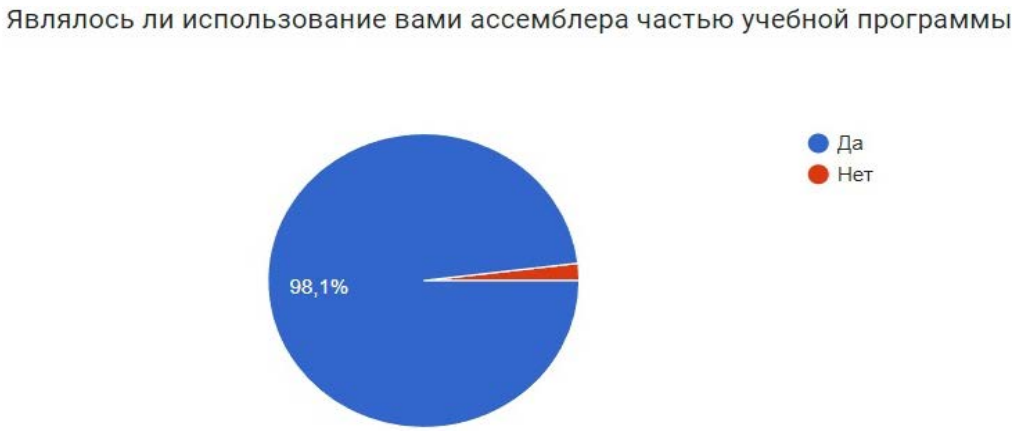


Рис. 3. Результаты опроса. Вопрос 5:
Являлось ли использование вами ассемблера частью учебной программы?

Также было проведено исследование рынка труда. На портале HeadHunter представлено 225 вакансий по запросу «Assembler» в России [7]. При этом все из них размещены за последний месяц, а 105 за неделю.

Зарплата в большей части предложенных вариантов колеблется в диапазоне от 70 до 150 тысяч рублей. Распространенные специальности –

reverse engineer, system/embedded developer, malware analyst (реверс-инженер, программист встроенных систем, аналитик вредоносного ПО и др.). Наиболее известными работодателями являются такие компании, как Доктор Веб, Сколково, ВТБ, Лаборатория Касперского и др.

Количество резюме, содержащих в себе ассемблер в качестве ключевого навыка, превышает отметку в 5 тыс., причем большая часть соискателей старше 30 лет.

В ходе анализа вакансий, не связанных с языком ассемблера, было выяснено, что знание ассемблера является преимуществом при трудоустройстве на должность разработчика.

В результате исследования было выявлено, что язык ассемблера обладает уникальными преимуществами в сравнении с высокоуровневыми языками, специалисты в области ассемблера востребованы на рынке труда. При этом сравнение результатов опроса и анализа рынка труда показывает, что ассемблер более популярен у старшего поколения разработчиков.

Таким образом, гипотеза исследования подтвердилась: ассемблер по-прежнему является фундаментальным средством разработки ПО.

Список использованных источников

1. Моргунова О. М., Абрамова О. Ф. Есть ли будущее у ассемблера? [Электронный ресурс] / 11-я Международная студенческая конференция «Студенческий научный форум – 2019». URL: https://scienceforum.ru/2019/article/2018010638

2. Зубков С. В. Assembler для DOS, Windows и Unix. ДМК Пресс, 2017. 638 с. ISBN 978–5–97060–535–6.

3. Аблязов Р. Программирование на ассемблере на платформе x86–64. ДМК Пресс, 2016. 302 с., ISBN 978-5-97060-364-2.

4. Красников Е. За что я люблю ассемблер? [Электронный ресурс] // 22.07.2021. Habr.Com. URL: https://habr.com/ru/articles/569204/ (дата обращения 08.05.2023).

5. Карев А. Погружение в ассемблер. Зачем учить ассемблер в 2020 году? [Электронный ресурс] // 20.10.2020. Xakep.Ru. URL: https://xakep.ru/2020/10/20/asm-course-8/ (дата обращения 08.05.2023).

6. Анкетирование молодых специалистов и студентов СПбГУТ. URL: https://docs.google.com/forms/d/e/1FAIpQLSdmA4lvOqFz8RwE156t-8JhHa4T5Jp9xgjgdn1IDPleMVp2lg/viewform (дата обращения 08.05.2023).

7. Портал HeadHunter, вакансии по запросу «Assembler». URL: https://hh.ru/search/vacancy?text=assembler (дата обращения 22.05.2023).

Статья представлена научным руководителем, доцентом кафедры ПИВТ СПбГУТ, кандидатом технических наук Бирюковым М. А.

УДК 621.391

М. С. Белов (студент гр. ИКТФ-26м, СПбГУТ)

ДИНАМИЧЕСКИЕ ЭФФЕКТЫ В УСИЛИТЕЛЯХ EDFA ПРИ РЕЗКОМ ИЗМЕНЕНИИ ВХОДНОЙ МОЩНОСТИ

В работе исследуются переходные эффекты в усилителях на эрбиевом волокне (EDFA), вызванные добавлением или удалением каналов в системе. Рассматриваемые эффекты оказывают влияние на работу систем с мультиплексированием по длине волны (WDM). Рассматриваются различные сценарии изменения числа каналов на входе усилителя и их влияние на параметры усиления и шумы. Основным методом исследования является компьютерное моделирование. Результат показывает переходный эффект мощности EDFA, когда число каналов изменяются вследствие сбоя или реконфигурации системы.

оптический усилитель на волокнах, легированный эрбий, динамические эффекты, система со спектральным мультиплексирование, EDFA.

В системе WDM существует проблема, что при удалении или добавлении оптических каналов в системе WDM с использованием EDFA мощность выходных сигналов, существующих или оставшихся каналов соответственно возрастает или уменьшается вследствие изменения насыщения EDFA [1].

Изменения мощности так же могут влиять на уровень ошибок битов. Производительность WDM-системы может серьезно пострадать от добавления или удаления каналов, поскольку все присутствующие каналы одновременно усиливаются каждым усилителем и делят имеющийся насыщенный коэффициент усиления. Цель данной статьи продемонстрировать, как изменяется усиление при резком изменении средней входной мощности.

В данной статье анализируются переходные эффекты с использованием компонента EDF Dynamic Analytical. Моделирование анализирует переходные эффекты, вызванные удалением и добавлением каналов в систему. Схема моделирования приведена на рис. 1 (см. ниже).

Волокно, легированное эрбием, обычно накачивается полупроводниковыми лазерами с длиной волны 980 нм или 1480 нм. Для лазеров с длиной волны 980 нм можно использовать трехуровневую модель, в то время как для лазеров с длиной волны 1480 нм обычно достаточна двухуровневая модель [2, 3].

Полная инверсия может быть достигнута при накачке длиной волны 980 нм, но не при накачке длиной волны 1480 нм [3].

Спонтанное время жизни метастабильного энергетического уровня составляет около 10 мс, что значительно медленнее скорости передачи сигналов, которая представляет практический интерес [4].

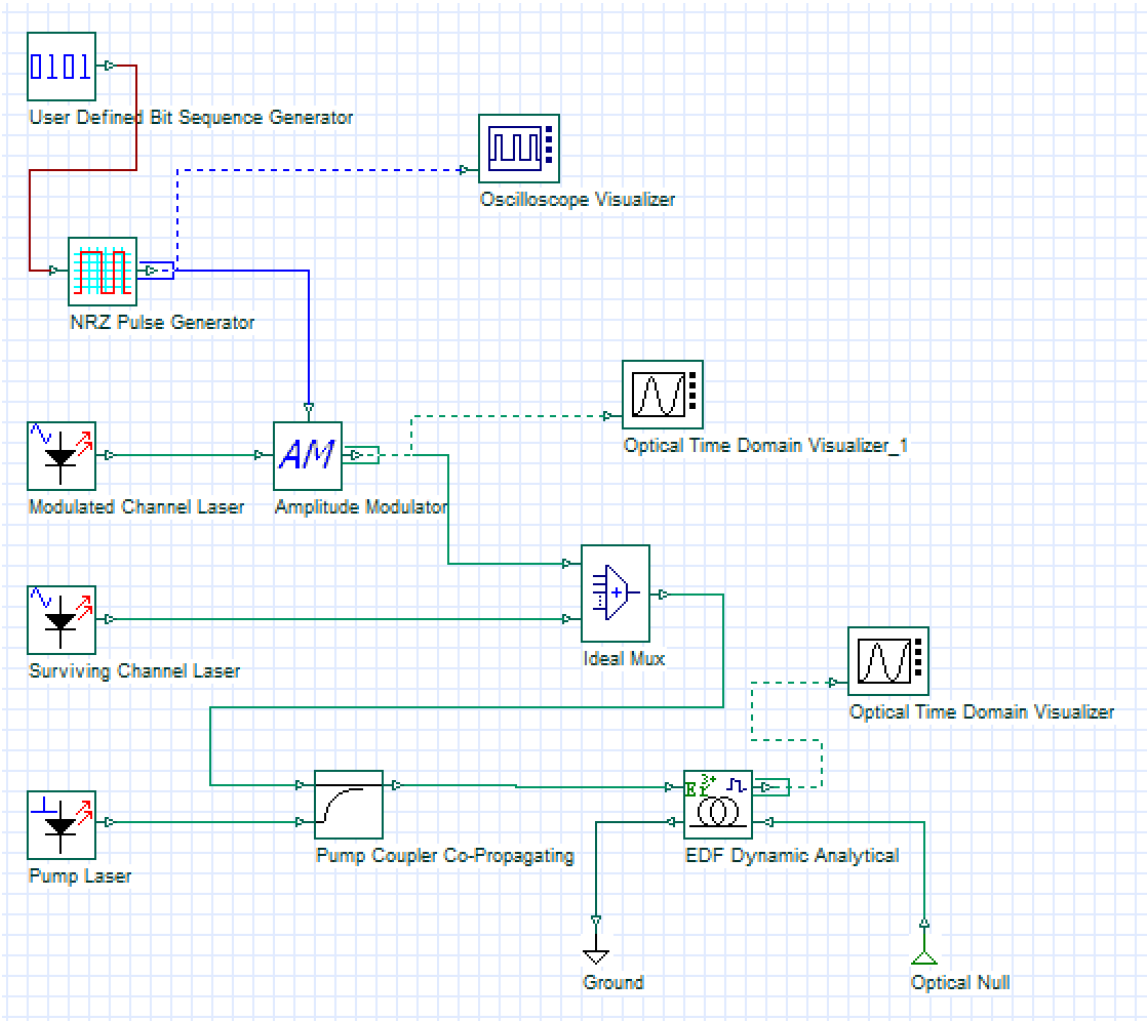


Рис. 1. Схема системы для исследования переходных эффектов при удалении и добавлении каналов

В схеме используется двухуровневая модель EDFA, в которой происходит накачка и спонтанный переход сигнала. Одним из основных факторов, влияющих на усиление, является характер входного сигнала с длиной волны 1550 нм. Кроме того, все сигналы, включая сигналы накачки и входные сигналы, оказывают влияние на процесс инверсии.

При излучении сигнала происходит истощение количества электронов на верхнем уровне, и требуется время для его восстановления посредством накачки. Эффекты насыщения и восстановления усиления приводят к переходным процессам и искажениям усиленного выходного сигнала.

При удалении канала мощность генерации на определенной длине волны увеличивается, а при добавлении канала она уменьшается. Удаление каналов оказывает влияние на активную среду и остальные каналы. Этот эффект наблюдается в течение спонтанного излучения и вызывает нарушение динамической структуры. В результате на верхнем уровне остается больше электронов, что приводит к изменению процесса усиления других каналов.

Приемник обладает определенным уровнем насыщения и минимальным допустимым уровнем. В случае резкого изменения сигнала, он может выйти за пределы этих уровней. Усилитель имеет возможность управления усилением и мощностью. Управление усилением позволяет компенсировать увеличение затухания путем увеличения коэффициента усиления, а управление мощностью позволяет усилить сигнал в случае его пропадания.

В схеме на рис. 1 используются:

лазер «Modulated Channel Laser», длина волны 1557,2 нм;

лазер «Surviving Channel Laser». Длина волны 1554 нм;

лазер накачки «Pump Laser», длина волны 1480 нм, мощность 17,4 дБм;

EDF «Er Doped Dynamic Analytical», длина 10 м, алюмогерманосиликатное стекло.

Различные пропорции мощности для модулированного канала на 1557,2 нм и оставшегося канала на 1554 нм приблизительно соответствуют разным комбинациям числа потерянных и оставшихся каналов. Предполагается, что суммарная мощность двух каналов соответствует 8 каналам, каждый из которых запускал –14,7 дБм выходной мощности в EDF. В таблице 1 представлено распределение входной мощности по каналам.

ТАБЛИЦА 1. Распределение входной мощности по каналам

Количество удаляемых/ добавляемых каналов	1	2	3	4	5	6	7
Входная мощность модулированного канала, дБм	–14,7	–11,69	–9,9	–8,68	–7,7	–6,9	–6,25
Входная мощность оставшегося канала, дБм	–6,25	–6,9	–7,7	–8,68	–9,9	–11,69	–14,7

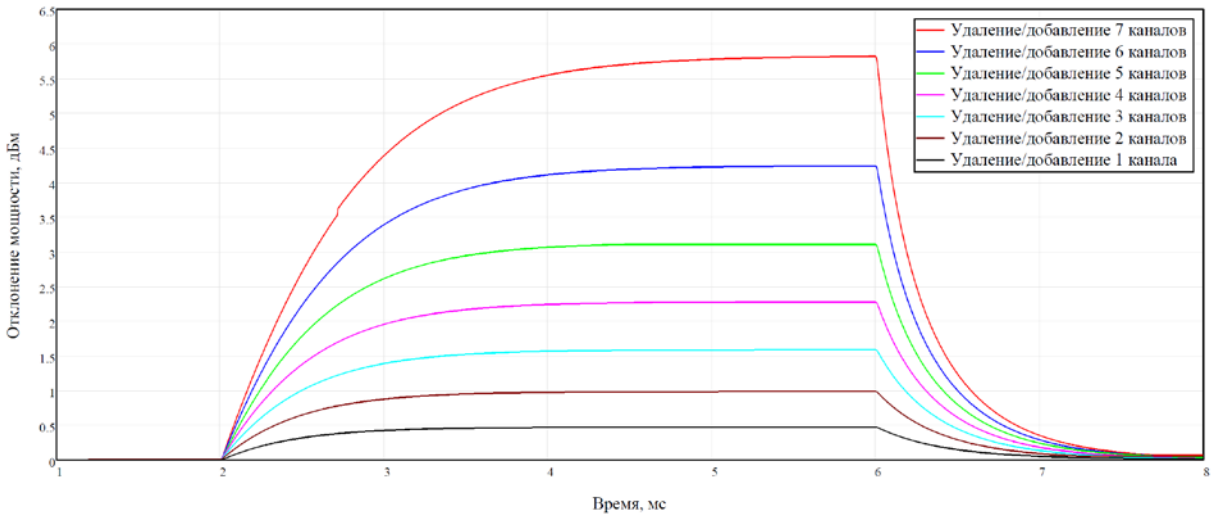


Рис. 2. Отклонение мощности остаточного канала при удалении и добавлении в процентах от общей входной мощности

Каналы были удалены и добавлены соответственно через 2 и 6 мс. На рис. 2 показаны результаты для различных значений отношения оставшейся мощности каналов к общей мощности (которая остается неизменной). Удаление каналов в системе вызывает увеличение выходной мощности оставшихся каналов. Добавление каналов в системе вызвало переходные процессы мощности на выходе оставшегося сигнала. Амплитуда скачка мощности пропорциональна количеству добавленных каналов.

Список используемых источников

1. M. Karasek and J.A. Valles, “Analysis of channel addition/removal response in all-optical gain-controlled cascade of erbium-doped fiber amplifiers,” J. Lightwave Technol., Vol. 16, pp. 1795–1802, 1998.

2. C. R. Giles and E. Desurvire, “Modeling Erbium-Doped Fiber Amplifiers,” J. Lightwave Tech., Vol. 9, No. 2, Feb. 1991, pp. 271–283.

3. Y. Sun, J. L. Zyskind, and A. K. Srivastava, “Average Saturation Level, Modeling, and Physics of Erbium-Doped Fiber Amplifiers,” IEEE J. Selected Areas in Quantum Elect., Vol. 3, No. 4, Aug. 1997, pp. 991–1007.

4. Курков А. С., Наний О. Е. Эрбиевые волоконно-оптические усилители // Lightwave Russian Edition. 2003. № 1. С. 14–19.

Статья представлена научным руководителем, доцентом кафедры ФиЛС СПбГУТ, кандидатом технических наук Дюбовым А. С.

УДК 004.8
ГРНТИ 28.23.01

М. Бен Гедрия, И. А. Кальницкий, Е. В. Мартынова, А. А. Силина
(студенты гр. ИКПИ-15, СПбГУТ)

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ ДЛЯ ТРАНЗАКЦИЙ: ВАРИАНТЫ ПРИМЕНЕНИЯ В СОВРЕМЕННЫХ УСЛОВИЯХ

Искусственный интеллект произвел революцию во многих отраслях, включая финансовые операции. Применение искусственного интеллекта для транзакций может радикально изменить способ, которым мы осуществляем транзакции. В статье рассмотрены преимущества и примеры использования искусственного интеллекта для финансовых транзакций, а также исследованы данные опроса, предлагающего оценить полезность современных приложений и их преимущества, облегчающие финансовые операции.

искусственный интеллект, транзакции, финансовые операции, чат-бот.

Искусственный интеллект (ИИ) – это технология, которая может имитировать человеческое поведение для выполнения задач, обучаться и принимать решения, используя большие объемы данных. В последние годы ИИ стал чаще применяться в различных отраслях. Преимущества использования ИИ особенно заметны в сферах, где используется объем информации, который не способен систематизировать и проанализировать человек за разумный промежуток времени. Одной из областей, демонстрирующих наиболее широкое применение ИИ, является область финансов.

Как сообщила пресс-служба «Сбера» [1], финансовый эффект от внедрения искусственного интеллекта в 2022 году превысил 230 млрд рублей. При этом в 2019 году (рис. 1) ИИ обеспечил только 700 млн долларов (приблизительно 45,5 млрд рублей) дополнительной прибыли.

Исходя из динамики доходов крупнейшего банка РФ, которые обеспечил искусственный интеллект, следует, что технологии ИИ в финансовой сфере будут применяться все чаще.

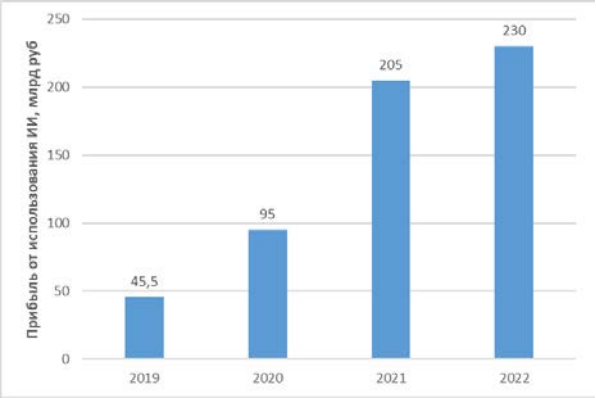


Рис. 1. Доходы «Сбера» от внедрения ИИ

Банковская индустрия использует искусственный интеллект в различных направлениях [2; 3]. Непосредственно для транзакций ИИ применяется в следующих процессах:

1. Обнаружение мошенничества в процессе транзакций. С помощью алгоритмов машинного обучения ИИ может анализировать большие объемы данных и выявлять аномальные транзакции, которые могут быть связаны с незаконными переводами денег. На основе сканирования обширных транзакционных данных ИИ может обнаружить операции, которые происходят в необычное время или с необычной суммой, и предупредить банк или платежную систему о возможном мошенничестве. Чем раньше ИИ сообщит об угрозе, тем быстрее банк ее устранит, сохранив доверие клиентов. Отслеживание нерегулярных моделей поведения наделяет искусственный интеллект уникальной способностью к обнаружению и минимизации риска банковского мошенничества.

2. Оптимизация процессов транзакций. Увеличение скорости обслуживания клиентов и сокращение времени обработки транзакций – одна из основных нужд в современных реалиях, когда все больше пользователей отдают предпочтение интернет-банкам. При помощи ИИ стало возможным осуществлять автоматическое распознавание платежей и определять категории транзакций, проверять подлинность платежей и принимать решения о необходимости дополнительной проверки, анализировать данные транзак-

ций и предоставлять рекомендации клиентам. Применение ИИ при перечисленных действиях улучшает эффективность платежных систем, сокращая время обслуживания.

3. Предсказание спроса и персональные предложения. Знание спроса потребителя помогает компаниям осуществлять управление ценами и стратегическое планирование. ИИ позволяет использовать данные о продажах, чтобы создать модель, которая будет автоматически предсказывать будущий спрос на товары и услуги, а также выявлять зависимость спроса от определённых факторов. Также ИИ может анализировать публикации из социальных сетей, выявляя тенденции в поведении потребителей и новейшие тренды. Обрабатывая данные из социальных сетей и информацию о транзакциях, ИИ получает «портрет» пользователя, создает персонализированные предложения, которые с большой вероятностью заинтересуют клиента.

4. Предоставление информации о транзакциях и общение с клиентами. Крупные банки внедряют чат-ботов, или виртуальных ассистентов, в свои мобильные приложения. Ассистенты отвечают на вопросы пользователей, предоставляют информацию о возможностях банка и осуществляют консультации – выполняют работу службы поддержки (рис. 2). Если бот не может понять вопрос клиента, он передает запрос сотруднику, однако все чаще ИИ самостоятельно справляется с этой задачей.

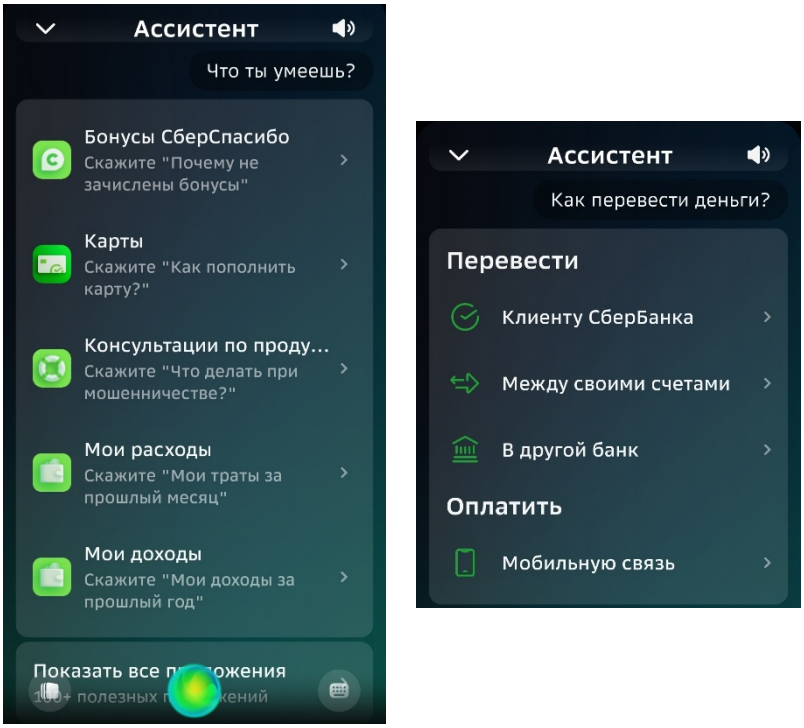


Рис. 2. Общение с чат-ботом на примере Ассистента «Сбера»

✓ Как видно по рис. 2, чат-бот на основе искусственного интеллекта понимает запросы пользователя и представляет решение задачи, предлагая

совершить действие (транзакцию) или перенаправляя на страницу с дополнительной информацией.

✓ Для определения общественного мнения о применении искусственного интеллекта для транзакций было проведено анкетирование (рис. 3) 53 человек возрастной группы 18–22 лет. В анкету включены вопросы о том, как респонденты воспринимают роль ИИ в банковском деле.

Рис. 3. Отношение пользователей к применению ИИ для транзакций

✓ Анализ результатов анкетирования показал, что большинство пользователей приложений банков положительно относятся к использованию ИИ для осуществления транзакций или получения информации о них. По дополнительным комментариям участников опроса составлены следующие выводы: многие пользователи доверяют ИИ участие в совершении операций, однако уверены, что технологии нуждаются в доработке.

Таким образом, искусственный интеллект представляет собой мощный инструмент для улучшения процессов транзакций и обеспечения безопасности платежей. Многие банковские операции проходят в электронной форме, для чего все чаще применяется ИИ, и в будущем его область применения будет становиться шире.

Список используемых источников

1. Искусственный интеллект помог Сбербанку заработать за год дополнительные $700 млн : Первый зампред правления банка Александр Ведяхин прогнозирует, что в 2020 году эта цифра достигнет миллиарда долларов [Электронный ресурс] // ТАСС. 2020. URL: https://tass.ru/ekonomika/7594027 (дата обращения 03.05.2023).

2. Искусственный интеллект в банках [Электронный ресурс] // tadviser. 2022. URL: https://www.tadviser.ru/index.php/Статья:Искусственный_интеллект_в_банках (дата обращения 03.05.2023).

3. 8 amazing ways consumers can benefit from the impact of artificial intelligence on the banking sector, financial sectors [Электронный ресурс] // Financial Express. 2017. URL: https://www.financialexpress.com (дата обращения 05.05.2023).

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 654.739

О. В. Бреднева, А. А. Земцов, Е. С. Пестряева, П. А. Ремденок
(студенты гр. ИКПИ-15, СПбГУТ)

ИССЛЕДОВАНИЕ ПРЕИМУЩЕСТВ ТЕХНОЛОГИИ THUNDERBOLT ПЕРЕД СУЩЕСТВУЮЩИМИ ТЕХНОЛОГИЯМИ ВЫСОКОСКОРОСТНОГО ПОДКЛЮЧЕНИЯ УСТРОЙСТВ

Технология Thunderbolt, разработанная совместными усилиями Intel и Apple, является новым стандартом для передачи данных и подключения периферийных устройств. Thunderbolt предоставляет уникальные возможности для пользователей, которые хотят иметь быстрый и удобный доступ к внешним устройствам. В данной статье проведено исследование преимуществ технологии Thunderbolt перед существующими технологиями подключения устройств.

Thunderbolt, интерфейс, устройство, порт, USB, PCI-e, DisplayPort, Type-C.

Современные компьютеры не могут существовать без эффективного и быстрого порта, который используется для подключения различных устройств. Thunderbolt - это один из наиболее передовых портов, который вырос из сотрудничества между Apple и Intel. Запущенный в 2011 году, Thunderbolt получил промышленное признание и нашел широкое применение в различных областях.

Thunderbolt – это высокоскоростной интерфейс, который соединяет периферийные устройства с компьютером через порт Thunderbolt, который объединяет в себе DisplayPort и PCI Express. Эта технология была выбрана, чтобы достигнуть наивысшей производительности периферийных устройств. Thunderbolt имеет скорость передачи данных до 40 Гбит/с, что делает его быстрее, чем другие технологии подключения устройств, такие как USB 3.0 и Lightning.

Порт Thunderbolt – это порт аудио-видео интерфейса, который использует высокоскоростную технологию передачи данных. Первоначально Thunderbolt был создан в качестве интерфейса для MacBook, но сегодня технология стала поддерживаться многими производителями компьютеров, включая Dell, Lenovo, HP и другие. Thunderbolt использует универсальный порт, который поддерживает передачу данных, электропитание, дисплей и другие функции.

Технология имеет ряд преимуществ, которые делают его портом будущего. Во-первых, Thunderbolt обеспечивает высокую скорость передачи данных, что позволяет быстро переносить большие файлы и использовать многозадачность. Во-вторых, предлагается универсальность, поскольку

порт может использоваться для нескольких функций. В-третьих, обеспечивает эффективное энергопотребление, что особенно важно для пользователей ноутбуков. В-четвертых, Thunderbolt поддерживает технологию DisplayPort, что позволяет передавать видео с разрешением 4K и выше.

Thunderbolt продолжает развиваться и имеет свое будущее. Интерфейс Thunderbolt 4, который появился в 2020 году, предлагает новые возможности, включая поддержку док-станций и расширительных коробок. Thunderbolt 4 поддерживает передачу данных с минимальной скоростью 40 Гбит/с, что делает его одним из самых быстрых портов, доступных на рынке. В будущем можно ожидать еще большего развития технологий Thunderbolt, включая поддержку беспроводной передачи данных и интеграцию в интернет вещей.

Thunderbolt, USB Type-C и Lightning имеют свои преимущества и недостатки. Порт Thunderbolt обеспечивает более высокую скорость передачи данных и поддерживает технологию DisplayPort, но имеет более высокую стоимость и является ограниченным в использовании только с определенными устройствами. USB Type-C, напротив, является универсальным и более доступным, но не поддерживает технологию DisplayPort без использования адаптеров. Lightning обеспечивает совместимость с устройствами Apple, но требует специальных кабелей и может иметь более медленную скорость передачи данных [1].

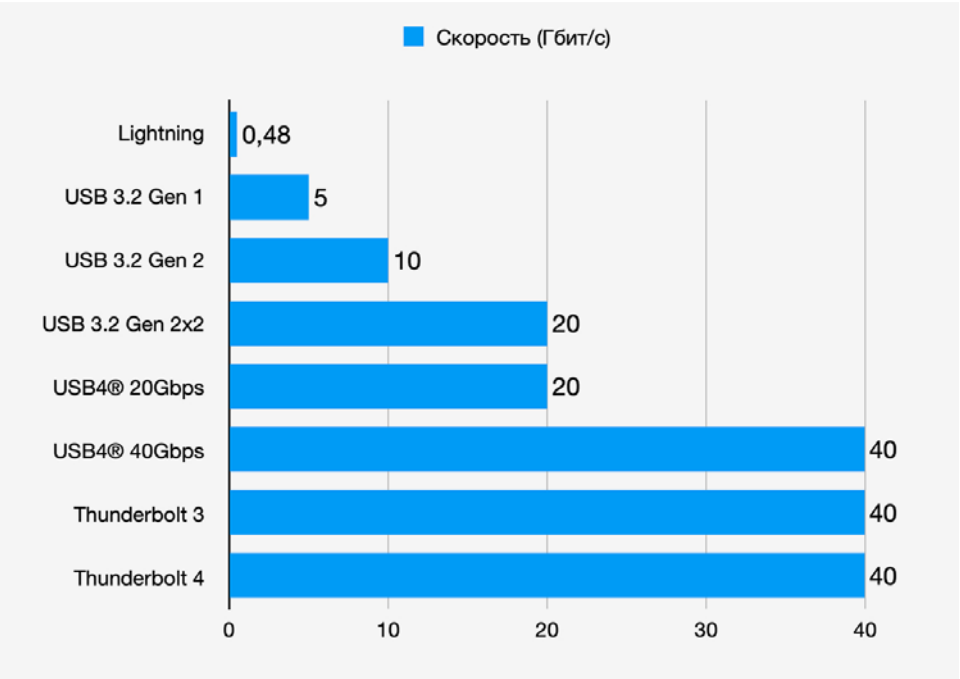


Рис. 1. Сравнение скорости передачи данных портов

Рассматриваемые порты (рис. 2) сильно отличаются с точки зрения распиновки и удобства в использовании, на данный момент Thunderbolt 3 и 4 используют разъем Type-C, который является наиболее удобным для конечного потребителя за счет симметричной распиновки, благодаря которой можно подключать устройство в любом положении.

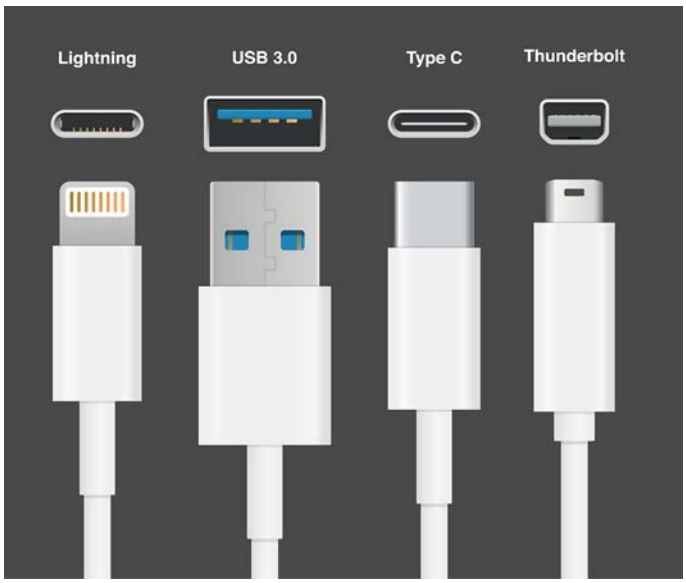


Рис. 2. Разъемы портов Lightning, Usb 3.0, Type C, Thunderbolt

Эти три технологии имеют свои преимущества и недостатки. При выборе порта для своих устройств, пользователи должны учитывать свои потребности, бюджет и совместимость с устройствами. В будущем можно ожидать еще большего развития этих технологий и внедрения новых функций и возможностей [2].

ТАБЛИЦА 1. Сравнение версий Thunderbolt 1, 2, 3 и 4

Особенности	Thunderbolt 1	Thunderbolt 2	Thunderbolt 3	Thunderbolt 4
Год запуска	2011	2013	2015	2020
Тип подключения	MDP (Mini DisplayPort)	MDP (Mini DisplayPort)	USB-C	USB-C
Скорость передачи данных, Гб/сек	20	20	до 40	40
Количество контактов	20	20	24	24
Протокол	4 x PCI-e 2.0 и DisplayPort 1.1a	4 x PCI-e 2.0 и DisplayPort 1.2	4 x PCI-e 3.0, DisplayPort 1.2, HDMI-2 и USB 3.1 Gen-2	4 x PCI-e 3.0, DisplayPort 2.0 и USB4

Главным отличием Thunderbolt 4 от Thunderbolt 3 является увеличение пропускной способности при работе по PCI-e в два раза (32 Гб/сек) и возможность подключения одного 8К монитора или двух 4К мониторов, в свою очередь Thunderbolt 3 позволяет подключить лишь один 4К монитор [3, 4].

Список используемых источников

1. Thunderbolt vs. USB-C: What's the Difference? URL: https://www.pcmag.com/news/thunderbolt-3-vs-usb-c-whats-the-difference
2.Thunderbolt 3 vs USB C: Understanding the Difference and What it Means URL: https://www.usbmemorydirect.com/blog/thunderbolt-3-vs-us-c/
3. Difference Between 4 Types of Thunderbolt URL: https://www.etechno-philes.com/types-of-thunderbolt-differences/
4. Что нового в Thunderbolt 4? URL: https://rozetked.me/articles/12250-chto-novogo-v-thunderbolt-4 // Technology@Intel. 2004. № 8. С. 27–29.

Статья представлена научным руководителем, доцентом кафедры ПИВТ СПбГУТ, кандидатом технических наук Бирюковым М. А.

УДК 004.356.2

А. С. Брусиловский, К. Д. Волосников, Г. С. Добродеев, Н. Д. Никель, А. Д. Щербаков, (студенты гр. ИКПИ-11, СПбГУТ)

ОСОБЕННОСТИ 3D ПРИНТЕРОВ И ФОРМИРОВАНИЯ МОДЕЛЕЙ ДЛЯ НИХ

Целью данной работы является исследование перспектив развития, сравнение различных технологий и материалов, используемых при 3Д-печати, анализ данных на основе различных характеристик, и создание модели фиксатора конечностей из PLA пластика.

3Д-принтер, PLA, FDM, SLA, пластик.

В настоящее время активно развивается сфера 3Д-печати. Ученые создают новые материалы, которые могут быть использованы в любой сфере: строительство, электроника, тяжелая промышленность, фармацевтическая промышленность.

3Д-печать представляет собой проектирование трехмерных моделей заданной геометрической формы. Процесс получения макета основывается на поэтапном построении объекта с помощью нанесения слоев с использованием разных материалов [1–3].

Устройство принтера

Конструкция SLA

В данном виде 3Д-принтера есть лазер, который через специальное стекло послойно «засвечивает» на поверхности площадки, которая погружена в жидкий полимер, слои (рис. 1). Таким образом, создается идеальное

очертание каждой ступени. Данный вид принтеров работает с помощью ультрафиолетового излучения. Также имеется кожух, сделанный из органического стекла, который не позволяет излучению проходить наружу.

Здесь только один подвижный элемент – столик, который нужен для перемещения по оси Y, а за остальные оси отвечает УФ-лазер.

Рис. 1. SLA-принтер

Конструкция FDM

В отличие от модели SLA, в FDM есть моторы, зубчатые ремни и оси, с их помощью происходит перемещение печатающего узла (рис. 2). Экструдер – печатающий элемент. Он затягивает в себя пластиковую нить, затем с помощью нагревающего элемента плавит его. После этого из сопла (отверстие для выдачи пластика) выдавливается жидкий материал.

Рис. 2. FDM-принтер

Принципы работы

Моделирование методом наплавления (FDM)

3D-принтер представляет собой аппарат, способный осуществлять перемещение по осям X, Y и Z. Поэтому они оснащены ступенчатыми двигателями. Эти механизмы управляются контроллером и способны перемещать печатную головку, которая выталкивает расплавленный полимер, создавая деталь послойно. Для более точного позиционирования принтеры используют специальные зубчатые ремни и ролики вдоль осей X и Y, а также винты с резьбой или специальные шпиндели вдоль оси Z.

FDM – самый распространенный в мире вид 3Д-печати, принцип работы которого заключается в «послойном» наплавлении нитей из расплавленного пластика при создании модели. Печатающий элемент перемещается по 3 осям, пластик продавливается через сопло, где он плавится и выдавливается слоями на стол 3Д-принтера.

Стереолитография

Данный вид печати использует свет для того, чтобы в специальной емкости с фотополимерной смолой, «вырастить» модель. Здесь, объект тоже появляется слой за слоем, но после высыхания каждого слоя фотополимера.

Главное отличие от FDM печати заключается в том, что в стереолитографии созданная модель и грани получаются более гладкими и аккуратные.

Перспективы развития

3Д-принтер сильно изменил мир, после своего появления. Стало возможным создавать любые модели самых различных форм, при этом не используя вспомогательные инструментов для литья. Относительная доступность данной технологии, стала привлекать ученых для дальнейшего её развития, создания новых материалов, обеспечивающих более эффективные показатели, а также совершение научных открытий во многих сферах жизни.

Таким образом, уже в скором будущем, человечеству откроется возможность полного воспроизведения органов с использованием синтетических материалов, которые будут способны полностью дублировать функции настоящих человеческих органов. Благодаря этому, намного больше людей сможет получить лечение.

Также, развитие 3Д-принтеров позволит проектировать любые вещи, которые понадобятся человеку, так как есть возможность их проектировать по своим требованиям и нуждам.

Анализ данных

Также, в ходе работы были применены навыки анализа данных с помощью языка программирования Python 3.8 и подключаемых библиотек: pandas, seaborn, matplotlib.

Используя базу данных с различными параметрами печати, такими, как высота слоя (мм), толщина стенки (мм), скорость печати (мм/с), материал (ABS/PLA), были получены следующие зависимости.

На «тепловой карте» (рис. 3) можно увидеть, что скорость печати зависит от толщины стенки. Чем тоньше слой, тем быстрее принтер печатает (20–30), и чем толще, тем больше времени тратится на печать макета (0–2,6–10).

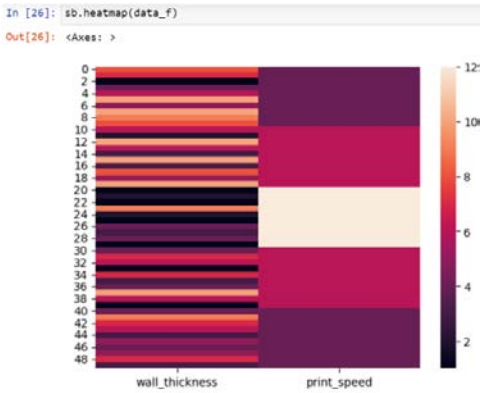


Рис. 3. Распределение скорости печати и толщины стенки

На следующем графике (рис. 4) можно увидеть, что скорость печати одинаковая и зависит не от выбранного материала (ABS/PLA), а от иных характеристик.

Здесь мы видим, что при минимальной толщине стенки (20–30), мы получаем

среднее значение высоты (рис. 5). При максимальной толщине стенки – минимальная высота (0–8), и при значении толщины, чуть меньшем среднего, – максимальная высота (40–48).

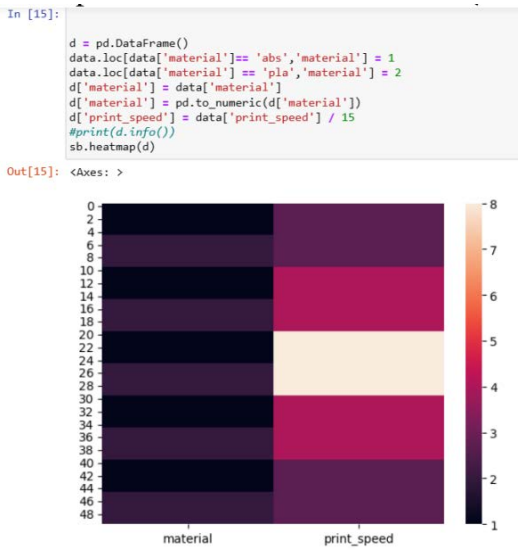


Рис. 4. Распределение скорости печати и материала

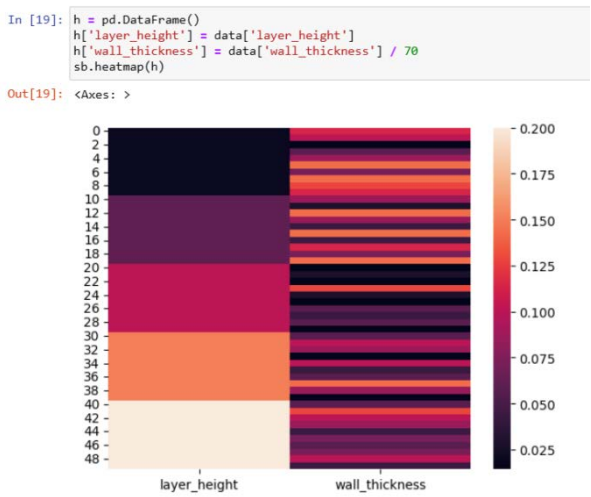


Рис. 5. Распределение высоты стенки и толщины стенки

Разработка модели

Учитывая различные характеристики печати, были выбраны следующие материалы и форм фактор:

«Соты» – данная форма была выбрана из-за следующих свойств: сохранение подвижности, легкая адаптация к форме конечности, наличие отверстий, которые способствуют циркуляции воздуха и обеспечивают постоянным контролем избыточного тепла между протезом и конечностью (рис. 6).

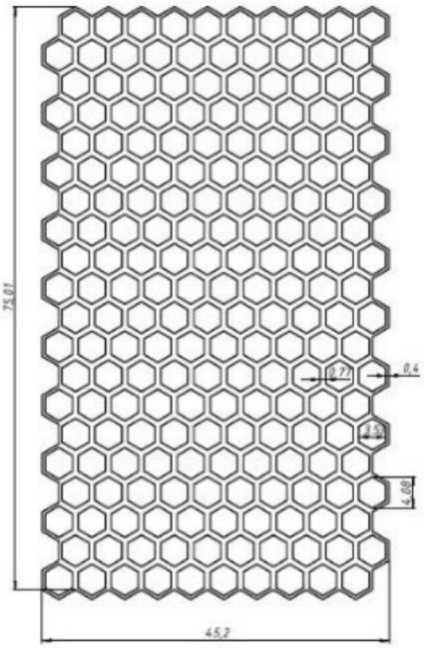


Рис. 6. «Соты»

PLA – материал, обеспечивающий высокое качество поверхности, достаточно прочный и гибкий, и ввиду своей биосовместимости, полиактид широко применяется в медицине и протезировании.

В ходе работы, с использованием компьютерной программы Blender, был разработан макет фиксатора. Данная модель печатается из PLA пластика, который становится мягким при температуре (90–110) градусов и легко принимает необходимую форму. Материал быстро остывает, поэтому паци-

ент не получит ожогов. При использовании созданной модели можно зафиксировать часть тела или конечность, защитить от внешнего воздействия поврежденную область.

Модели были распечатаны на принтере, используя технологию FDM.

Вывод

В ходе работы было рассказано об основных принципах 3Д-печати, был произведен анализ данных печати при различных параметрах (материал, толщина стенки, высота стенки, скорость печати), была разработана модель фиксатора конечностей из PLA пластика, отвечающая всем поставленным в техническом задании параметрам и особенностям.

Список используемых источников

1. https://losprinters.ru/articles/kak-ustroen-3d-printer/
2. https://www.cdwg.com/content/cdwg/en/articles/hardware/how-does-a-3d-printer-work.html#:~:text=It%20works%20by%20exposing%20a,depends%20upon%20the%20printer%20type
3. https://www.kaggle.com/datasets/afumetto/3dprinter?resource=download&select=data.csv

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.056.5

Е. С. Бугрова (студент гр. ИКБ-95, СПбГУТ)
Е. О. Романюк (студент гр. ИКТБ-27м, СПбГУТ)
В. А. Шестакова (студент гр. ИКТС-23м, СПбГУТ)

ОТСЛЕЖИВАНИЕ АТАКИ ТИПА ОТКАЗ В ОБСЛУЖИВАНИИ И ИХ СИМУЛЯЦИЯ НА ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ OPENSTACK С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ МОНИТОРИНГА GRAFANA

В данной научной статье поднимается актуальная проблема, связанная с обнаружением и моделированием атак типа отказ в обслуживании на облачной инфраструктуре OpenStack. В работе рассматриваются различные подходы к симуляции таких атак и методы для мониторинга и отслеживания процесса атаки с использованием системы мониторинга Grafana. Кроме того, выполняется тестирование эффективности настроенной системы. Полученные результаты исследования могут быть применимы при проектировании и разработке облачных инфраструктур, а также при выборе оптимальных подходов к защите от атак типа отказ в обслуживании.

система мониторинга, облачная инфраструктура, обнаружение атак, информационная безопасность.

В современных условиях мониторинг является одной из важнейших задач в области IT-технологий. В частности, при работе с виртуализированными окружениями необходимо иметь возможность мониторинга ресурсов, чтобы обеспечить высокую производительность и надежность системы.

В данной статье будет описан процесс создания и настройки мониторинговой системы для трех виртуальных машин в OpenStack с использованием Prometheus, Node Exporter и Grafana. Также будет рассмотрен процесс тестирования системы на реакцию на эмулирование DDoS атаки.

OpenStack [1] – это программное обеспечение с открытым исходным кодом, которое предоставляет инфраструктуру облачных вычислений [2].

Использование OpenStack обеспечивает ряд преимуществ, включая:

– гибкость и масштабируемость: OpenStack позволяет пользователям масштабировать свои ресурсы в соответствии с изменяющимися потребностями, что обеспечивает гибкость и масштабируемость облачной инфраструктуры;

– открытый исходный код: OpenStack является проектом с открытым исходным кодом, что позволяет пользователям настраивать и модифицировать программное обеспечение в соответствии с их потребностями;

– совместимость: OpenStack является совместимым с различными типами оборудования и ПО, что позволяет пользователям создавать гибкие и настраиваемые облачные решения;

– безопасность: OpenStack обеспечивает высокий уровень безопасности для облачных вычислений, что позволяет пользователям быть уверенными в защите своих данных и ресурсов.

Таким образом, использование OpenStack является эффективным способом для создания и управления облачной инфраструктурой с учетом потребностей и требований пользователя.

Мониторинг является важным компонентом безопасности системы OpenStack. Он позволяет оперативно обнаруживать и предотвращать атаки, в том числе DDoS, снижает вероятность сбоев и утечек данных, и позволяет оперативно реагировать на любые изменения в системе, которые могут негативно сказаться на безопасности.

Одним из эффективных инструментов мониторинга является Grafana [3], который позволяет отслеживать различные метрики и анализировать их для выявления подозрительной активности.

Облачные инфраструктуры [4] представляют собой распределенные вычислительные среды, которые позволяют предоставлять вычислительные ресурсы в виде сервиса через сеть. Типовая модель облачной инфраструк-

туры включает в себя различные компоненты, каждый из которых выполняет свою функцию и совместно обеспечивает работу облачной инфраструктуры.

Controller представляет собой центральный узел управления, который включает в себя компоненты, таких как Keystone, Nova, Glance, Horizon, Cinder и Neutron. Keystone является идентификационным сервисом, который позволяет пользователям и приложениям получать доступ к облачным ресурсам. Nova – это сервис управления виртуализацией, который управляет жизненным циклом виртуальных машин и запускает их на вычислительных узлах. Glance – это сервис хранения образов виртуальных машин, который позволяет быстро создавать и запускать новые виртуальные машины. Horizon – это веб-интерфейс управления, который предоставляет возможность администрирования облачной инфраструктуры. Cinder – это сервис хранения блочных данных, который обеспечивает доступ к блочным устройствам для виртуальных машин. Neutron – это сервис управления сетью, который обеспечивает возможность создания и управления виртуальными сетями в облачной инфраструктуре.

Compute01 и Compute02 представляют собой вычислительные узлы, на которых запускаются виртуальные машины. Они включают в себя сервис управления виртуализацией Nova и сервис управления сетью Neutron, который обеспечивает доступ к виртуальным сетям и маршрутизации трафика (рис. 1).

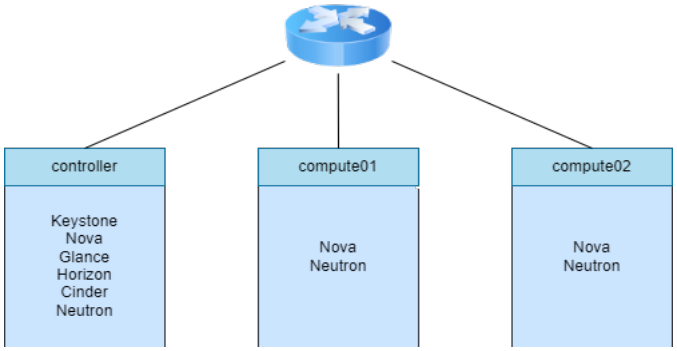


Рис. 1. Типовая модель облачной инфраструктуры

Для мониторинга ресурсов виртуальных машин необходимо использовать инструменты, которые обеспечивают надежный сбор, хранение и визуализацию данных.

Prometheus [5] – это система мониторинга с открытым исходным кодом, которая предназначена для сбора, хранения и анализа метрик различных компонентов в среде микросервисов.

Node Exporter [6] – это программный продукт для сбора метрик операционной системы хоста, который позволяет использовать эти метрики в Prometheus для мониторинга системы.

Prometheus является одним из наиболее распространенных инструментов, который обеспечивает сбор, хранение и анализ метрик. Node-exporter, в свою очередь, является агентом сбора метрик, который собирает информацию о состоянии узлов в Openstack и отправляет ее в Prometheus. Комбинация Prometeus и Node-exporter обеспечивает полный и точный мониторинг состояния инфраструктуры Openstack, что позволяет операторам быстро выявлять и решать проблемы и снижать риски, связанные с безопасностью.

Для настройки мониторинга необходимо установить Prometheus, Node Exporter и Grafana [7] на первую виртуальную машину в OpenStack. После установки необходимо настроить Prometheus для сбора метрик с Node Exporter на второй и третьей виртуальных машинах (рис. 2).

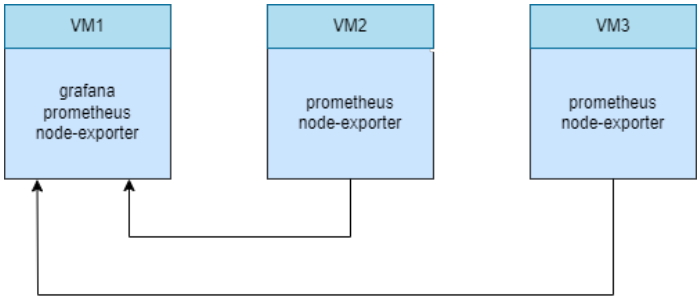


Рис. 2. Типовая модель системы мониторинга

После настройки сбора метрик необходимо настроить панель мониторинга в Grafana для отображения метрик. На панели можно отслеживать такие параметры, как трафик, CPU и RAM для каждой виртуальной машины (рис. 3).

Рис. 3. Подготовленная панель Grafana

В качестве типовой атаки на информационную систему выбрана атака типа отказ в обслуживании. Выбор данного типа атаки обусловлен его наглядностью и относительной легкостью реализации.

После настройки мониторинга, был написан скрипт для эмулирования DDoS-атаки [8] на одну из виртуальных машин.

DDoS-атака (англ. *Distributed Denial of Service* – «распределённая атака на отказ») – это один из наиболее распространенных способов атак на веб-ресурсы, который заключается в создании большой нагрузки на серверы или сеть, в результате чего они перестают отвечать на запросы пользователей. DDoS-атака может привести к серьезным последствиям, таким как отключение веб-сайта, кража данных или даже потеря контроля над устройством.

Подготовка к эксперименту проводилась с целью изучения поведения системы в условиях DDoS-атаки [9] и оценки ее устойчивости к нагрузкам. Для этого были использованы три виртуальные машины, развернутые на платформе OpenStack под управлением операционной системы Ubuntu.

На первой виртуальной машине были установлены Grafana, Prometheus, Node-exporter и скрипт для генерации трафика. На второй виртуальной машине были установлены Prometheus, Node-exporter и тот же скрипт для генерации трафика. Третья машина служила в качестве целевой, на которую направлялась DDoS-атака.

Скрипт был написан на языке Python [10] и включал в себя механизм генерации случайных пакетов. В рамках эксперимента было направлено два потока сетевого трафика на порт 22. Каждый поток содержал 500 параллельных соединений, которые передавали UDP пакеты. Общее количество переданных пакетов составило 500000.

В рамках подготовки к эксперименту был запущен скрипт на первой и второй виртуальной машинах, который генерировал udp-трафик на третью виртуальную машину, тем самым создавая нагрузку на систему.

На панели мониторинга Grafana был отображен результат подготовки к эксперименту. Визуализация данных была выполнена с помощью Prometheus, который собирал метрики с Node-exporter на каждой машине. Эти метрики показывали количество пакетов, отправленных на целевую машину, а также другие параметры, такие как загрузка процессора и объем использованной памяти (рис. 5, см. ниже).

В ходе подготовки к эксперименту была подготовлена инфраструктура, состоящая из трех виртуальных машин, где на первых двух запускались скрипты, генерирующие трафик на третью машину, что приводило к ее нагрузке. Для мониторинга состояния системы были установлены Prometheus, Grafana и Node-exporter.

Благодаря проведенному эксперименту была создана основа для дальнейших исследований и экспериментов. Была установлена необходимая инфраструктура и инструменты для мониторинга состояния системы, что позволит проводить дальнейшие эксперименты с высокой точностью

и надежностью. В результате проведенного эксперимента можно увидеть, что система мониторинга корректно отображает изменения в работе системы при запуске скрипта, что дает возможность проводить более глубокие исследования и эксперименты в будущем.

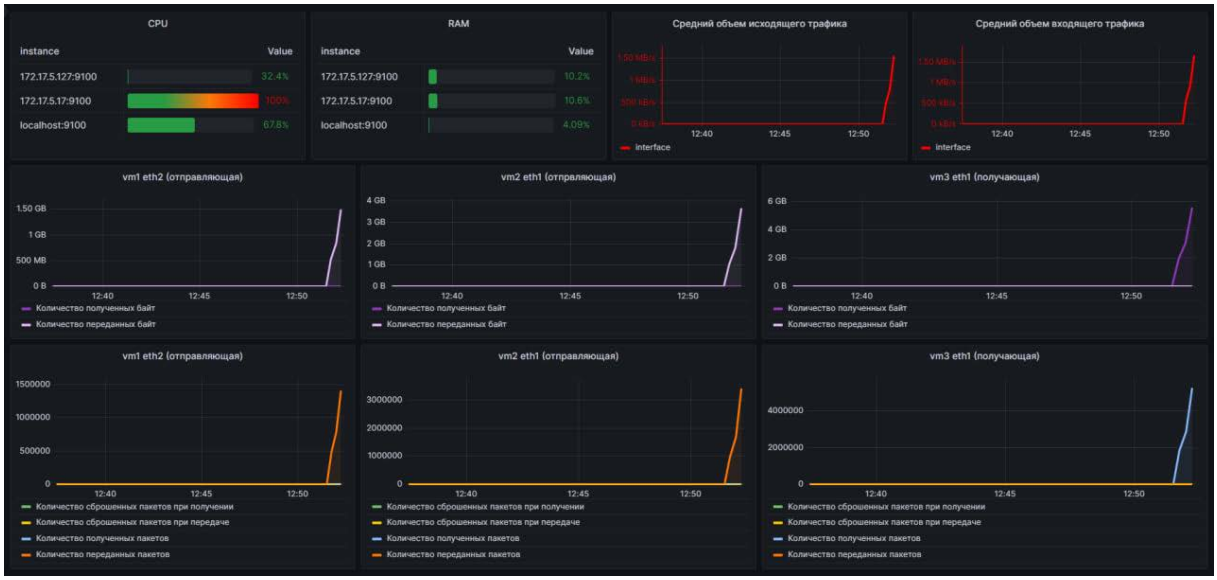


Рис. 5. Результат запуска скрипта на панели мониторинга

Таким образом, можно с уверенностью сказать, что подготовленная инфраструктура и установленные инструменты позволят проводить успешные исследования и эксперименты в области облачных вычислений, а результаты, полученные в ходе эксперимента, будут успешными и обнадеживающими для дальнейшей работы в этой области.

Список используемых источников

1. Openstack. URL: https://www.openstack.org/ (дата обращения 19.04.2023).
2. Алехин Р. В., Красов А. В., Макарова А. Д., Орлов Г. А. Облачные сервисы. Принцип работы, классификация и модели обслуживания // Актуальные проблемы инфотелекоммуникаций в науке и образовании. XI международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб. : СПбГУТ, 2022. Т. 1. С. 70–74.
3. Дедухова А. А., Швидкий А. А. Сравнительная оценка систем мониторинга компонентов облачной инфраструктуры // Актуальные проблемы инфотелекоммуникаций в науке и образовании. X юбилейная международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб. : СПбГУТ, 2021. Т. 1. С. 292–296.
4. Гельфанд А. М., Косов Н. А., Красов А. В., Орлов Г. А. Защита для распределенных отказов в обслуживании в облачных вычислениях // Актуальные проблемы инфотелекоммуникаций в науке и образовании. VIII международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб. : СПбГУТ, 2019. Т. 1. С. 329–334.
5. Prometheus. URL: https://prometheus.io/ (дата обращения 19.04.2023).
6. Node Exporter. URL: https://github.com/prometheus/node_exporter (дата обращения 19.04.2023).

7. Grafana. URL: https://grafana.com/ (дата обращения 19.04.2023).
8. Гришин Н. А., Косов Н. А., Мазепин П. С. Актуальность методов борьбы с DDoS атаками // Актуальные проблемы инфотелекоммуникаций в науке и образовании. X Юбилейная международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб. : СПбГУТ, 2021. Т. 2. С. 176–179.
9. Хабр. URL: https://habr.com/ru/companies/southbridge/articles/674218/ (дата обращения 19.04.2023).
10. Python. URL: https://www.python.org/ (дата обращения 19.04.2023).

Статья предоставлена заведующим кафедрой ЗСС СПбГУТ,
кандидатом технических наук, доцентом Красовым А. В.

УДК 654.739

М. М. Важник, В. Д. Галкина, В. Е. Райхерт, Е. В. Рыбкина,
Д. И. Хмиль (студенты гр. ИКПИ-14, СПбГУТ)

АССЕМБЛЕР: «ЗА» или «ПРОТИВ»

Ассемблер является низкоуровневым языком программирования, который тесно связан с аппаратным обеспечением компьютера. В данной статье рассматривается его роль и применение, а также анализируются преимущества и недостатки этого языка. Обсуждается устаревание ассемблера и возможные альтернативы, а также оценивается ценность изучения ассемблера в современном информационном мире.

ассемблер, низкоуровневый язык, системное программирование, криптография, компьютерная архитектура, микроконтроллеры, работа с памятью, бытовые поля.

Ассемблер – это низкоуровневый язык программирования, который используется для написания программ, более непосредственно взаимодействующих с аппаратным обеспечением компьютера. Он представляет собой мнемоническое представление машинных инструкций и адресов памяти, что позволяет программисту более точно контролировать выполнение кода и использование ресурсов [1–4].

Ассемблер находит применение в различных областях, требующих низкоуровневого программирования и прямого взаимодействия с аппаратурой. Например, он широко используется в разработке операционных систем, драйверов устройств, встроенных систем, микроконтроллеров и других критически важных систем, где требуется максимальная производительность и контроль над ресурсами.

Ассемблер имеет несколько значительных преимуществ, почему его считают вечной основой программирования:

– Прямой доступ к аппаратным ресурсам: Ассемблер позволяет программисту более тесно контролировать аппаратное обеспечение компьютера, что важно для оптимизации производительности и разработки систем с жесткими требованиями к ресурсам.

– Максимальная эффективность: поскольку ассемблер генерирует машинный код, программы, написанные на нем, работают очень быстро и эффективно.

– Гибкость и низкоуровневое программирование: Ассемблер позволяет точно манипулировать каждой инструкцией и каждым байтом памяти, что полезно в случаях, когда требуется прямое взаимодействие с аппаратным обеспечением.

Несмотря на свои преимущества, ассемблер также имеет некоторые недостатки:

– Сложность и трудоемкость разработки: Ассемблер является низкоуровневым языком, требующим от программиста глубоких знаний архитектуры процессора и инструкций, а также тщательного контроля каждой детали программы.

– Портабельность и переносимость: из-за прямой привязки к аппаратному обеспечению, программы на ассемблере часто зависят от конкретной архитектуры процессора и не могут быть просто перенесены на другую платформу.

– Низкая абстракция: поскольку ассемблер представляет машинные инструкции, код на нем обычно трудно читать и поддерживать, что может затруднить сопровождение и разработку сложных программных систем.

Некоторые люди могут считать ассемблер устаревшим из-за его сложности и специфичности. В современном программировании существуют более высокоуровневые языки, которые предоставляют более удобные средства для разработки и поддержки программных проектов.

Существуют альтернативы ассемблеру, такие как языки высокого уровня (например, C, C++, Python), которые предоставляют более высокий уровень абстракции и сокращают время разработки. Однако, несмотря на это, ассемблер все еще широко используется в некоторых областях, требующих максимальной производительности и низкоуровневого контроля.

Изучение ассемблера может быть полезным для программистов, занимающихся разработкой низкоуровневых систем и оптимизацией производительности. Понимание ассемблера позволяет лучше понять работу аппаратного обеспечения и принципы выполнения программ на более высокоуровневых языках. Это может быть особенно полезно в областях, таких как разработка операционных систем, драйверов устройств, встроенных систем и безопасности.

По нашему мнению, ассемблер имеет свое место в программировании, особенно в контексте низкоуровневого разработки и оптимизации произво-

дительности. Хотя существуют более удобные и выразительные языки программирования, ассемблер обеспечивает уникальную возможность полного контроля над аппаратным обеспечением и достижения максимальной эффективности. Кроме того, понимание ассемблера помогает программисту лучше понять внутреннее устройство компьютерных систем и улучшает его способность анализировать и исправлять проблемы на низком уровне.

Однако не стоит забывать, что изучение ассемблера требует времени и усилий, и его применение может быть ограничено определенными областями программирования. Поэтому решение о том, стоит ли учить ассемблер, должно быть основано на конкретных целях и потребностях программиста.

В ходе исследования было опрошено 18 человек среди факультетов ИКСС, ИСиТ, РТС.

Из них 11 человек обучаются на втором курсе, 4 человека на 1, 2 человека на 3, и 1 опрошенный на 4.

Подавляющее большинство знает, что такое ассемблер и язык ассемблера (16 человек).

Из плюсов языка ассемблера обучающиеся выявили несколько основных пунктов:

– работа напрямую с «железом»;
– скорость работы.

+ В этом ЯП предоставлен полный (насколько я знаю) контроль над распределением памяти => более "гибкая" разработка (чего-либо) и возможность ускорить работу программы, по максимуму задействовав возможности процессора
+ Даёт отличное понимание всего базового программирования в целом (на низком уровне)

Возможно, его изучение помогает хорошо разобраться с устройством памяти компьютера и тем, как реализуются базовые операции (мб для какой-то специальности такие знания пригодятся)

Прямая работа с железом

Скорость работы программ, возможность работы с железом

Высокая производительность

Напрямую работа с железом

Из минусов ассемблера также обозначено несколько пунктов:

– сложный синтаксис;
– мало информации для изучения;
– разные устройства – разный результат.

непостоянство использования, невозможность получить одинаковый результат на разных устройствах

- Не интуитивно-понятный по сравнению с другими языками программирования
- Для разной архитектуры процессора используется свой язык ассемблера (насколько я знаю) => может усложнять жизнь разработчику (как минимум из-за разного синтаксиса и набора регистров)

Слишком сложно разобраться в нем, необходимо знать много, чтобы реализовать даже простую программу.

Громоздскость

Сложный синтаксис, малое количество информации для изучения

Сложный синтаксис, понимание языка

Сложность в освоении

Высокая сложность написания программ, крайне низкая скорость написания кода, нехватка качественных туториалов.

Тяжёлый синтаксис

Каждое железо - свой подход

Ответы на основной вопрос, ассемблер – вечная основа или устаревший подход, разделились примерно поровну. Некоторые студенты дали неоднозначные ответы.

основа, так как позволяет понять работу с памятью и в целом изучить работу с компьютером на низком уровне

Вечная основа

* Однако я очень плохо знаю область применения ассемблера, чтобы судить об этом, но мне кажется, что это даже не основа, а просто незаменимая штука.
Скорее всего это ненадолго (т.к. технологии и ЯП развиваются) и далеко не везде ассемблер незаменим, но я думаю, что это основа :D

Я не сильно разбираюсь в языках низкого уровня, но считаю, что их изучение в любом случае важно. Какой именно язык изучается, наверное, не играет большой роли, главное помогает ли это прояснить для себя более сложную организацию компьютерных программ. Поэтому мой ответ, что это скорее вечная основа.

Мне кажется, что ассемблер это не вечная основа, но в данный момент это единственный способ решения многих практических задач, с которыми он отлично справляется

Устаревший подход

Устаревший подход. Сложно учить. Мало кто захочет тратить на это время, да и в наше время многие низкоуровневые вещи можно написать на Си

Основа, для студентов важен для понимания программирования, но мазохистов пишущих на нем серьёзные проекты немного

Устаревший подход. Необходимо много знаний для написания простой программы. Большой листинг(нечитабельно).

устаревший подход

это и то и другое, потому что не стоит делить мир на чёрное и белое.

Каждый язык программирования в основном создан для решения определённого класса задач. Поэтому, я бы ответил на ваш вопрос так: есть подходящие для этого языка задачи и есть не подходящие

Не имею достаточно знаний для ответа, но считаю, что Ассемблер все еще нужен

Список используемых источников

1. Столяров А. В. Программирование, введение в профессию. Т. 2. Низкоуровневое программирование. М. : МАКС Пресс, 2016. 496 с. ISBN 978-5-317-05301-7.

2. Секаев В. Г. Основы программирования на Ассемблере : учеб. пособие. Новосибирск : Изд-во НГТУ, 2010. 100 с. ISBN 978-5-7782-1473-6.

3. Юров В. И. Assembler : учебник для вузов. СПб. : Питер, 2003. 637 с.: ил. ISBN 5-94723-581-1.

4. Зубков С. В. Assembler – язык неограниченных возможностей. ДМК Пресс, 1999.

Статья представлена научным руководителем,
ассистентом кафедры ПИВТ СПбГУТ Степаненковым Г. В.

УДК 621.39

Н. С. Васильев, В. П. Клименко (студенты гр. ИКФ-11, СПбГУТ)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АМПЛИТУДНОЙ И ФАЗОВОЙ БИНАРНОЙ МОДУЛЯЦИИ ЦИФРОВЫХ ОПТИЧЕСКИХ СИГНАЛОВ

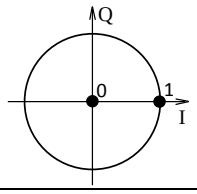
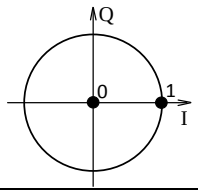
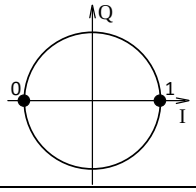
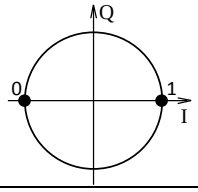
Работа посвящена сравнительному анализу двух методов бинарной модуляции цифровых оптических сигналов для волоконно-оптических систем связи (ВОСС). Это амплитудная (АМ) и фазовая (ФМ) модуляции с NRZ и RZ кодированием. Проведено имитационное моделирование и сравнение качества связи.

волоконно-оптическая система связи, Q-фактор, модуляция, кодирование, одномодовое оптическое волокно, хроматическая дисперсия.

Амплитудно-фазовые векторные диаграммы для АМ и ФМ

Для наглядного представления дискретных двухуровневых и многоуровневых форматов модуляции при определенном состоянии линейной поляризации используют амплитудно-фазовые векторные диаграммы (АФВД), на которых ось *I* (вещественная) соответствует фазовому сдвигу 0, а ось *Q* (мнимая) – сдвигу π/2 (таблица 1). Длина вектора определяет амплитуду напряженности Em электрического поля в световой волне, а угол поворота – его фазу φ. Часто сами вектора E не показывают, а оставляют только точки, соответствующие вершинам векторов [1].

ТАБЛИЦА 1. АФВД для бинарной АМ и ФД
для двух ортогональных состояний поляризации

№	Формат модуляции	Фазовые диаграммы для ориентации вектора *E* вдоль осей	
		x	*y*
1	Амплитудная бинарная модуляция ASK (*Amplitude Shift Keying*) или OOK (*On Off Keyng*)		
2	Дифференциальная бинарная фазовая модуляция DPSK (*Diferential Phase Keying*)		

Расположение концов векторов *E* на фазовой плоскости можно зарегистрировать экспериментально в виде сигнальных созвездий, точки н которых вырождаются в области неопределенности, размеры которых определяются аддитивными шумами. Каждый из рассмотренных видов модуляции может иметь формат линейного кодирования NRZ (без возврата к нулю) или RZ (с возвратом к нулю).

Для увеличения в два раза пропускной способности линейного тракта можно одновременно передавать по ОВ два независимых высокоскоростных оптических сигнала с ортогональными поляризациями. Например, при горизонтальной (*x*) и вертикальной (*y*) поляризациях.

Отметим, что для рассмотренных АФВД определяющим является взаимное расположение точек на окружностях, а положение осей *I* и *Q* не меняет формата модуляции.

Описание схемы исследования

Схема исследования на основе моделирующей программы (*OptiSystem*) [2] приведена на рис. 1. Верхняя схема реализует модуляцию DPSK, а нижняя ASK. Обе схемы содержат оптические передатчики со скоростью передачи *B* = 10 Гбит/с, одинаковые линейные тракты, состоящие из 100 км стандартного оптического волокна (ОВ1) и 10 км компенсирующего ОВ2, полностью компенсирующего хроматическую дисперсию (ХД).

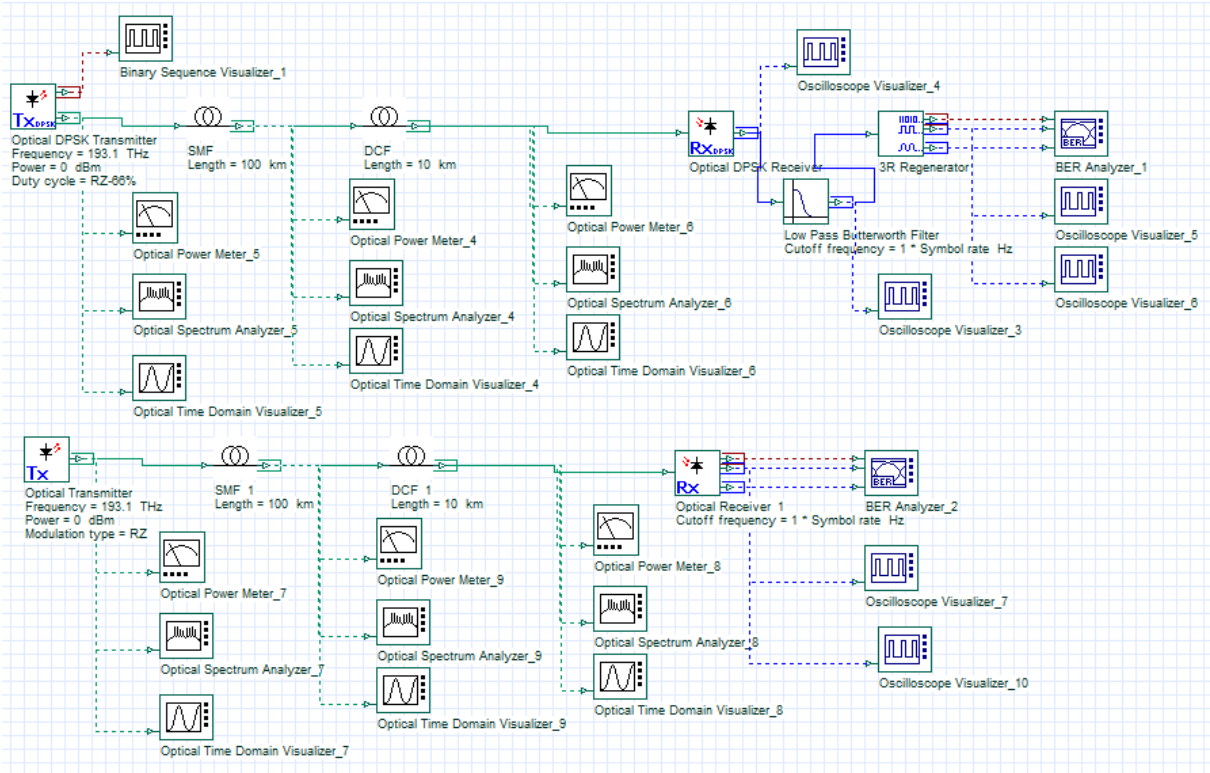


Рис. 1. Схемы ВОСС с бинарными АМ и ФМ и энергетическим приемом

Для приема сигналов DPSK используют схему с балансным ФП (рис. 2), состоящим из двух фотодиодов ФД1 и ФД2, которые могут представлять собой p-i-n ФД или ЛФД. Входной оптический сигнал поступает на один вход интерферометра Маха-Цендера (ИМЦ), состоящего из двух направленных ответвителей (НО). В одно плечо ИМЦ помещают устройство временной задержки оптического сигнала (Т) на один такт битовой последовательности $\Delta T = 1/B$. Напряженности электрических полей задержанного и не задержанного сигналов складываются (интерферируют), снова разделяются и подаются на входы балансного ФП. Схема рис. 2 является оптическим демодулятором, преобразующим сигнал DPSK в два сигнала с АМ. Она входит в состав приемника Rx DPSK, показанного на рис 1.

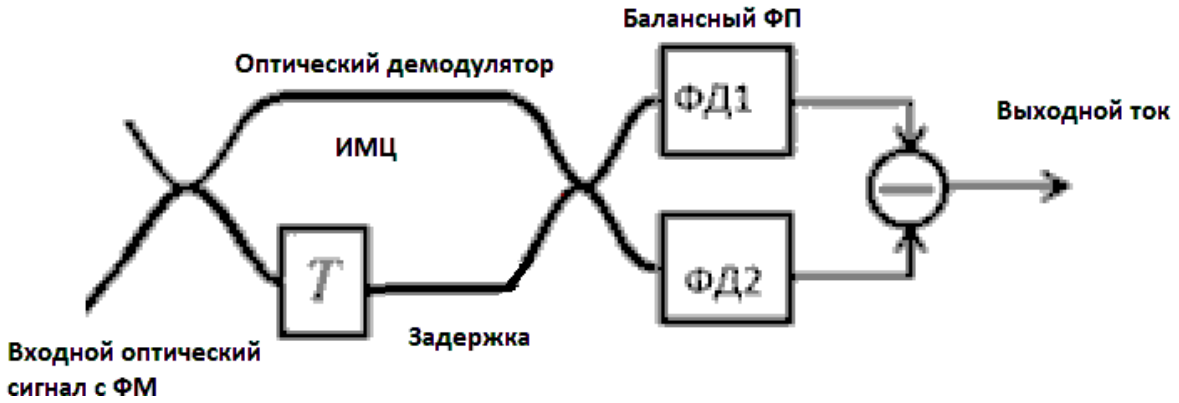


Рис. 2. Структурная схема приемника сигналов с модуляцией DPSK

На рис. 3а (см. ниже) показан бинарный сигнал NRZ DPSK. Там же показана фазовая диаграмма такого сигнала, показывающая, что сигналы логической «1» и «0» имеют одинаковую мощность, а фазы отличаются на 180 градусов. Ниже на рис. 3б показаны сигналы задержанный и не задержанный, а также результат суммирования напряженностей электрических полей этих сигналов. Сигналы с одинаковыми фазами складываются, образуя сигнал логической единицы, а сигналы в противофазе при суммировании подавляются, образуя сигнал логического нуля. В результате сигналы с фазовой модуляцией приобретают АМ (рис. 3б, см. ниже) и могут регистрироваться обычным ФП [1].

Исходные данные для исследований

Параметры ВОСС: скорость передачи B = 10 Гбит/с, длина ОВ1 равна 100 км, длина ОВ2 от 7 до 10 км. При длине ОВ2 10 км обеспечивается полная компенсация ХД.

Параметры передатчиков: уровень пиковой мощности на входе в ОВ p_m от 0 до 10 дБм, кодирование RZ-66 и NRZ.

Параметры приемников: чувствительность $S = 1$ А/Вт, темновой ток $I_T = 10$ нА, частота отсечки фильтра нижних частот (ФНЧ) $\Delta f = 10$ ГГц, спектральная плотность шума 10^{-22} Вт/Гц. Для приемника ФМ дополнительно задана длительность задержки $\Delta T = 100$ пс.

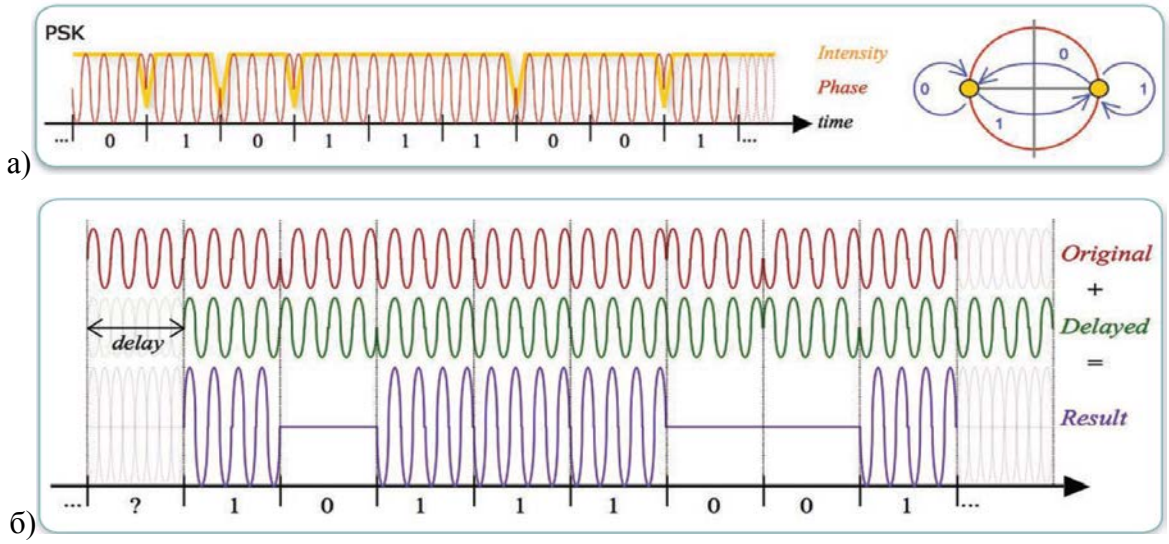


Рис. 3. Временные и фазовые диаграммы сигнала NRZ DPSK (а) и преобразование фазовой модуляции в амплитудную (б)

Результаты исследований зависимости качества связи (Q-фактора)

От входной мощности и коэффициента лавинного умножения (табл. 2), от длины компенсирующего ОВ и типа кодирования (табл. 3), от установленной задержки ΔT и частоты отсечки Δf ФНЧ в ВОСС с ФМ (табл. 4).

ТАБЛИЦА 2. Результаты исследований зависимости качества связи от входной мощности и коэффициента лавинного умножения

№	Установленные величины		Измеренное значение Q-фактора	
	p_0, дБм	M	Для АМ	Для ФМ
1	10	1 (p-i-n ФД)	12,3	12,7
2	10	6	65,2	32,1
3	10	10	94,5	34
4	10	20	132	35,3
5	7	1 (p-i-n ФД)	6,3	6,7
6	7	6	37,7	27
7	7	10	61	31,8
8	7	20	111	35
9	4	6	19,4	18,2
10	0	6	7,7	8,3
11	0	10	13,0	13,2

Из таблицы 2 видно, что при энергетическом приеме преимущества ФМ перед АМ явно невыражены. Незначительные преимущества проявляются при использовании p-i-n ФД (строки 1 и 5) и при малых входных сигналах (строки 10 и 11).

Из таблицы 3 видно, что наиболее эффективным оказалось кодирование RZ-66. оптимальная длина компенсирующего ОВ для RZ кодирования меньше той длины, которая обеспечивает полную компенсацию ХД. Это связано с тем, что с уменьшением длины компенсирующего ОВ уменьшается затухание, но при этом несколько возрастают искажения сигнала за счет ХД.

ТАБЛИЦА 3. Исходные данные: p_0 = 0 дБм, M = 10

№	Установленные величины		Измеренное значение Q-фактора	
	Код	l_2, км	Для АМ	Для ФМ
1	RZ-66	8	21	17
2		9	24	19
3		10	21.7	18.6
4	NRZ	8	13	11
5		9	15	13
6		10	15.5	14.7

Из таблицы 4 видно, что при энергетическом приеме и ФМ качество связи сильно зависит от точности установки задержки ΔT, причем наилучшее качество наблюдается при небольшом отклонении ΔT от номинального значения 100 пс, при ΔT = 100,1 пс. Качество связи зависит также от частоты отсечки ФНЧ Δf на входе регенератора. Оптимальное значение Δf = 7,5 ГГц. После оптимального выбора ΔT и Δf качество связи возросло на 1,5 дБ.

ТАБЛИЦА 4. Исходные данные: p_m = 0 дБм, M = 10, кодирование RZ-66

№	ΔT, *пс*	Δf, *ГГц*	Q
1	100	10	13.2
2	100	5	11.4
3	100	7.5	13.9
4	99.9	7.5	11.7
5	100.1	7.5	15.3
6	100.3	7.5	14.5

Продолжением этой работы будут исследования других перспективных форматов модуляции и кодирования с использованием когерентного приема цифровых оптических сигналов.

Список используемых источников

1. Леонов А., Наний О., Трещиков В. Совершенствование форматов модуляции в оптических системах связи DWDM // Первая миля. 2019 г. № 8. С. 30–36.
2. Богданова Е. Г., Глаголев С. Ф. Исследование высокоскоростных волоконно-оптических систем связи с различными видами модуляции Элект // Бюллетень результатов научных исследований. 2015. Вып. 2 (15). С. 5–16.

Статья представлена научным руководителем, доцентом кафедры ФиЛС СПбГУТ, кандидатом технических наук, доцентом Глаголевым С. Ф.

УДК 003.26.09

А. А. Воронцов (студент гр. ИКТЗ-21м, СПбГУТ)
С. Н. Шемякин (к.т.н., доцент кафедры ЗСС, СПбГУТ)

ИСПОЛЬЗОВАНИЕ ИДЕАЛЬНОГО ШИФРА В КАНАЛЕ ОБРАТНОЙ СВЯЗИ В КВАНТОВОЙ КРИПТОГРАФИИ

Квантовая криптография представляет собой одно из передовых направлений развития современных средств защиты информации. В данной статье предлагается одно из возможных решений проблемы обеспечения целостности данных в классическом канале обратной связи в квантовой криптографии, а именно, использование алгоритмов совершенного имитостойкого шифрования.

информационная безопасность, защищенная связь, модели выбора ключей, секретность ключа, квантовая криптография.

Введение

Квантовая криптография – технология защиты информации в сетях передачи данных, основанная на принципах квантовой физики. Классическая криптография использует математические методы для обеспечения защиты информации. Квантовая криптография изучает возможности передачи данных при помощи объектов квантовой механики. Протоколы квантовой криптографии позволяют создать систему связи, которая всегда сможет обнаружить вторжение злоумышленника.

Основная часть

Алгоритмы квантовой криптографии используются для построения систем распределения секретных ключей между удаленными пользователями. Квантовая криптография предполагает передачу ключей при помощи кван-

товых состояний, при этом секретность ключей гарантируется фундаментальными законами физики. Одной из проблем квантовой криптографии является необходимость использования помимо квантового канала дополнительно классического канала связи для передачи сообщений коррекции ошибок в ключе, отправленном по квантовому каналу связи. Физически оба канала связи могут быть размещены как в одной оптоволоконной линии, так и в совершенно разных средах передачи. Основное различие между ними – принципы передачи сообщений. Единственное требование, предъявляемое к классическому каналу – сохранение целостности данных. Такой канал является математической идеализацией. Для сохранения целостности данных в реальных условиях используются процедуры аутентификации и контроля целостности данных [8].

Злоумышленник может нарушать целостность канала передачи следующими способами: он может осуществлять изменение сообщения, т. е. атаку подмены или введение нового сообщения, т.е. атаку имитации. Способность шифра противостоять воздействиям противника называют имитостойкостью. Количественной мерой имитостойкости шифра являются вероятность успеха имитации p_0 и вероятность успеха подмены p_1 [1].

Совершенный имитостойкий шифр – это шифр, для которого вероятности успеха имитации p_0 и успеха подмены p_1 принимают минимально возможные значения. Совершенный шифр имитозащиты, обеспечивающий минимальные значения вероятности успешной подмены p_1 и имитации p_0 может быть построен на Штейнеровских блок-схемах [2].

Шифр может считаться совершенным при выполнении следующих условий [8]:

- ключ секретен и его знают лишь легитимные пользователи;
- длина ключа в битах не меньше длины сообщения;
- ключ случаен. Поскольку количество источников истинно случайных числовых последовательностей ограничено и большинство из них не обладают требуемой скоростью вычислений, на практике применяются источники псевдослучайных чисел, генерирующие псевдослучайные битовые последовательности, по своим свойствам близкие к истинно случайным;
- ключ используется для шифрования только одного сообщения и в дальнейшем отбрасывается.

Предлагается ввести в канале обратной связи шифратор, работающий по следующему алгоритму. Передаваемые сообщения подвергаются двум видам кодирования: помехоустойчивому кодированию для обеспечения защиты передаваемых сообщений от ошибок в канале передачи и аутентификации бит для защиты сообщений от действий нарушителя.

Помехоустойчивое кодирование может осуществляться с помощью кодов БЧХ. Коды БЧХ, являющиеся частным случаем циклических кодов кор-

рекции ошибок, имеют достаточно простую аппаратную и программную реализацию, однако они подходят лишь для защиты от помех среды передачи и не обеспечивают защиту от нарушителей. Главное отличительное свойство кодов БЧХ – возможность построения кода с заранее определёнными корректирующими свойствами, а именно, минимальным кодовым расстоянием.

Аутентификация бит производится при помощи совершенного шифра имитозащиты. Для описания данного шифра требуется ввести некоторые обозначения [3]:

$M = \{m\}$ – множество всех возможных сообщений, $|M| = \lambda$ – общее число этих сообщений. Особенностью квантового канала является равномерное распределение вероятностей $P(M)$.

$C = \{c\}$ – множество всех возможных криптограмм, $|C| = \mu$ – общее число шифрсообщений.

$K = \{k\}$ – множество всех возможных ключей зашифрования, $|K| = \pi$.

Вероятности имитации и подмены для шифров имитозащиты могут быть вычислены согласно следующим неравенствам, которые приводятся здесь без доказательства [3]:

$$p_0 \geq \lambda/\mu;$$

$$p_1 \geq (\lambda - 1)/(\mu - 1).$$

Шифр имитозащиты называется совершенным, если эти неравенства выполняются как равенства. Равенство имеет место тогда и только тогда, когда образы множества M в множестве C, $E_k : M \rightarrow C$, образуют 2 – (μ, λ, 1)-схему.

Для решения задачи аутентификации бит вводится таблица шифрования, пример которой размещен ниже (табл. 1, см. ниже). На вход шифратора поступают сообщения m_0, m_1 в виде битовых последовательностей. Сообщения передаются в шифратор поочередно. Кроме того, на шифратор поступают ключи шифрования k_0–k_2, сформированные либо распределенные предварительно. Преобразование осуществляется согласно матрице кодера путем побитного сопоставления входного значения и ключа. Шифратор проводит поиск по таблице ячейки, которая соответствует строке ключа и столбцу сообщения. Значение в этой ячейке шифрует передаваемое сообщение. Если ключ или сообщение не найдены в таблице, они отбрасываются, а шифратор готовится к передаче нового сообщения. Полученные шифрограммы c_0–c_2 передаются по каналу связи.

Если исходные тексты и ключи являются равновероятными, то данный шифр оказывается совершенным. В таблице приведен пример реализации

блок-схемы с параметрами 2 – (2, 3, 1). Схема с такими параметрами не может быть использована в реальных криптографических системах, но может применяться для исследования свойств шифров данного класса. Для применения шифра в практических целях длины ключа и шифрограммы должны быть существенно увеличены.

ТАБЛИЦА 1. Таблица шифрования.

K \ M	m_0	m_1
k_0	c_0	c_1
k_1	c_2	c_0
k_2	c_1	c_2

Функция дешифрования осуществляется согласно той же представленной таблице, которая использовалась в прямом преобразовании. В шифраторе удаляются избыточные символы из принятых сообщений. Механизм шифратора проводит поиск шифрограммы по таблице шифрования и, в случае успеха, выдает исходное сообщение, соответствующее принятой шифрограмме. Если принятая шифрограмма не найдена в таблице шифрования, она отбрасывается и фиксируется ошибка приема.

Далее сообщения передаются на декодер БЧХ. Существуют различные алгоритмы декодирования кодов БЧХ. Среди них наиболее широко используемыми являются алгоритм Берлекемпа-Месси, алгоритм Питерсона-Горенстейна-Цирлера, алгоритм Ченя, алгоритм Форни. Выбор конкретного алгоритма декодирования в этой работе не рассматривается.

Отличительной особенностью данной схемы является то, что для представления одного бита исходной информации требуется несколько бит зашифрованного текста. Так, в блок-схеме с параметрами 2 – (2, 3, 1), для защиты от навязывания одного бита данных в канале связи требуется 2 бита при передаче шифрсообщения. Для шифрования расходуется 2 бита ключа. При таком преобразовании вероятность навязывания будет зависеть от количества избыточных символов. Целесообразно увеличить количество избыточных символов в шифрограммах, однако это повлечет за собой увеличение длины передаваемых сообщений и потребует использования каналов связи с большой пропускной способностью. Следовательно, параметры конкретной системы шифрования будут зависеть от характеристик используемого канала связи.

Предлагаемая схема обеспечения контроля целостности обладает существенным недостатком, а именно, большим расходом ключа. В работе [4] предложены несколько примеров почти совершенных шифров, имеющих меньший расход ключа и при этом обеспечивающих стойкое шифрование и аутентификацию. Приведем один пример почти совершенного шифра из [4].

Пусть F_q – поле характеристики 2, состоящее из q элементов, и r – произвольное натуральное число. Рассмотрим шифр $\sum_1$, для которого $S = (F_q)^r$, $K = (F_q)^2$ и $S = (F_q)^{r+1}$. Правило зашифрования строки $s = (s_1, ..., s_r)$ на ключе $k = (a, b)$ определим формулой:

$$E_k(S) = (u_1, \dots, u_r, a + u_1 * b^1 + u_r * b^r),$$

где

$$u_i = s_i + c_i * a + d_i * b, i = 1 \dots r,$$

а $c_1, ..., c_r, d_1, ..., d_r$ — произвольные ненулевые константы из F_q, такие, что

$$c_i * d_j \neq c_j * d_i$$

при $i \neq j$.

Доказывается, что данный шифр является вычислительно-стойким шифром. Длина ключа этого шифра может быть много меньше длины сообщения без существенной потери стойкости шифра в случае, если каждый ключ используется для передачи не более одного сообщения.

Существуют также иные способы обеспечения целостности данных в канале обратной связи, например, алгоритмы строго универсального хеширования. Эти алгоритмы также обладают свойством повышенного расхода ключа и длина шифрсообщения для подобных алгоритмов превышает длину исходного текста в два и более раз.

Заключение

Описываемая схема позволяет обеспечить целостность передачи информации по каналу связи. Достоинством данной схемы является применение кода аутентификации, основанного на алгоритме шифрования с доказанной стойкостью [3].

Недостатками схемы является необходимость ключа большой длины, а также удлинение передаваемого сообщения в несколько раз. Эти обстоятельства ограничивают возможность применения идеальных шифров. Однако, при построении сетей специальной связи этими недостатками можно пренебречь.

Список используемых источников

1. Зубов А. Ю. Коды аутентификации : Монография. М. : Издательство Гелиос АРВ, 2017. 256 е., ил. ISBN 978-5-85438-262-5.

2. Зубов А. Ю. Коды аутентификации с секретностью (обзор) // Математические вопросы криптографии. 2017. Т. 8, вып. 3. С. 5–40. DOI: https://doi.org/10.4213/mvk230.

3. Зубов А. Ю. Совершенные шифры : Монография. М. : Издательство Гелиос АРВ, 2003. 160 с., ил. ISBN 5-85438-076-5.

4. Зубов А. Ю. Почти совершенные шифры и коды аутентификации // Математические вопросы криптографии. 2011. Т. 4, вып. 14. С. 28–33.

5. Коржик В. И., Просихин В. П., Яковлев В. А. Основы криптографии : учебное пособие / рец.: Р. Р. Биккенин, Б. В. Изотов. СПб. : СПбГУТ, 2014. 277 с. : ил. ISBN 978-5-89160-097-3.

6. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применения : учебное пособие. М. : Техносфера, 2005. 320 с. : ил. ISBN 5-94836-035-0.

7. Арбеков И. М. Критерии секретности ключа // Математические вопросы криптографии. 2016. Т. 7, вып. 1. С. 39–56.

8. Тимофеев А. В., Помозов Д. И., Маккавеев А. П., Молотков С. Н. О структуре открытого классического канала связи в квантовой криптографии: коррекция ошибок, целостность и аутентичность // ЖЭТФ, Т. 131:5 (2007), С. 771–789.

УДК 004.045

Д. И. Восьмаков, Н. В. Сергеев,
О. А. Якубова (студенты гр. ИКПИ-14, СПбГУТ)

РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА ГОРОДСКОЙ ИНФРАСТРУКТУРЫ

В настоящей работе предлагается концепция приложения, обеспечивающего возможность сбора и обработки информации об инфраструктурных объектах городской среды. Представлено описание варианта реализации приложения, построенного на данной концепции. Описаны особенности системы, возможности применения программного решения в государственном управлении.

инфраструктура, городская среда, автоматизация, цифровизация, централизация, мониторинг, база данных.

Преимущества и недостатки современных систем мониторинга

Качество муниципального и государственного управления во многом зависит от возможности получать данные о состоянии управляемых объектов и представлять их в удобном для пользователя виде. С развитием технологий всё больше ведомств стараются применить программные решения для обеспечения своей деятельности. Как правило, цифровизации подвергается не только деятельность подразделений, связанных с оперативной работой, но и работа управляющих структур [1]. Современные программные решения обеспечивают возможность прохождения потока данных с низового уровня на самый верх управляющего контура.

Так, в настоящее время широко используются системы мониторинга городской инфраструктуры, передающие данные непосредственно от объектов лицам, принимающим решения. Городская система видеонаблюдения

позволяет получить видеоматериалы, на которых зафиксированы противоправные действия, непосредственный доступ к ним имеют только лица, имеющие право работать с подобной информацией [2]. Датчики, установленные на трубопроводах, помогают заблаговременно обнаружить отклонения от нормы, свидетельствующие о неисправностях, и устранить угрозу аварии, минимизируя расходы на ремонт [3].

Преимущества использования подобных систем в сфере государственного управления перечислены ниже [4]:

- Возможность предотвращения и быстрого реагирования на чрезвычайные ситуации техногенного характера, связанные с состоянием инфраструктурных объектов.
- Повышение эффективности планирования ремонта, обслуживания и строительства инфраструктурных объектов благодаря возможности получения актуальных данных в режиме реального времени.
- Сокращение времени на обнаружение и устранение дефектов и недостатков объектов инфраструктуры, например, опасных поворотов, разрушенного дорожного полотна, неисправных светофоров.
- Удобство сбора и обработки статистических данных, которые в дальнейшем можно использовать для оценки текущего состояния города и прогнозирования.

Современная практика использования программных решений в сфере государственного управления показывает следующие недостатки систем мониторинга и информирования [5]:

- Многие системы мониторинга узкоспециализированы и предназначены для работы лишь с ограниченной категорией объектов.
- Обработка большого количества данных из различных изолированных друг от друга систем мониторинга создает трудности для внедрения новых программных решений, использующих эти данные в своей работе, может стать причиной дублирования информации, что приводит к увеличению затрат технических ресурсов.
- Централизованный сбор данных об объектах инфраструктуры часто требует внесения изменений в программные интерфейсы источников данных для обеспечения одинакового формата данных в системе и вне её.

Агрегирование различных показаний датчиков, установленных в городской среде, позволит использовать их для работы интеллектуальной системы, способной управлять отдельными объектами инфраструктуры, например, светофорами, уличными фонарями и прочими устройствами для более экономичного потребления ресурсов, комфорта населения и т. д. [6].

Системы централизованного мониторинга городской инфраструктуры

Сегодня на российском рынке представлены в основном узкоспециализированные системы сбора и обработки данных, никак не взаимодействую-

щие между собой. Однако, существуют проекты, отвечающие идее централизованного сбора информации. Например, концепция технологии «умного города» Санкт-Петербурга, разработанная сотрудниками университета ИТМО [6], как раз предполагает получение информации от датчиков и дальнейшее её использование для анализа и оптимизации принятия решений. Одним из принципов «умного города» авторы проекта выделяют целостное цифровое описание города на основе мониторинга и открытие этих данных для населения и бизнеса, но не приводят описания архитектуры и принципа работы программного обеспечения, предназначенного для этой задачи.

Концепция системы мониторинга городской инфраструктуры

Целью настоящей работы является описание концепции и вариантов реализации универсальной системы, позволяющей осуществлять сбор и хранение данных объектов городской инфраструктуры, лишённой вышеперечисленных недостатков.

Актуальность разработки обоснована перечисленными выше недостатками существующих программных решений для мониторинга инфраструктуры.

Необходимым условием для функционирования системы является наличие источников данных в виде программных интерфейсов (API), позволяющих получать массив данных об инфраструктурных объектах.

Согласно поставленной задаче, разрабатываемое решение должно обеспечивать следующие возможности:

- Сбор данных от объектов инфраструктуры, подразумевающий выполнение запроса к программному интерфейсу, получение данных в формате JSON и преобразование полученных данных во внутренний формат приложения.
- Сохранение полученной информации в базе данных.
- Непосредственный доступ к данным для их редактирования или удаления.
- Доступ к графическому интерфейсу пользователя как пример представления данных.
- Экспорт данных в другие системы.

Для обеспечения выполнения вышеуказанных функций система должна содержать следующие компоненты:

- База данных.
- Модуль, обеспечивающий сбор данных и их сохранение в базе данных.
- Клиентское приложение, позволяющее выполнять обработку данных и представлять их в удобном для пользователя виде.

Схема архитектуры программного решения представлена на рис. 1.

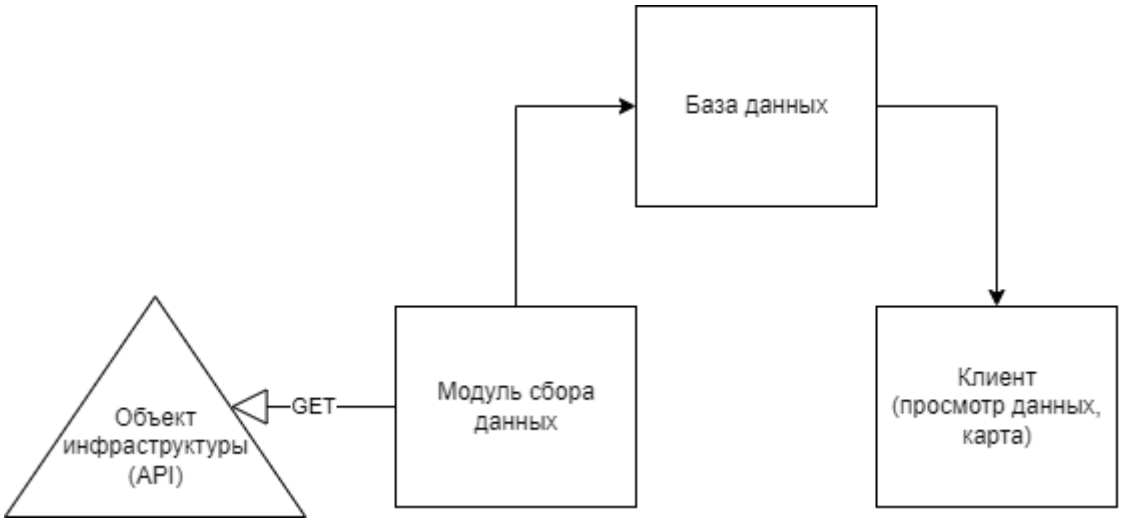


Рис. 1. Схема архитектуры программного решения

Объектная модель системы включает в себя абстрактный класс, содержащий поля для основных сведений об объекте любого типа (название, локация), а также классы, характеризующие ту или иную категорию объекта с присущими только им полями (объект энергетической, транспортной инфраструктуры, объекты водоснабжения и т. д.).

Подробно следует рассмотреть принцип работы модуля сбора данных. Для получения информации от источников данных могут быть использованы нижеперечисленные способы (рис. 2):

● Отправка запросов к источникам данных. Недостаток данного подхода состоит в том, что формат данных, полученных в результате запроса, может отличаться от внутреннего формата программы. Таким образом, данные, поступающие в систему, требуют дополнительной обработки.

● Отправка запросов от источников данных к системе. Очевидный недостаток состоит в необходимости внесения изменений в другие системы на этапе внедрения.

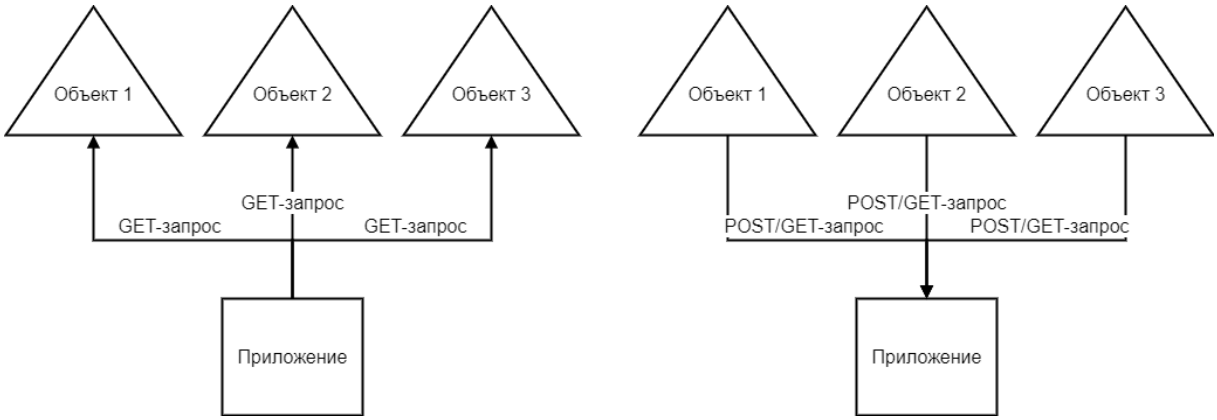


Рис. 2. Подходы к сбору данных от объектов инфраструктуры

Наименее затратным в данном случае является первый подход к получению данных, поэтому в настоящей работе применён именно он. Следовательно, модуль сбора данных должен содержать функционал для приведения полученных данных к требуемому формату.

База данных системы должна состоять из таблиц, в каждой из которых находятся данные об объектах определённой категории.

Система должна предусматривать возможность экспорта данных в другие системы. С помощью данных, собранных от различных источников в одной системе, автоматизированные средства и ответственные лица могут быстро, без дополнительных затрат на сбор необходимой информации, принимать эффективные решения.

Данное программное решение может быть использовано для обеспечения деятельности исполнительной власти на уровне муниципалитета (город, село). Кроме того, возможно использование системы, построенной на данных принципах, для отслеживания состояния оборудования на предприятии, а также для мониторинга IT-инфраструктуры. В отдалённой перспективе возможно внедрение данной системы в киберсреду «Умных городов».

Вывод

Таким образом, проблема централизованного сбора информации об объектах инфраструктуры может быть решена с использованием универсального программного решения с достаточно простой архитектурой без необходимости внесения изменений в существующие программные интерфейсы источников данных.

Список используемых источников

1. Цифровое государственное управление: Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации. [Электронный ресурс]. URL: https://digital.gov.ru/ru/activity/directions/882/ (дата обращения 20.05.2023).

2. СПб ГКУ ГМЦ [Электронный ресурс]. URL: http://spb112.ru/catalogue/4/ (дата обращения 20.05.2023).

3. ТАСС [Электронный ресурс]. URL: https://tass.ru/obschestvo/15197003 (дата обращения 20.05.2023).

4. «Умный город» СмартСити. Концепция, безопасность и экономика Умного города [Электронный ресурс]. URL: https://intelvision.ru/services/smartcity (дата обращения 20.05.2023).

5. 15 преимуществ и недостатков централизированной базы данных [Электронный ресурс]. URL: https://nextphase.ru/15-%D0%BF%D1%80%D0%B5%D0%B8%D0%BC%D1%83%D1%89%D0%B5%D1%81%D1%82%D0%B2-%D0%B8-%D0%BD%D0%B5%D0%B4%D0%BE%D1%81%D1%82%D0%B0%D1%82%D0%BA%D0%BE%D0%B2-%D1%86%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D0%B8%D0%B7/ (дата обращения 20.05.2023).

6. «Умный Санкт-Петербург»: комплексный подход к внедрению информационных технологий управления мегаполисом [Электронный ресурс]. URL: https://controlengrussia.com/otraslevye-resheniya/zhkh/umnyj-sankt-peterburg/ (дата обращения 20.05.2023).

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.338

А.С. Дворцов, Д. В. Киселев, А. И. Мамлеев
(студенты гр. ИКПИ-13, СПбГУТ)

SIDE CHAIN – ПЕРСПЕКТИВЫ И ВАРИАНТЫ ПРИМЕНЕНИЯ

Side Chain – это технология, позволяющая создавать дополнительные блокчейны с индивидуальными правилами и параметрами, связанные с главным блокчейном. Ее применение может увеличить скорость и эффективность обработки транзакций в финансовой сфере, улучшить контроль качества и сократить время доставки в логистике, а также повысить безопасность медицинских данных пациентов в медицине. Side Chain открывает новые возможности для улучшения эффективности и безопасности в различных сферах деятельности. Кроме того, Side Chain может использоваться для создания новых децентрализованных приложений, которые будут иметь большую гибкость и возможность развития, чем приложения, работающие на основном блокчейне.

blockchain, side chain, вариативность применения.

Sidechain. Принцип работы (рис. 1)

Работа сайдчейна основана на следующих ключевых принципах:

● Two-way pegging (Двусторонняя привязка): Это означает, что токены можно передавать между основным блокчейном и сайдчейном в любом направлении. Например, если у вас есть токен Ethereum, вы можете заблокировать его на основном блокчейне и получить эквивалентное количество токенов на сайдчейне. Позже вы можете вернуть токены обратно на основной блокчейн, заблокировав их на сайдчейне.

● Безопасность: Безопасность сайдчейна должна быть гарантирована основной блокчейн-сетью. Это гарантирует, что средства, заблокированные на основной сети, не могут быть похищены, даже если сайдчейн подвергается атаке.

● Самостоятельность: Сайдчейны могут иметь свои собственные уникальные функции и протоколы, которые могут быть адаптированы под специфические потребности. Это означает, что они могут использовать другие

алгоритмы консенсуса, поддерживать умные контракты, обеспечивать большую пропускную способность и так далее.

Вот как происходит переход между основной блокчейн и сайдчейн:

1. Токены блокируются на основной блокчейн сети.
2. Доказательство блокировки передается в сайдчейн.
3. Сайдчейн выпускает эквивалентное количество своих токенов.
4. Пользователи могут использовать эти токены внутри сайдчейна, как им угодно.
5. Чтобы перевести токены обратно на основной блокчейн, токены на сайдчейне сначала блокируются.
6. Доказательство блокировки передается обратно на основной блокчейн.
7. Основной блокчейн разблокирует эквивалентное количество токенов.

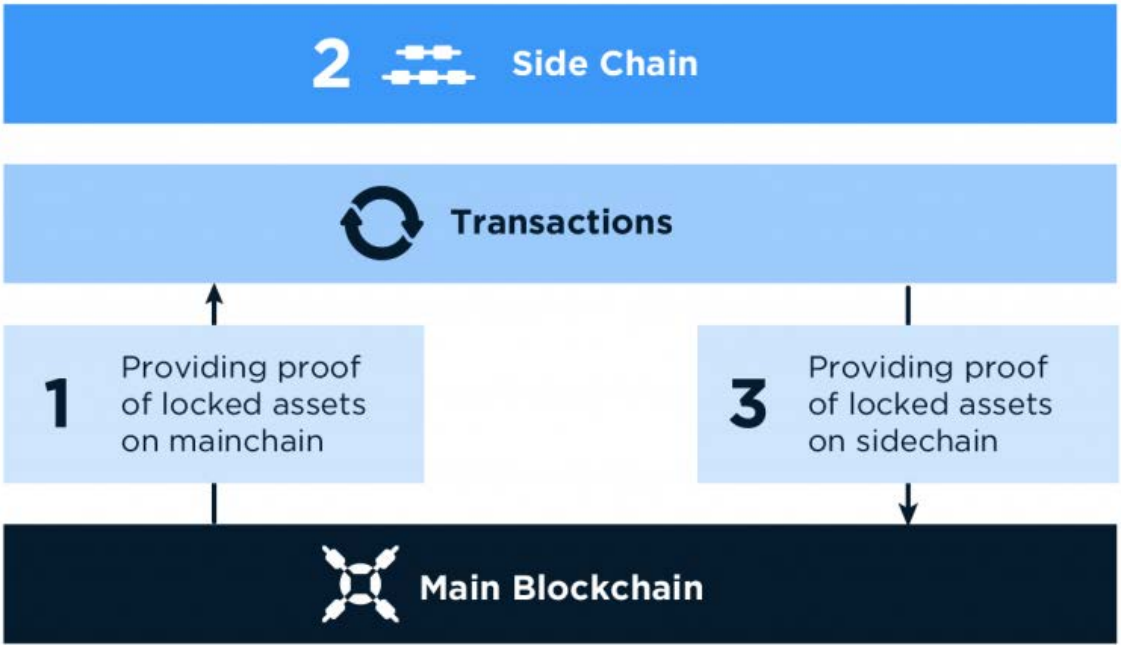


Рис. 1. Принцип работы Side Chain

Примеры сайдчейнов

1. **Lightning Network (рис. 2):** Это второй слой (Layer 2) технологии, разработанный для увеличения скорости транзакций Bitcoin. Он работает, создавая каналы платежей между участниками, которые могут проводить любое количество транзакций внутри канала, минуя блокчейн Bitcoin. Только когда канал открывается или закрывается, информация записывается в основной блокчейн. Это существенно увеличивает скорость транзакций и уменьшает сборы.

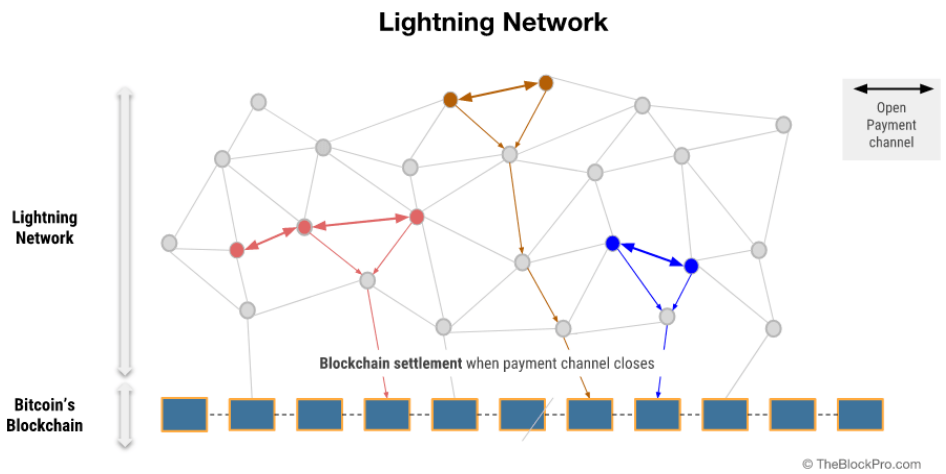


Рис. 2. Визуализация работы Lightning Network

2. **Polygon (ранее *Matic Network*) (рис. 3):** Polygon представляет собой многослойную сеть масштабирования для Ethereum, предлагающую структуры как Layer 2, так и сайдчейн. Сайдчейны Polygon работают параллельно основной сети Ethereum и предлагают значительно более быстрые и дешевые транзакции. Polygon поддерживает различные протоколы масштабирования, включая PoS, Plasma и другие.

3. **Binance Smart Chain (BSC) (рис. 4):** Binance Smart Chain это блокчейн, работающий параллельно и совместимый с Binance Chain. Он добавляет поддержку смарт-контрактов и совместим с Ethereum Virtual Machine (EVM), что делает его совместимым с Ethereum-деплоями, инструментами и библиотеками. BSC не является сайдчейном в классическом понимании этого термина, так как он не использует двустороннюю привязку токенов с Binance Chain и имеет собственный набор валидаторов. Однако, он обеспечивает схожую функциональность, позволяя быстрые и дешевые транзакции и создание децентрализованных приложений (DApps).

Рис. 3. Polygon network

Рис. 4. Binance Smart Chain

Итак, хотя все три из этих систем предназначены для улучшения масштабируемости и производительности сетей блокчейн, они делают это поразному и имеют разные свойства и функциональные возможности.

Заключение

Технология **Side Chain** – это революционная концепция, которая имеет потенциал изменить различные отрасли, улучшая эффективность и безопасность. Ее применение может привести к более быстрым временам обработки

транзакций, снижению комиссий за транзакции и улучшению контроля качества.

Кроме того, технология **Side Chain** может использоваться для использования новых децентрализованных приложений, которые предлагают большую гибкость и потенциал развития, чем приложения, работающие на основном блокчейне.

Список используемых источников

1. Enabling Blockchain Innovations with Pegged Sidechains [Электронный ресурс].URL: https://blockstream.com/sidechains.pdf (дата обращения 05.06.2023).

2. What Are Sidechains? Scaling Blockchain on the Side [Электронный ресурс]. URL: https://crypto.com/university/what-are-sidechains-scaling-blockchain (дата обращения 02.06.2023).

3. An Introduction to Sidechains [Электронный ресурс]. URL: https://www.coindesk.com/learn/an-introduction-to-sidechains/ (дата обращения 05.06.2023).

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.942

Л. О. Дубовиков, И. Е. Серебряный, Д. А. Юклутов
(студент гр. ИКТС-01, СПбГУТ)

ПРИМЕНЕНИЕ ИНЕРЦИАЛЬНОЙ НАВИГАЦИИ В СИСТЕМАХ LBS

В настоящее время всё большую популярность приобретают телекоммуникационные услуги, использующие данные о местоположении абонента (LBS). Такие данные могут быть получены с помощью систем спутниковой навигации (ГНСС), или на основе обмена данными между мобильным терминалом и мобильными станциями оператора сотовой связи. Однако в условиях плотной городской застройки на распространение сигнала оказывает влияние множество факторов, что негативно сказывается на точности позиционирования. В данной статье рассматривается применение инерциальной навигации в системах LBS.

LBS,ГНСС,АТ,ИНС,МЭМС, гироскоп, акселерометр.

Введение

Российская Федерация является одним из лидеров по количеству мобильных телефонов на душу населения, обогнав такие развитые страны как США, Китай, Японию и Южную Корею. Среди наиболее востребованных приложений, устанавливаемых на смартфоны или планшеты, являются ГИС приложения, которые позволяют прокладывать маршруты и определять местоположение пользователя мобильного устройства на карте, а также получать полезную информацию об окружающих объектах.

1 Сервисы, основанные на местоположении

Location-based service, LBS – тип информационных и развлекательных услуг, основанных на определении текущего местоположения пользователя. Одним из наиболее популярных сервисов является Глобальная Навигационная Спутниковая Система (ГНСС) – система, которая предназначена для определения местоположения различных объектов, будь то они в воздухе или на земле, путем определения их географических координат [1]. Несмотря на всю надежность и точность данной системы, сигналы от спутников достигают земли очень ослабленными и в условиях плотной застройки или нахождения устройств в здании, определение их местоположения может быть неправильным. Система инерциальной навигации может стать отличным дополнением, учитывая, что смартфоны есть у подавляющего большинства людей, а в смартфонах есть микроэлектромеханические системы (МЭМС).

2 Основные устройства на базе МЭМС

Микроэлектромеханическая система, или МЭМС, представляет собой миниатюрное устройство, которое изготовлено как из механических, так и из электрических компонентов, используя методы микрообработки. Они преобразуют некоторые механические сигналы в электрические или оптические[2]. На их базе строятся следующие устройства для определения местоположения объекта с использованием инерциальной системы навигации.

Акселерометр – это прибор для измерения ускорения, который работает как датчик изменения положения устройства в пространстве.

Три ортогональных акселерометра, жестко закрепленных на теле, проводят измерения ускорения, скорости и положения тела в трехмерном пространстве. Однако до тех пор, пока акселерометры жестко прикреплены к движущемуся телу, их опорная система отсчета является связанной системой отсчета, которой как правило, недостаточно для обеспечения отслеживания движения тела по отношению к внешней системе отсчета. Вращение тела по отношению к внешней системе отсчета, то есть положение тела, должно быть вычислено из мгновенного углового положения тела, полученного из трех гироскопов, жестко закрепленных на теле по ортогональным

осям акселерометра. Таким образом, информация о положении тела используется для производства измерений акселерометром в опорной системе отсчета [3].

Алгоритм инерциальной системы определяет текущее местоположение пользователя на основании начальных данных и измеренных ускорений по трем осям декартовой системы координат:

$$\begin{aligned} Y_з(t) &= Y_з(0) + v_{y,0}\cdot t + \int_0^T v_y(t)dt = Y_з(0) + v_{y,0}\cdot t + \int_0^T\int_0^T a_y(t)dtdt \\ X_з(t) &= X_з(0) + v_{x,0}\cdot t + \int_0^T v_x(t)dt = X_з(0) + v_{x,0}\cdot t + \int_0^T\int_0^T a_x(t)dtdt \\ Z_з(t) &= Z_з(0) + v_{z,0}\cdot t + \int_0^T v_z(t)dt = Z_з(0) + v_{z,0}\cdot t + \int_0^T\int_0^T a_z(t)dtdt \end{aligned} \quad (1)$$

где $X_з(0)$, $Y_з(0)$, $Z_з(0)$ – начальные координаты пользователя; $v_{x,0}$, $v_{y,0}$, $v_{z,0}$ – начальные скорости объекта вдоль трех осей системы координат здания; $v_x(t)$, $v_y(t)$, $v_z(t)$ – скорости движения вдоль трех осей системы координат здания.

Величины кажущихся ускорений рассчитываются согласно выражению до выполнения пространственного преобразования:

$$\begin{pmatrix} \hat{a}_x(t) \\ \hat{a}_y(t) \\ \hat{a}_z(t) \end{pmatrix} = \begin{pmatrix} a_{\text{изм},\,x}(t) \\ a_{\text{изм},\,y}(t) \\ a_{\text{изм},\,z}(t) \end{pmatrix} - \begin{pmatrix} g_x(t) \\ g_y(t) \\ g_z(t) \end{pmatrix} = \begin{pmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{pmatrix} + \begin{pmatrix} \Delta a_x(t) \\ \Delta a_y(t) \\ \Delta a_z(t) \end{pmatrix}, \quad (2)$$

где $\hat{a}_x(t)$, $\hat{a}_y(t)$, $\hat{a}_z(t)$ – оценки абсолютных ускорений по трем направлениям; $a_{\text{изм},x}(t)$, $a_{\text{изм},y}(t)$, $a_{\text{изм},z}(t)$ – кажущиеся ускорения, снятые акселерометром по трем направлениям; $\Delta a_x(t)$, $\Delta a_y(t)$, $\Delta a_z(t)$ – погрешности измерений по трем направлениям, обусловленные дрейфом акселерометра.

С другой стороны, в каждый момент времени вектор кажущихся ускорений может быть представлен как:

$$\begin{pmatrix} a_{\text{изм},\,x}(t) \\ a_{\text{изм},\,y}(t) \\ a_{\text{изм},\,z}(t) \end{pmatrix} = \mathbf{R}'\cdot \begin{pmatrix} 0 \\ 0 \\ \|a_{\text{изм}}(t)\| \end{pmatrix}, \quad (3)$$

где $\|a_{\text{изм}}(t)\| = \sqrt{a^2_{\text{изм},\,x}(t) + a^2_{\text{изм},\,y}(t) + a^2_{\text{изм},\,z}(t)}$ – норма вектора ускорения в момент времени t, $\mathbf{R}'$ – матрица вращения, описывающая положение телефона в пространстве, которая может быть получена по данным гироскопа и акселерометра.

Тогда выражение (2) может быть преобразовано к следующему виду:

$$\begin{pmatrix} \hat{a}_x(t) \\ \hat{a}_y(t) \\ \hat{a}_z(t) \end{pmatrix} = \mathbf{R}' \cdot \left(\begin{pmatrix} 0 \\ 0 \\ \|a_{\text{изм}}(t)\| \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \right) = \mathbf{R}' \cdot \begin{pmatrix} 0 \\ 0 \\ \|\hat{a}(t)\| \end{pmatrix}. \quad (4)$$

Таким образом, выражение (1) с учетом (2)–(4) принимает вид:

$$\begin{pmatrix} \hat{X}_з(t) \\ \hat{Y}_з(t) \\ \hat{Z}_з(t) \end{pmatrix} = \begin{pmatrix} X_з(0) \\ Y_з(0) \\ Z_з(0) \end{pmatrix} + \begin{pmatrix} v_{x,0} \\ v_{y,0} \\ v_{z,0} \end{pmatrix} t + \int_0^T \int_0^T \begin{pmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{pmatrix} dtdt + \int_0^T \int_0^T \begin{pmatrix} \Delta a_x(t) \\ \Delta a_y(t) \\ \Delta a_z(t) \end{pmatrix} dtdt = \begin{pmatrix} X_з(t) \\ Y_з(t) \\ Z_з(t) \end{pmatrix} + \begin{pmatrix} \Delta X_з(t) \\ \Delta Y_з(t) \\ \Delta Z_з(t) \end{pmatrix},$$

где $\hat{X}_з(t)$, $\hat{Y}_з(t)$, $\hat{Z}_з(t)$ – оценки координат объекта; $\Delta X_з(t)$, $\Delta Y_з(t)$, $\Delta Z_з(t)$ – погрешности оценки координат.

Гироскоп представляет собой датчик, способный измерять вращательное движение, определяя скорость вращения, которое датчик совершает вокруг предопределенной оси. Таким образом, угловая ориентация тела может быть вычислена путем интегрирования измерений угловой скорости при условии, что задана начальная ориентация оси датчика по отношению к опорной оси [3]. Углы Эйлера оказались неудобными для описания гироскопических явлений. Причина состоит в следующем. Ось быстро вращающегося ротора в большинстве случаев мало отклоняется от некоторого среднего положения. В прикладной теории гироскопов широкое применение находят углы Резаля. Для более наглядного представления углов Резаля воспользуемся схемой гироскопа (рис. 1).

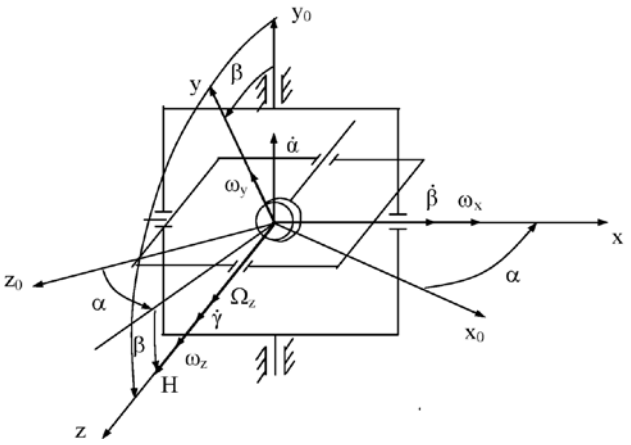


Рис. 1. Углы Резаля

3 Системы координат здания, человека и телефона, переход от одной системы координат к другой

В настоящее время все чаще стали использоваться интегрированные системы определения местоположения и навигации посетителей внутри помещений, использующие показания датчиков явлений различной физической природы.

Задача поиска координат объекта при использовании инерциальных датчиков представляет собой задачу нахождения соответствия между системами координат здания, человека и мобильного устройства. Данные координаты основаны на инерциальной системе отсчета. Началом данной системы отсчета является центр Земли, ось z которой совпадает с полярной

осью Земли. Эту систему отсчета можно применить к каждому объекту в отдельности. Система отсчета будет состоять из 3 осей *X* (крен), *Y* (тангаж), *Z* (рысканье). Поворот рысканья формирует новую систему координат. Получается, что ось *Z* остаётся совмещенной с инерциальной осью, а *X* и *Y* поворачиваются на задаваемый угол.

4 Результаты испытаний гироскопа и акселерометра

В качестве экспериментального устройства использовался смартфон с установленным на нем приложением *Physics Tooldox Suite*. На рисунках представлены данные акселерометра и гироскопа после перемещения смартфона по небольшому участку (рис. 2). Исходя из рисунков можно определить в каких направлениях двигалось устройство, время в течении которого проводился эксперимент, а также углы поворота устройства в пространстве. Все эти данные могут помочь точно определить местоположение объекта в здании.

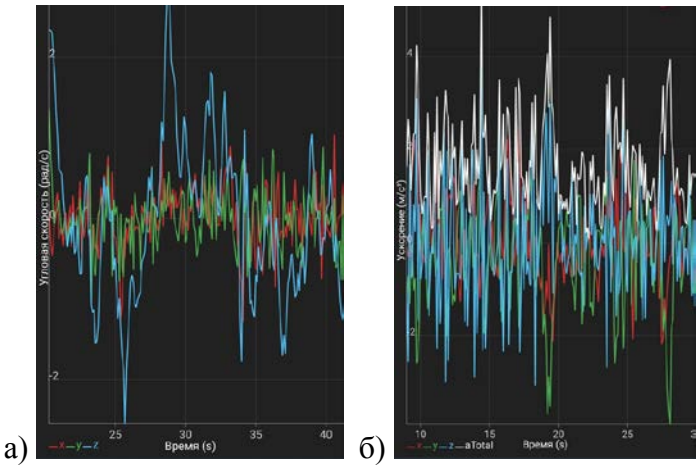


Рис. 2. График гироскопа (а), график акселерометра (б)

Заключение

Системы инерциальной навигации представляют широкий круг возможностей для вычисления местоположения объекта в пространстве. Имея алгоритмы обработки данных, при помощи множественных измерений можно довольно точно определять координаты объекта. Знание точного местоположения объекта в данный момент времени позволит повысить качество предоставляемых пользователю услуг.

Список используемых источников

1. https://dic.academic.ru/dic.nsf/ruwiki/116944
2. https://new-science.ru/chto-takoe-mems-mikroelektromehanicheskaya-sistema-tipy-i-primenenie/
3. Махиянова Е. Б. Методы спутникового и наземного позиционирования. Перспективы развития технологий обработки сигналов / под ред. к. т. н. Турилова В. А. : пер. с англ. М. : ТЕХНОСФЕРА, 2012. 532 с.

Статья представлена научным руководителем, доцентом кафедры ССиПД СПбГУТ, кандидатом технических наук Гришиным И. В.

УДК 004.75

К. С. Ефременко, А. С. Кузьмин (студенты гр. ИКТК-12, СПбГУТ)

ОБЗОР СУЩЕСТВУЮЩИХ КОНЦЕПЦИЙ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

В настоящей работе представлен обзор концепций распределенных вычислений, цель которого – аналитическое сравнение, систематизация информации и сопоставление существующих концепций и моделей. В настоящее время в открытых источниках используются различная терминология и представление о распределенных вычислениях, при этом зачастую информация носит разрозненный характер.

распределенные вычисления, облачные технологии, туманные вычисления, граничные вычисления, микро-облака, макро-облака.

На сегодняшний день известны три наиболее популярные существующие концепции распределенных вычислений: облачные, туманные и граничные. Помимо этого, также существует концепция «вычислений в росе» (с англ. *Dew Computing*) [1]. Отдельно существуют модели макро-, микро-, мини-облаков [2], а также модель мульти-облака. В условиях обрезавшегося разнообразия терминов видится целесообразным осуществить обзор существующих на сегодняшний момент концепций, а также выполнить их первичное сравнение и сопоставление с целью определения взаимосвязей и различий в предлагаемых сегодня научно-техническим сообществом вариантов.

На первом этапе обзора принято решение рассмотрения концепций облачных, туманных и граничных вычислений.

Облачные вычисления представляют собой вычислительные службы (в том числе серверы, хранилища, базы данных, сети, программное обеспечение и т. д.), которые предоставляются через Интернет (так называемое «облако»). Когда между облаком и пользовательскими устройствами есть связь, то между ними происходит обмен данными в обоих направлениях: устройства отправляют телеметрические и пользовательские данные в облако, а облако отправляет устройствам команды или уведомления и результаты обработки пользовательских данных. В итоге, благодаря облачным вычислениям у пользовательских устройств появляется больше возможностей и повышается их производительность, а также множество пользователей могут работать с данными одновременно независимо от местоположения.

Как было указано в начале данной статьи, в рамках концепции облачных вычислений выделяются также модели облаков (отличные от моделей обслуживания – SaaS, IaaS и др.). Так в открытых источниках предлагается также выделять такие модели как:

– микро-облака, которые размещаются на уровне пользователей,

– мини-облака, размещаемые на уровне беспроводных сетей и групповых элементов оптических сетей доступа,

– макро-облака, размещаемые на уровне ядра операторской сети,

– мульти-облака, представляющие собой сочетание облаков для хранения больших объемов данных с централизованными вычислительными облаками,

– удаленные облака, представляющие собой ресурсы крупных мировых компаний.

Следует вернуться к рассмотрению концепции, обозначенных для обзора на первом этапе и параллельно выполнять сравнение с указанными выше моделями.

Туманные вычисления – концепция распределенных вычислений, которая занимает промежуточную позицию между пользовательскими устройствами и облачными центрами обработки данных. Туманные вычисления возможно соотнести с мини-облаком, поскольку мини-облако аналогичным образом располагается в «промежуточной» позиции.

Также, упоминая о туманных вычислениях, также стоит рассмотреть про такое понятие, как «вычисления в росе». Так называемые «росистые вычисления» находятся в непосредственной близости к пользователю, а обработка или хранение информации выполняется на конечных узлах сети, на которых также происходит максимально возможная предобработка данных перед передачей на облачный уровень. Например, датчик температуры и влажности может самостоятельно определить, когда в помещении становится слишком сухо или жарко и дать сигнал на управляющее устройство, такое как кондиционер или увлажнитель, чтобы они включились. Стоит отметить, что при этом полноценный анализ, а также хранение данных, собранных за время работы системы, осуществляется в облаке. Таким образом, вычисления в росе возможно рассматривать как вычисления схожие с микросервисами, т. к. росистые приложения – это специальная группа приложений, удовлетворяющих критериям независимости и сотрудничества [3].

Подходы туманных и граничных вычислений близки, однако имеют отличия. В туманных информационных системах вычисления данных производятся в узле, недалеко от места получения, в то время как граничные вычисления позволяют обрабатывать информацию в том же месте, где они и были получены, что позволяет обеспечить минимальную задержку отклика. Данное обстоятельство связано с тем, что в граничных вычислениях используется программируемый логический контроллер, который может входить в систему управления производственными процессами.

Примером граничных вычислений могут послужить автономные транспортные средства, такие как беспилотные автомобили, которые анализируют данные датчиков на борту транспортного средства, чтобы уменьшить

задержку из-за огромного объема данных. Также примером служат и умные дома [4].

Сравнивая граничные вычисления и вычисления в росе также стоит обратить внимание на характерную особенность росистых вычислений – фокус на приложении, тогда как граничные вычисления концентрируются на граничном объекте-устройстве (автономное транспортное средство, умный дом и т. д.). Таким образом, граничные вычисления и вычисления в росе возможно соотнести с моделью микро-облака, располагающегося наиболее близко к конечным пользовательским устройствам.

Относительно макро-облака, мульти-облака и удаленного облака, следует отметить, что данные модели видится целесообразным рассматривать и соотносить в каждом отдельном случае и примере облака, поскольку очевидна прямая зависимость от расположения и назначения (на уровне ядра операторской сети, хранение больших объемов данных с централизованными вычислительными облаками, ресурсы крупных мировых компаний).

В качестве конечного результата настоящего обзора составлена сравнительная таблица, рассматривающая три основные концепции вычислений, определенных на первом этапе работы (табл. 1).

ТАБЛИЦА 1. Сравнение концепций распределенных вычислений

Требование	Облачные вычисления	Туманные вычисления	Граничные вычисления
Задержки (качественная характеристика)	Высокие	Низкие	Практически неощутимы
Неустойчивость синхронизации (качественная характеристика)	Высокая	Низкая	Практически отсутствует
Расположение серверов	В глобальной сети	На периферии сети	В самом устройстве
Расстояние между клиентом и сервером	Множественные переходы	Один или несколько переходов	Один переход
Известно ли место расположения данных	Нет	Да	Да
Безопасность	Не определена	Может быть определена	Определена
Атака на данные в процессе их передачи	Высокая вероятность	Очень низкая вероятность	Очень низкая вероятность
Географическое распределение	Централизованное	Децентрализованное или централизованное в определенных областях	Децентрализованное

Требование	Облачные вычисления	Туманные вычисления	Граничные вычисления
Количество серверов	Несколько	Очень много	Несколько
Поддержка масштабируемости	Ограниченная	Поддерживается	Поддерживается
Тип подключения	Выделенная линия	Беспроводное соединение	Беспроводное соединение
Аппаратное обеспечение	Значительное и масштабируемое память и вычислительные мощности	Ограниченная память и вычислительные мощности	Ограниченная память и вычислительные мощности
Условия эксплуатации	Большие здания с хорошей системой кондиционирования	На улице или внутри неспециализированных помещений	На улице, в технике и в домах

В результате второго этапа настоящего исследования и обзора составлена следующая схема, иллюстрирующая соотношение всех упомянутых в работе моделей и концепций (рис. 1). Следует отметить, что поскольку представленный обзор является первичным, то в последующих работах настоящая схема может быть уточнена.

Рис. 1. Схема соотношения рассмотренных моделей и концепций

Таким образом, в настоящей работе отражены результаты образа – попытка систематизации разрозненной информации о существующих моделях и концепциях распределенных вычислений. Следует предположить, что подобная ситуация образования различных связанных концепций могла возникнуть вследствие разработки и рассмотрения вычислений различными сообществами – отдельно техническим и научным. Так следует акцентировать

внимание актуальности вопросов стандартизации и возможно относительной синхронизации исследований, а также на так называемом мультистейкхолдерном подходе, который подразумевает координацию действий розничных заинтересованных сторон, в том числе и в рамках конференций и форумов.

Список используемых источников

1. Yingwei Wang. Cloud-dew architecture // Int. J. Cloud Computing. 2015. Т. 4. № 3. С. 199–210.

2. Бородин А. С., Кучерявый А. Е. Сети связи пятого поколения как основа цифровой экономики // Электросвязь. 2017. № 5. С. 45–49.

3. Росистые вычисления – новая парадигма и развитие облачных вычислений // Портала DocPlace. 2017. URL: https://docplace.ru/dewcomputing/ (дата обращения 10.05.2023).

4. Ильин П. А. Основные направления применения облачных, граничных, туманных вычислений // Журнал StudNet. 2021. № 6. С. 250–257.

Статья представлена кандидатом технических наук, доцентом Зарубиным А. А.

УДК 004.896

И. А. Зайцев, Л. Е. Катеринчук, В. С. Назарук, А. В. Смирнов, М. А. Шилов (студенты гр. ИКПИ-14, СПбГУТ)

МАШИННОЕ ОБУЧЕНИЕ – КЛЮЧЕВЫЕ АЛГОРИТМЫ, СОВРЕМЕННЫЕ ИНСТРУМЕНТЫ И ВАРИАТИВНОСТЬ ПРИМЕНЕНИЯ

Машинное обучение сегодня является сферой, которая ключевым образом способствует развитию искусственного интеллекта, созданию и совершенствованию моделей, способных обучаться на основе данных без прямого программного указания. На сегодняшний день, машинное обучение применяется в разных сферах, которые включают в себя медицину, банковское дело, транспорт, производство и другие. Цель данной научной работы – описать и классифицировать ключевые алгоритмы машинного обучения, предоставить обзор современных инструментов машинного обучения, рассмотреть вариативность их применения и трудности внедрения.

машинное обучение, искусственный интеллект, ключевые алгоритмы машинного обучения, современные инструменты машинного обучения.

В первой части работы будут рассмотрены различные типы обучения, такие как обучение с учителем, обучение без учителя и обучение с подкреплением, и будет представлен обзор ключевых алгоритмов, таких как Logistic и Linear Regression, Decision Tree, Random Forest, K-Means Clustering, SVM.

Затем будут рассмотрены современные инструменты, такие как TensorFlow, Keras и PyTorch, и их особенности. В дальнейшем будут представлены различные области применения машинного обучения, такие как классификация и предсказание текста, классификация изображений, распознавание речи и рекомендательные системы. В завершении работы мы рассмотрим вариативность применения машинного обучения в разных отраслях и перспективы развития этой области в будущем.

Машинное обучение играет важную роль в современном мире, предоставляя компаниям и организациям возможность повысить эффективность работы и улучшить качество продуктов и услуг, которые они предоставляют. В этой работе мы рассмотрим ключевые аспекты машинного обучения, которые помогут дать читателю понимание об этой развивающейся и увлекательной области.

Типы обучения и ключевые алгоритмы машинного обучения

В машинном обучении существует несколько типов обучения: обучение с учителем, обучение без учителя и обучение с подкреплением. Каждый тип обучения имеет свои особенности и используется в различных областях.

Обучение с учителем – это тип обучения, в котором модель обучается на основе предоставленных ей размеченных данных, то есть данных, в которых имеются ярлыки или метки, которые указывают на правильный ответ. Алгоритмы обучения с учителем используются для решения задач классификации и регрессии. Они работают по принципу нахождения зависимостей между входными признаками и выходом, или целевой переменной.

Обучение без учителя – это тип обучения, в котором модель сама находит зависимости в данных, без явной пометки или разметки данных. Алгоритмы обучения без учителя используются для поиска скрытых признаков и структур в данных.

Обучение с подкреплением – это тип обучения, в котором модель обучается на основе взаимодействия с окружающей средой, при этом получая положительную или отрицательную обратную связь на каждом шаге. Алгоритмы обучения с подкреплением используются для принятия решений в динамических средах, где необходимо учитывать последствия совершаемых действий.

Некоторые из наиболее распространенных алгоритмов машинного обучения включают в себя:

1. Линейная регрессия (*Linear Regression*) – метод регрессионного анализа, который используется для предсказания количественной переменной на основе одного или нескольких признаков. Линейная регрессия представляет собой уравнение прямой линии, которое наилучшим образом соответствует данным. Этот алгоритм может использоваться для анализа спроса на товары или услуги, прогнозирования доходов или оценки динамики цен на рынке [1].

2. Логистическая регрессия (*Logistic Regression*) – метод классификации, используемый для предсказания двоичной переменной (например, положительный или отрицательный ответ). Этот алгоритм подходит для обработки категориальных данных в случае, когда нужно определить, принадлежит ли объект к определенному классу. Он может использоваться для анализа клиентской базы, прогнозирования вероятности дефолта или оценки рисков в инвестиционной деятельности [1].

3. Деревья решений (*Decision Trees*) – алгоритм, который выстраивает дерево на основе набора правил для принятия решений. Деревья решений могут быть использованы для выполняющих отбора признаков, определения приоритета факторов, которые влияют на результат, и прогнозирования значений переменных. Этот алгоритм может использоваться в финансовых оценках, биологии, генетике и других областях [1].

4. Случайный лес (*Random Forest*) – алгоритм машинного обучения, который используется для классификации и регрессии. Он основывается на комбинировании нескольких деревьев решений и включает в себя выбор случайного подмножества функций и данных для каждого дерева. Этот алгоритм широко используется для обработки данных в медицине, экологии и бизнесе [1].

5. Метод K-ближайших соседей (KNN) – алгоритм, который основывается на сходстве между объектами. Он используется для классификации и регрессии и определяет значение переменной на основе ближайших соседей из набора данных. Этот алгоритм может использоваться для анализа рынка недвижимости или прогнозирования доходов [2].

6. Метод опорных векторов (SVM) – алгоритм классификации, который работает путем нахождения гиперплоскости, которая максимально разделяет различные классы объектов. SVM широко используется в анализе изображений, распознавании речи и биомедицинских приложениях [3].

Каждый из этих алгоритмов имеет свои преимущества и может использоваться в определенных ситуациях для получения наилучших результатов.

Современные инструменты машинного обучения

В настоящее время TensorFlow, Keras и PyTorch являются наиболее распространенными инструментами машинного обучения. Они используются для создания, обучения и оценки различных моделей машинного обучения, включая нейронные сети [4, 5, 6].

TensorFlow – это открытая библиотека машинного обучения, которую разработал Google Brain Team. Он работает с графами вычислений, что означает, что операции и данные представлены в виде узлов и ребер графа соответственно. TensorFlow содержит множество инструментов для создания и обучения нейронных сетей, включая модули для обработки данных и оценки моделей [4].

Keras – это высокоуровневая библиотека машинного обучения, которая облегчает создание нейронных сетей. Он работает поверх TensorFlow, обеспечивая более простой и интуитивно понятный интерфейс для создания и обучения моделей [5].

PyTorch – это библиотека машинного обучения, которая работает с тензорами. Он также используется для создания и обучения нейронных сетей. PyTorch обладает более простым синтаксисом и может помочь при работе с некоторыми задачами. Он также имеет режим отладки, который позволяет легко отслеживать ошибки [6].

Scikit-learn – это инструмент машинного обучения, который предоставляет широкий спектр алгоритмов машинного обучения, таких как регрессия, классификация, кластеризация, а также многие другие. Scikit-learn также предоставляет удобный интерфейс для работы с данными, включая предобработку, нормализацию и трансформацию [6].

Примеры применения машинного обучения

Машинное обучение широко применяется в различных областях, включая медицину, бизнес, науку и технологии. Некоторые примеры применения машинного обучения включают:

– классификация и предсказание текста: машинное обучение может помочь в автоматической классификации текстовых документов и предсказании вероятности наличия определенного класса в новом документе. Примеры таких задач включают классификацию тематики текстовых документов, анализ тональности текстов и определение языка текстов [1, 2].

– классификация изображений: машинное обучение может использоваться для классификации изображений по типу объекта или сцены, определения наличия объекта на изображении и распознавания лиц на фотографиях. Эти технологии используются в системах компьютерного зрения, робототехнике и биометрических технологиях, таких как системы контроля доступа [3].

– распознавание речи: машинное обучение может использоваться для распознавания речи и текста, транскрибирования речевых записей, анализа эмоционального окраса и определения языка речи. Это приложения находят применение в голосовых помощниках, автоматических системах телефонного обслуживания, транскрипции интервью и медицинской документации [3, 7].

– рекомендательные системы: машинное обучение может использоваться для анализа и предсказания предпочтений пользователя и рекомендации наиболее подходящих товаров, услуг или контента. Примеры таких систем включают музыкальные стриминговые сервисы, онлайн-магазины и социальные сети [1, 8].

Вариативность применения машинного обучения

Машинное обучение является крайне вариативным в использовании и может быть применено в различных отраслях, от здравоохранения и финансов до сельского хозяйства и производства. Некоторые отрасли, где машинное обучение находит широкое применение, включают:

– медицина: машинное обучение может использоваться в области медицины для диагностики различных заболеваний и определения лучшего лечения. Например, модели машинного обучения могут предсказывать вероятность случаев заболевания раком на основе клинических данных пациентов [1].

– финансы: машинное обучение широко применяется в финансовой отрасли для прогнозирования рынка и решения инвестиционных решений. К примеру, использование моделей машинного обучения позволяет предсказать изменения цен на акции, проанализировать финансовые рынки и сократить финансовые риски [1].

– транспорт: машинное обучение применяется в беспилотных автомобилях для безопасности и оптимизации маршрутов. Без использования моделей машинного обучения, беспилотные автомобили не будут отличать людей от других объектов, движущихся по дороге [3].

– инфраструктура: машинное обучение может быть применено в управлении городской инфраструктурой, такой как энергетика, транспорт, водоснабжение и канализация. Применение этих технологий позволяет оптимизировать их работу, следить за состоянием инфраструктурных объектов и улучшать их эффективность [3].

– производство: машинное обучение применяется в производственной отрасли для оптимизации процессов, распознавания дефектов на производственной линии и автоматического контроля качества продукции [2].

Трудности внедрения машинного обучения

Несмотря на все перспективы и новые возможности машинного обучения, внедряя такие решения, можно столкнуться с такими трудностями как:

– недостаток качественных данных: машинное обучение требует большие объемы данных с высоким качеством. Однако сбор, очистка и обработка данных могут быть затратным и трудоемким процессом, особенно если данных очень мало [1].

– сложность интерпретации результатов: некоторые виды моделей машинного обучения (например, глубокое обучение) могут быть сложными для интерпретации. Если человек не может понять, как модель принимает решения, это может создавать проблемы с доверием к результатам [3].

– проблемы с безопасностью: машинное обучение также может подвергаться атакам злоумышленников, что может привести к сбою системы или утечки конфиденциальных данных [4].

Тем не менее, перспективы для машинного обучения более чем оптимистичны. Дальнейший прогресс в этой области может привести к созданию новых технологий и более глубокому пониманию человеческого сознания и поведения. В ближайшие годы ожидается бурный рост применения машинного обучения в различных отраслях в связи с ускоренным ростом вычислительной мощности и использованием больших данных в различных сферах.

Список использованных источников

1. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

2. Alpaydin, E. (2010). Introduction to machine learning (2nd ed.). Cambridge, Mass: MIT Press.

3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

4. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation.

5. Chollet, F. (2015). Keras. https://keras.io

6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Antiga, L. (2019). PyTorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 8024-8035.

7. Амаратунга, Дж. (2018). Машинное обучение и анализ данных. Издательский дом Вильямс.

8. Domingos, P. (2015). The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books.

Статья представлена научным руководителем, доцентом кафедры ПИВТ СПбГУТ, кандидатом технических наук Дагаевым А. В.

УДК 004.01-004.93

Д. П. Зуев, Р. Г. Шарифов (студент гр. ИКТЗ-16, СПбГУТ)

РАЗРАБОТКА ПРОГРАММЫ РАСПОЗНАВАНИЯ ПОДЛИННОСТИ РУКОПИСНОЙ ПОДПИСИ

Подпись представляет собой один из самых распространенных методов для подтверждения документов. Несмотря на широкое использование подписей, визуальное отличие между подлинной и поддельной подписью может быть достаточно сложным. Иногда необходимо автоматизировать обработку документов в различных сферах, например, в банковской или судебной, так возникает задача верификации подписей. Для решения этой задачи требуются специальные системы, способные автоматически распознавать подписи с достаточно высоким уровнем точности.

подпись, верификация, подлинность, распознавание.

Подделка рукописных подписей является серьезной проблемой, при этом почерковедческая экспертиза является длительным и дорогостоящим процессом. Использование электронных цифровых подписей (ЭЦП) в настоящее время не является широко распространенным явлением среди гражданского населения, в особенности – квалифицированных ЭЦП, в том числе на основании определенных правовых норм [1].

Аутентификация личности по рукописной подписи основывается на уникальности и стабильности характеристик процесса написания – данные характеристики используются для сравнения двух образцов подписи. С развитием технологий стало возможно количественно оценивать различные аспекты написания подписи, особенно, к примеру, при использовании графических планшетов для создания подписей, которые записывают положение конца ручки, угол наклона и силу нажатия на планшет в разные моменты времени. Полученные данные отражают динамику движений руки и становятся уникальной биометрической характеристикой каждого человека. Безусловно, подпись менее стабильна по сравнению с некоторыми другими биометрическими характеристиками, но все же верификация подписи остается актуальной, поскольку такая задача является привычным способом подтверждения документов и операций. Для распознавания подписей применяются алгоритмы, основанные на распознавании образов или математическом анализе кривых, так как подпись может быть представлена в виде набора точек.

Предварительно, в рамках настоящей работы выполнен обзор рынка программ распознавания и верификации подписи, в результате которого выявлено, что с одной стороны на рынке представлен широкий выбор программ распознавания рукописного текста (перевод в машинописный вид), к примеру, программы DocuSign и signNow предлагают различные функции, такие как подписание документов на устройствах с сенсорным экраном, распознавание подписи на бумажных документах, однако в функционале отсутствует возможность верификации. Возможность верификации подписи в общедоступных программах представлена в продукте ProgressSoft [2], при этом данное решение поддерживается только в экосистеме продуктов компании. На основании обзора принято решение о разработке кода, реализующего функционал верификации рукописной подписи, что позволит выполнить более глубокое исследование, а также может быть использовано в образовательных целях.

Изначально определены этапы работы программы распознавания подписи: сканирование подписи, обработка (составление вектора свойств), сравнение особенностей подписи с шаблоном, выдача результата сравнения.

В настоящей работе, на первом этапе исследований вопроса и соответствующих разработок, принято решение выполнение разработок с использование кода на языке программирования Python, так как данный язык программирования является популярным, а, следовательно, в достаточной мере представленным в открытых источниках, которые возможно использовать для начинающих исследователей и разработчиков, к тому же Python довольно гибкий ввиду наличия достаточно обширного количества библиотек и является подходящим для начальных разработок.

Для разработки кода использовалась библиотека cv2 – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Помимо этого, потребовалось использования библиотеки NumPy, которая имеет поддержку многомерных массивов, высокоуровневых математических функций, предназначенных для работы с многомерными массивами. Дополнительно применялась библиотека tkinter – интерфейс Python к Tk GUI toolkit, который является популярной библиотекой для разработки графических пользовательских интерфейсов (GUI) на Python. В код добавлен модуль filedialog, который импортирует модуль диалога файлов из библиотеки tkinter, который в свою очередь предоставляет функции для открытия диалоговых окон файлов и позволяет пользователю выбирать файлы (рис. 1).

```
import cv2
import numpy as np
import tkinter as tk
from tkinter import filedialog
```

Рис. 1. Фрагмент кода

Далее подробно представлены фрагменты кода, позволившие реализовать требуемый функционал. В частности, ниже представлен фрагмент (рис. 2), позволяющий выполнить открытие картинок через диалоговое окно Windows.

```
root = tk.Tk()
root.withdraw()
file_path1 = filedialog.askopenfilename()

root = tk.Tk()
root.withdraw()
file_path2 = filedialog.askopenfilename()
```

Рис. 2. Фрагмент кода

После чего, в коде используется функция imread из библиотеки OpenCV для чтения изображения по пути к файлу, указанному в file_path 1.

Аргумент 0 указывает, что изображение должно быть прочитано в оттенках серого. Затем изображение присваивается переменной signature 1 (рис. 3).

```
signature1 = cv2.imread(file_path1, 0)
signature2 = cv2.imread(file_path2, 0)
```

Рис. 3. Фрагмент кода

Также в коде используется функция threshold из библиотеки OpenCV для преобразования изображения signature 1 в двоичный файл с использованием метода порогового значения Otsu. Аргумент 0 указывает используемое пороговое значение, а аргумент 255 указывает значение, которое следует присвоить пикселям, превышающим пороговое значение. Аргумент cv2.THRESH_BINARY указывает, что изображение должно быть пороговым в двоичном режиме (рис. 4).

```
_, signature1_bin = cv2.threshold(signature1, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
_, signature2_bin = cv2.threshold(signature2, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

Рис. 4. Фрагмент кода

Помимо этого, в код добавлены функции, позволяющие рассчитывать долю ненулевых элементов в бинарном изображении signature1_bin. В качестве ненулевых элементов считаются элементы, равные единице. Отдельно добавлена функция cv2.countNonZero(), которая используется для подсчета количества ненулевых элементов в массиве. Доля ненулевых элементов рассчитывается как соотношение количества ненулевых элементов к общему числу элементов. Результат расчета записывается в переменную (рис. 5).

```
minkowski1 = cv2.countNonZero(signature1_bin) / (signature1_bin.shape[0] * signature1_bin.shape[1])
minkowski2 = cv2.countNonZero(signature2_bin) / (signature2_bin.shape[0] * signature2_bin.shape[1])
```

Рис. 5. Фрагмент кода

Затем выполнено сравнение размеров Минковского двух изображений подписи и проверяется, составляет ли разница между ними меньше 0,01. Если утверждение верно, то будет выполнен код внутри блока if (рис. 6). После чего, результат отображается в окне сообщения tkinter.

```
if abs(minkowski1 - minkowski2) < 0.01:
    result = "The signatures are likely authentic"
else:
    result = "The signatures may not be authentic"
```

Рис. 6. Фрагмент кода

На основе разработанного в рамках настоящей работы кода проведен ряд первичных экспериментов (проверены рукописные подписи, в том числе выполненные с помощью графического планшета), в результате которых код продемонстрировал свою функциональность. На рис. 7 представлен пример работы разработанной программы: слева представлен пример успешного распознания и верификации, справа – отрицательного результата.

Для дальнейших исследований следует провести дополнительную, более расширенную серию экспериментов и оценить точность распознавания статистически.

Рис. 7. Пример работы программы

Таким образом, в рамках настоящей работы проведено первоначальное исследование, а именно: выполнен обзор рынка программ, определены этапы работы программ по распознаванию и верификации подписи, разработана первая версия программы.

Разработка и широкое внедрение программ распознавания подлинности рукописной подписи является актуальным направлением, несмотря на внедрение (ЭЦП). Такие программы позволят снизить риски подделки документов, сократить время проверки подписей в случае отсутствия возможности использования ЭЦП.

Список используемых источников

1. Козлова С. Ю. Подпись электронная и рукописная: инструкция по применению // Юридический справочник руководителя. 2008. № 1. 8 с.

2. О продуктах // ProgresSoft. 2023. URL: https://www.progressoft.com/ru (дата обращения 10.05.2023).

3. Rudyi S. S., Vovk T. A, Rozhdestvensky Y. V. Signature identification by Minkowski dimension // Chaos, 2019. № 29 (5).

Статья представлена кандидатом технических наук, доцентом Зарубиным А. А.

УДК 004.4

К. А. Зырянова, И. А. Ким, С. В. Черненко
(студенты гр. ИКПИ-12, СПбГУТ)

ИССЛЕДОВАНИЕ АКТУАЛЬНОСТИ МЕДИЦИНСКИХ ПРИЛОЖЕНИЙ

Статья посвящена обзору и анализу медицинских приложений, которые становятся все более значимой в сфере здравоохранения. В данной работе рассматриваются основные категории медицинских приложений, проанализированы преимущества и ограничения каждой категории приложений, а также обсуждаются вызовы и перспективы их развития. Целью исследования является осознание роли медицинских приложений в современной медицинской практике, а также выявление и анализ влияния таких приложений на здоровье пользователей и эффективность их использования в профилактике и лечении различных заболеваний.

приложение, здоровье, медицина, здравоохранение.

Основные концепции

В медицинских приложениях существует множество различных концепций, помогающих отслеживать и анализировать здоровье человека, рассмотрим подробнее несколько из них:

Мобильное здравоохранение (mHealth). Это понятие описывает использование мобильных устройств, таких как смартфоны и планшеты, для обеспечения здравоохранения и медицинской помощи. Медицинские приложения являются ключевым компонентом мобильного здравоохранения.

Мониторинг здоровья. Многие медицинские приложения предоставляют возможность пользователю отслеживать свои показатели здоровья, такие как пульс, кровяное давление, уровень активности, сон и другие физиологические параметры. Это позволяет пользователям контролировать свое здоровье и принимать соответствующие меры для его улучшения.

Управление хроническими заболеваниями. Медицинские приложения предоставляют инструменты для управления хроническими заболеваниями, такими как диабет, астма, сердечно-сосудистые заболевания и другие. Они позволяют пользователям отслеживать симптомы, контролировать лекарственное лечение, записывать результаты анализов и совершать другие действия, необходимые для эффективного управления заболеванием.

Психологическая поддержка. В последнее время все больше медицинских приложений появляется в области психологической поддержки и улучшения психического благополучия. Они предлагают медитации, тренировки по стрессу, психотерапевтические техники и другие инструменты,

помогающие пользователям справиться с психическими проблемами и улучшить свое эмоциональное состояние.

Преимущества и ограничения использования медицинских приложений в здравоохранении

Использование экосистем медицинских приложений предоставляет ряд значительных преимуществ для пациентов, медицинского персонала и здравоохранения в целом. Некоторые из основных преимуществ включают:

● *Легкий доступ к здравоохранению.* Медицинские приложения обеспечивают удобный доступ к информации о здоровье и медицинским услугам, что позволяет пользователям получать помощь и консультацию в любое время и в любом месте.

● *Повышение осведомленности и образования.* Медицинские приложения предоставляют доступ к образовательным материалам, здоровым советам и информации о различных заболеваниях и процедурах. Это помогает людям развивать здоровый образ жизни, принимать информированные решения и принимать меры для профилактики заболеваний.

● *Персонализированная медицина.* Некоторые медицинские приложения используют технологии искусственного интеллекта и анализа данных для предоставления персонализированных рекомендаций и консультаций. Это позволяет адаптировать лечение и советы к индивидуальным потребностям и характеристикам пациента.

Конечно же также существуют ограничения использования медицинских приложений в здравоохранении:

● *Ограниченность функционала.* Некоторые медицинские приложения могут иметь ограниченный функционал и не способны полностью заменить комплексные медицинские услуги и консультации специалистов. Они могут быть полезными в дополнение к традиционной медицине, но не могут полностью заменить ее.

● *Безопасность и конфиденциальность данных.* Сбор и обработка медицинских данных в мобильных приложениях вызывают вопросы о безопасности и конфиденциальности. Недостаточная защита данных может привести к утечке чувствительной информации о здоровье пользователей.

● *Ограничения доступности и цифрового неравенства.* Некоторые медицинские приложения могут быть недоступны для определенных групп населения, включая пожилых людей, малообеспеченных или тех, кто не имеет доступа к соответствующей технологии. Это создает цифровое неравенство в доступе к здравоохранению.

Необходимо принимать во внимание эти преимущества и ограничения при использовании медицинских приложений, чтобы обеспечить их безопасность, эффективность и соответствие потребностям пользователей.

Разнообразие медицинских приложений, доступных на рынке, включая приложения для разных платформ

На данный момент доступны мобильные приложения для iOS и Android, а также веб-приложения.

Одними из наиболее популярных медицинских приложений являются те, которые помогают контролировать состояние здоровья, например, приложения для отслеживания показателей кровяного давления, уровня глюкозы в крови, веса и физической активности. Также популярными являются медицинские справочники и диагностические приложения, которые помогают пользователю самостоятельно определить свое заболевание или найти информацию о лекарствах, например, Яндекс Здоровье, Доктор онлайн, Моя аптека, Water time, Lifesum.

Эффективность и безопасность медицинских приложений

Оценка эффективности и безопасности медицинских приложений играет важную роль в обеспечении качества предоставляемых ими услуг и защите здоровья пациентов. Рассмотрим некоторые способы оценки эффективности и безопасности медицинских приложений:

1. Оценка соответствия стандартам: медицинские приложения должны соответствовать определенным стандартам качества и безопасности, установленным национальными и международными организациями.

2. Клинические испытания: проведение клинических испытаний позволяет оценить эффективность и безопасность медицинского приложения с помощью контролируемых исследований.

3. Анализ отзывов пользователей: анализ отзывов пользователей медицинского приложения может дать представление о его эффективности и безопасности.

4. Мониторинг и контроль за использованием: мониторинг и контроль за использованием медицинского приложения позволяет оценить его эффективность и безопасность в реальной жизни и быстро выявлять проблемы.

5. Система обратной связи: система обратной связи между пользователями медицинского приложения и разработчиками позволяет получать информацию об эффективности и безопасности приложения, а также быстро устранять выявленные проблемы.

6. Обновления и исправления ошибок: обновление медицинского приложения и исправление ошибок может улучшить его эффективность и безопасность.

Оценка качества и достоверности данных, собранных медицинскими приложениями. Нормативные и правовые аспекты, связанные с использованием медицинских приложений в России

Оценка качества и достоверности данных, собранных медицинскими приложениями, является важным вопросом для обеспечения безопасности пациентов и правильности диагностики и лечения. Следует понимать, что оценка качества данных зависит от многих факторов, включая тип приложения, способ сбора данных, а также компетентность пользователя.

Первым шагом в оценке качества данных является проверка надежности и безопасности приложения, используя соответствующие сертификационные программы и стандарты. В РФ существует ряд нормативных документов и стандартов, которые регулируют работу медицинских приложений и обеспечивают безопасность и качество данных. Также существуют сертификационные программы, направленные на проверку соответствия медицинских приложений нормам и требованиям, такие как «Надежное ПО» и «Цифровая медицина».

Однако, несмотря на наличие нормативных документов и сертификационных программ, оценка качества и достоверности данных, собранных медицинскими приложениями в РФ, является сложной задачей, так как работа медицинских приложений не регулируется едиными стандартами и может различаться в зависимости от типа приложения и способа сбора данных.

Далее следует оценить, насколько точно и полно приложение собирает данные, а также наличие ошибок и неточностей в сборе этих данных. Для этого можно провести тестирование приложения на различных группах пользователей, а также сравнить полученные данные с другими источниками информации.

Также необходимо учитывать, что медицинские приложения часто собирают персональную информацию о пользователях, поэтому важно оценить соответствие приложения применимым нормам и правилам обработки и защиты персональных данных. Например, Федеральный закон «О персональных данных» и ГОСТ Р ИСО/МЭК 27001-2013 «Информационная технология. Методы обеспечения безопасности. Системы менеджмента информационной безопасности». Требования устанавливают правила обработки персональных данных и требования к системам менеджмента информационной безопасности.

Наконец, следует учитывать, что даже если данные, собранные медицинскими приложениями, кажутся достоверными на первый взгляд, всегда необходимо подтверждение специалиста или проведение соответствующих медицинских исследований для подтверждения диагноза и выбора оптимального лечения.

Риски и вызовы, связанные с использованием медицинских приложений, и предложить рекомендации для повышения их безопасности и эффективности

Одним из самых больших рисков, связанных с медицинскими приложениями, является безопасность и конфиденциальность данных. Поскольку в этих приложениях хранится все больше личной информации, важно, чтобы в них были предусмотрены надежные меры безопасности для защиты данных пользователя от доступа посторонних лиц или организаций. Кроме того, пользователи должны знать, как их личная информация используется и передается поставщиком приложения.

Еще один риск связан с точностью и надежностью медицинской информации, предоставляемой через эти приложения. Неточные или неполные данные могут привести к неправильным диагнозам или лечению, что может подвергнуть пациентов риску серьезных осложнений здоровья. Очень важно регулярное обновление продукта точной и актуальной информацией, чтобы пользователи могли доверять советам, которые они получают от него.

Наконец, пользователям важно понимать, что функции приложения не должны заменять профессиональную помощь врача или медицинского работника в случае необходимости.

Выводы

В заключении, можно сказать, что медицинские приложения являются одним из самых актуальных направлений в сфере здравоохранения. Они позволяют контролировать собственное здоровье и своевременно получать медицинскую помощь. Кроме того, медицинские приложения существенно упрощают работу врачей и помогают им внедрять новые методы диагностики и лечения.

Несмотря на то, что медицинские приложения имеют ряд преимуществ, нужно отдавать отчет возможным рискам и ограничениям, которые могут появиться в процессе их использования. Важно правильно подходить к выбору приложения и следить за тем, чтобы оно соответствовало всем медицинским стандартам и требованиям.

Итак, медицинские приложения играют важную роль в нашей жизни. Они помогают сохранять здоровье, облегчают работу врачей и улучшают качество медицинской помощи. Естественно, они не смогут заменить настоящего врача, но косвенное участие в процессе лечения они оказывают немаловажное.

Список используемых источников

1. Фокин С. Ю., Киричек Р. В. Обзор медицинских приложений, устройств и технологий связи интернета вещей // Информационные технологии и телекоммуникации. 2016. Т. 4. №. 4. С. 67.

2. Кудратиллаев М. Б. У. Мобильные медицинские приложения: значение в жизни человека и их особенности, обзор существующих приложении // Science and Education. 2023. Т. 4. №. 5. С. 803–811.

3. Правовой фонд правовых и нормативно-технических документов. URL: https://docs.cntd.ru/document/456027168

4. Ким О. Т., Дадаева В. А., Тельхигова А. А., Драпкина О. М. Мобильные медицинские приложения: возможности, проблемы и перспективы // Profilakticheskaya Meditsina. 2021. Т. 24. №. 7. С. 96–102.

Статья представлена научным руководителем,
ассистентом кафедры ИКС СПбГУТ Редругиной Н. М.

УДК 004.432.42

Ю. И. Кокин, М. В. Мартынов, И. С. Тарасов
(студенты гр. ИКПИ-14, СПбГУТ)
М. С. Суменков (студент гр. ИКПИ-12, СПбГУТ)

HASKELL – ПОТЕНЦИАЛ И ВЛИЯНИЕ НА СОВРЕМЕННОЕ ПРОГРАММИРОВАНИЕ

В статье рассматривается влияние и область применения языка Haskell. Haskell – функциональный язык программирования, обеспечивающий высокую безопасность и надежность кода благодаря ряду особенностей языка. Среди особенностей выделяется строгая типизация, позволяющая обнаружить большинство ошибок уже во время компиляции, наличие механизмов строгой проверки использования памяти, функциональный стиль программирования, предотвращающий изменение состояния объектов. Высокая производительность языка объясняется использованием ленивого вычисления. Haskell делает акцент на использовании «чистых» функций, которые всегда дают один и тот же результат для одних и тех же входных данных. Поэтому, Haskell прекрасно подходит для написания смарт-контрактов, которые могут быть реализованы в точном, формально проверенном коде, обеспечивающем высокий уровень уверенности.

язык программирования Haskell, безопасность, компиляция, ленивые вычисления.

Для того, чтобы оценить степень влияния и обозначить области применения Haskell, перейдем к рассмотрению устройства языка и его особенностей.

Основные принципы языка Haskell

Списки

List – контейнерный тип, который может содержать себе значения любого типа, представленного в языке Haskell, включая другие списочные

структуры. Все значения списка должны принадлежать одному типу. Для обозначения списков используется нотация вида: `[a1, a2, ..., an]` и определены следующие операции:

1. Конструктор (:) (двоеточие).
2. Получение головы списка head.
3. Получение хвоста списка tail.

Все три операции взаимосвязаны между собой следующим образом:

1. head (x:y) = x.
2. tail (x:y) = y.
3. (head (x:y)) : (tail (x:y)) = (x:y).

Функции

Общая нотация функции в языке Haskell выглядит так:

<имя_функции> <образец> «=» <выражение>

В Haskell используется технология клозов (от англ. *clause*), которые представляют собой варианты вычисления функции, для организации ветвления алгоритмов. Для каждого клоза указан образец (от англ. *pattern*) – выражение, используемое в определении функций для сопоставления с данными.

Функция last, которая возвращает последний элемент заданного списка:

```
last [] = error "Функция last не~может обработать пустой список."
last [x] = x
last (x:xs) = last xs
```

Haskell, как и многие другие функциональные языки программирования, основан на лямбда-исчислении (формальная система, разработанная американским математиком Алонзо Чёрчем). В лямбда-исчислении программа называется термом. В этой системе есть всего три правила для написания термов:

1. Переменные x, y, z и т. д. являются термами и составляют алфавит языка.
2. Если M и N являются термами, то (MN) также является термом.
3. Если x – переменная, а M – терм, то ($\lambda x.M$) также является термом.

Функцию ($\lambda x.(\lambda y.x)$) в языке Haskell можно записать как \$x$ – > (\$y$ – > x). Она принимает переменную x и возвращает другую функцию ($\lambda y.x$), которая, в свою очередь, принимает аргумент y и возвращает x.

Типы

Система типов в языке Haskell является полной, сильной и статической. Используется автоматический вывод типов. Базовыми типами являются

Char, Double, Float, Int, Integer. Остальные типы можно получить из них при помощи операций конструирования типов.

Наибольший интерес представляет функциональный тип. В Haskell любая функция является функцией одного параметра. Функции как бы нескольких параметров принимают первый аргумент и возвращают другую функцию, принимающую следующий аргумент и так далее, пока не будет достигнута финальная функция, возвращающая значение не функционального типа. Этот процесс называется каррированием (в честь математика Хаскелла Карри). Если для функции указана такая сигнатура: someFunction :: Integer – > Integer – > Double, то Haskell воспринимает её следующим образом: func :: Integer – > (Integer – > Double).

Классы

В языке Haskell общий вид определения классов выглядит так:

```
class Eq a where
 (==) :: a -> a -> Bool
```

Haskell поддерживает наследование. Конструкции языка также позволяют реализовать полиморфизм методов класса и инкапсуляцию данных.

Монады

Чтобы поддерживать чистоту функций, допуская выполнения операций, которые могут содержать в себе побочные эффекты, такие как ввод/вывод, отправка данных на сервер, в Haskell использует специальный механизм, называемый монадами. Монады представляют программную концепцию, которая обеспечивает выполнение последовательных действий с передачей результатов от одного действия к другому. Наибольшую популярность приобрела do-нотация, которая позволяет последовательно комбинировать вычисления, вызывая функции, а также извлекая необходимые значения из монад:

```
main = do
    putStrLn "Введите ваше имя: "
    name <- getLine
    putStrLn $ "Привет, " ++ name ++ "!"
```

Отличительные особенности функционального языка Haskell

1. Краткость и простота – простой синтаксис, декларативный стиль написания кода. Пример реализации быстрой сортировки Хоара:

```
quicksort [] = []
quicksort (x:xs) = lower ++ [x] ++ upper
      where lower = quicksort(filter (<= x) xs)
      upper = quicksort(filter (> x) xs)
```

2. Строгая типизация – позволяет свести к минимум ошибки еще на этапе компиляции.

3. Модульность – разбиение программы на независимые модули.

4. Функции – это значения и объекты вычисления (*first-class citizen*).

5. Чистота – отсутствие побочных эффектов и детерминированность.

6. Ленивые вычисления – вычисления выполняются только, когда нужен их результат. Это повышает производительность.

Потенциал и влияние

Строгая типизация, чистота функций, ленивые вычисления – всё это позволяет разработчикам на Haskell писать безопасный и производительный код. Поэтому Haskell применим в тех областях, где каждая ошибка может повлечь за собой колоссальные репутационные и финансовые потери. Например, разработчики Cardano (многоуровневая блокчейн-платформа на базе *Proof-Of-Stake*) в качестве языка для написания и тестирования смарт-контрактов на платформе Plutus Playground выбрали Haskell. Своеобразный “hello world” на языке смарт-контрактов, который создаёт контракт, логирующий сообщение:

```
hello :: Contract BlockchainActions T.Text ()
hello = logInfo @String “Hello world”
endpoints :: Contract BlockchainActions T.Text ()
endpoints = hello;
mkSchemaDefinitions ‘’BlockchainActions
$(mkKnownCurrencies [])
```

Haskell и в целом функциональный подход к программированию оказывает влияние на развитие мультипарадигменных языков. Разработчики этих языков всё чаще добавляют новые языковые конструкции, основываясь на паттернах функционального программирования. Они позволяют писать более декларативный код, отказываясь от императивных инструкций. Например, конструкция async/await, реализованная в таких языках как JavaScript, C#, Kotlin и упрощающая работу с асинхронным кодом, своим происхождениям обязана монадам и do-нотации. В самом деле, let *x* = await asyncFn() в JavaScript, ничто иное, как *x* <– asyncFn() в Haskell.

Широкое применения также нашли алгебраические типы данных – это такие типы данных, которые являются композицией других типов. Пример: data Bool = True | False. Такую конструкцию в Typescript можно записать как: type Bool = true | false.

Использование ленивых или же отложенных вычислений в Haskell, лежит в основе генераторов в Python:

```
-- Haskell. Определяем бесконечный список чисел Фибоначчи, используя лени-
вые вычисления
fib :: [Integer]
fib = 0 : 1 : zipWith (+) fib (tail fib)
```

```
#Python. Проделываем тоже самое, но уже при помощи генератора, выдающего
по вызову следующее в последовательности число Фибоначчи.
def fib_gen():
    a, b = 0, 1
    while True:
        yield a
        a, b = b, a + b
```

Таким образом, было показано, что, используя все вышеперечисленные преимущества языка, разделяя побочные эффекты от основной логики программы везде, где это возможно, программисты на Haskell могут сделать свой код более простым для анализа и понимания, безопасным и предсказуемым. Haskell используется для создания сложных систем, включая проекты по анализу больших данных, криптографии и блокчейн-разработке, а его инновационные языковые конструкции оказали влияние на развитие других языков.

Список используемых источников

1. Душкин Р. В. Функциональное программирование на языке Haskell. М. : ДМК Пресс, 2007. 608 с., ил. ISBN 5-94074-335-8.
2. Какой вклад внесло функциональное программирование в современные языки? [Электронный ресурс]. URL: https://anton-k.github.io/ru-haskell-book/book/home.html
3. Учебник по Haskell [Электронный ресурс]. URL: https://anton-k.github.io/ru-haskell-book/book/home.html

Статья представлена научным руководителем, профессором кафедры ПИВТ СПбГУТ, кандидатом технических наук, доцентом кафедры Бирюковым М. А.

УДК 621.391.64

М. В. Комар (студент гр. ИКТО-91, СПбГУТ)

ПРИМЕНЕНИЕ ВОЛОКОННО-ОПТИЧЕСКИХ ТЕХНОЛОГИЙ В СИСТЕМАХ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ

В настоящее время вопросы экологии приобретают все более важное значение. В работе рассмотрены вопросы применения волоконно-оптических технологий в экологическом мониторинге промышленных предприятий. Ввиду острой проблемы загрязнения окружающей среды тема доклада является весьма актуальной и отражает современное состояние вопросов контроля загрязнения атмосферы вокруг экологически опасных объектов.

экология, волоконно-оптические технологии, загрязнение атмосферы, контроль состояния чистоты воздуха, волоконно-оптический датчик, оптическое волокно.

Введение

Современный экологический мониторинг в сочетании с волоконно-оптическими технологиями позволяет человеку вести удаленный, беспрерывный контроль для обеспечения безопасности людей и окружающей среды на территории промышленных предприятий. В промышленных отраслях требуется применение волоконно-оптических газоанализаторов, объединенных в оптические системы, для контролирования загазованности атмосферного воздуха. Актуальность проблемы экологического контроля объясняется наличием метана (CH_4) на производственных участках, таких как газопровод, газохранилище, дымовая труба. Опасность газа в воздушной среде заключается в том, что метан токсичен для рабочего персонала и взрывоопасен.

Целесообразно выбрать и использовать оптические газоанализаторы, основанные по принципу спектрального оптического поглощения, которые имеют преимущество перед использованием традиционных термокаталитических и электрохимических методов исполнения приборов газоанализа. Они способы улавливать даже малейшие колебания паров, газов в помещении и на территории промышленного объекта, имеют высокое быстродействие, возможность контроля одного или сразу нескольких газов, достаточную степень защиты от повреждений и воздействия внешних факторов, а также не поддаются электромагнитным помехам.

В статье приведены существующие системы и технологии, методика расчетов основных параметров для простой схемы волоконно-оптических датчиков, которая может применяться на практике.

Анализатор концентрации метана

Разработан датчик, в задачи которого входит удаленный и непрерывный контроль концентрации метана в атмосферном воздухе на объектах предприятия с газовым оборудованием [1]. На рис. 1 изображена схема работы анализатора.

Принцип работы.

На необходимой локации промышленного предприятия для экологического контроля устанавливаются датчики обнаружения метана. По оптическому волокну световые сигналы посылаются к установленным датчикам.

В датчиках находятся микроконцентрации метана, которые при частичном поглощении света преобразовывают возвращающийся световой импульс.

Оператор удаленно анализирует информацию с датчиков с высокой скоростью обновления.

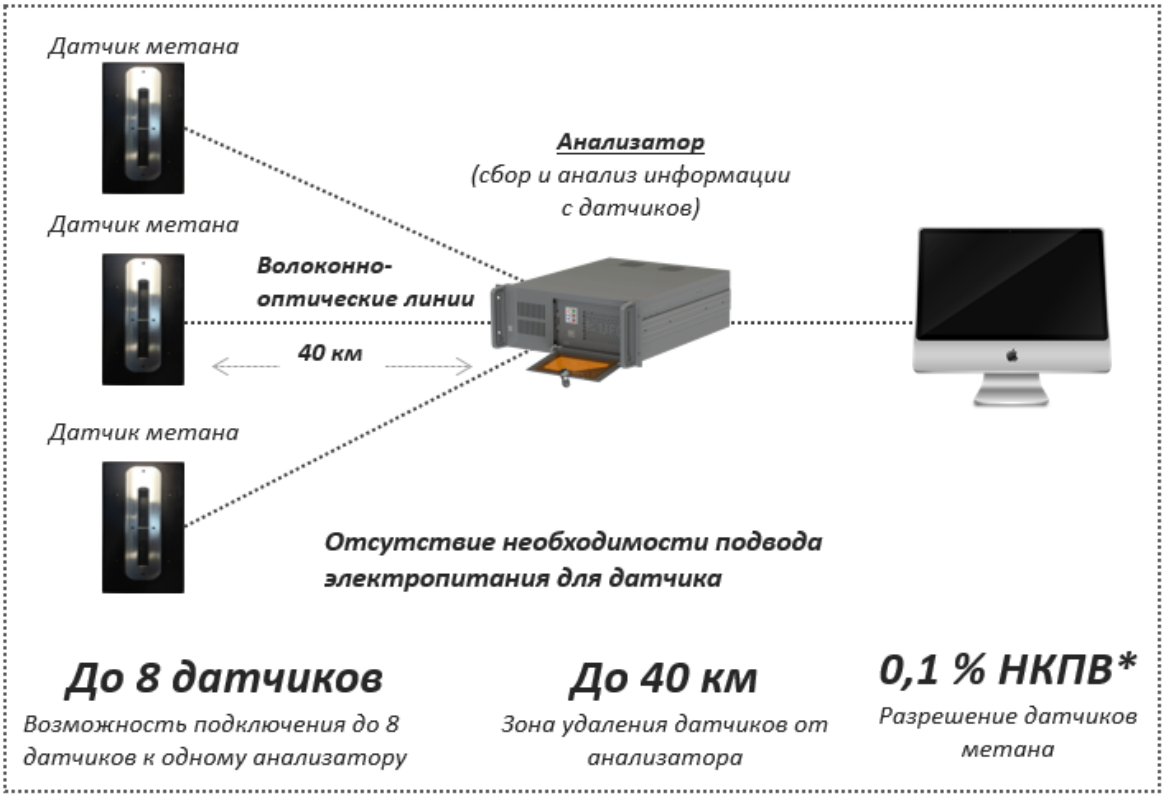


Рис. 1. Работа анализатора

Характеристики:

– контроль оптического поглощения – 1650 нм;

– оповещение о достижении концентрации метана в атмосферном воздухе – 0,1 % нижнего предела взрывной концентрации;

– удаленность датчика на расстояние до 40 км;

– количество датчиков – 4/8;

– период получение обновленной информации – 30/70 с;

– сенсор – многопроходная проточная кювета.

Детектор метана в составе модернизированной волоконно-оптической системы мониторинга протяженных объектов

На рис. 2 (см. ниже) представлен волоконно-оптический детектор метана (ВОДМ).

Лазер генерирует в ближнем инфракрасном диапазоне (1651 нм) излучение мощностью более 10 мВт. Аналитическая кювета (12) с волоконным входом и выходом имеет длину оптического пути 5 см с суммарными потерями менее 1 дБ. Кювета выполняется в форме скобы с волоконно-оптическими коллиматорами на торцах. Детекторы аналитического сигнала (10) и сигналы сравнения (9) – это $InGaAs\ pin$ фотодиоды с диаметром активной площадки 0,2 см [2].

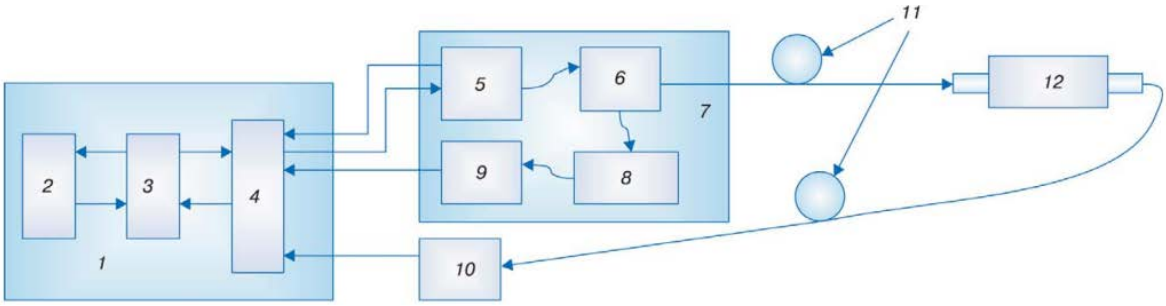


Рис. 2. Принципиальная схема волоконно-оптического детектора:
1 – блок управления и приема данных, 2 – цифровой программируемый модуль,
3 – модуль цифроаналоговых и аналогово-цифровых преобразователей (ЦАП и АЦП),
4 – модуль преобразователей аналоговых сигналов, 5 – модуль диодного лазера,
6 – волоконный разветвитель, 7 – блок лазерного излучения, 8 – кювета сравнения,
9 – детектор сигнала сравнения, 10 – детекторы аналитического сигнала,
11 – волоконно-оптический кабель, 12 – аналитическая кювета

Технические характеристики: удаленность датчика на расстояние до 50 км, чувствительность при обнаружении с выносной кюветой 5 см равняется 0,01 объемных % при времени регистрации не более 30 с.

Методика расчета характеристик для простой схемы волоконно-оптического датчика

Принцип работы рассматриваемого на рис.3 ВОД концентрации метана основан на дифференциальном оптическом поглощении [3].

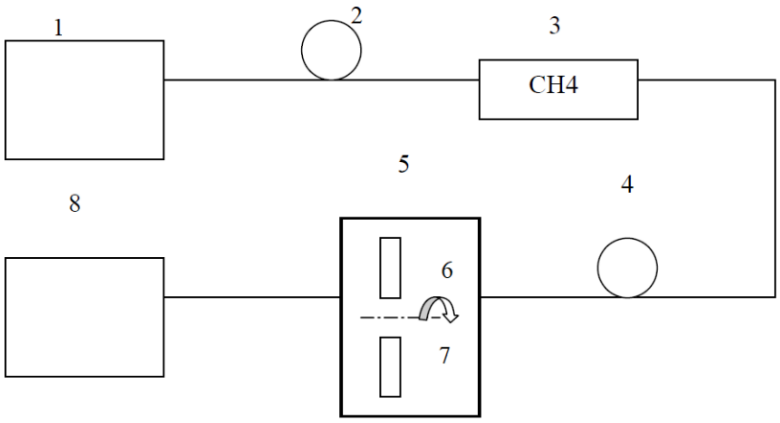


Рис. 3. Функциональная схема волоконно-оптического датчика для дистанционного измерения концентрации метана: 1 – диод светоизлучающий, 2 – входное оптическое волокно, 3 – измерительная газовая ячейка, 4 – выходное оптическое волокно, 5 –модулятор, 6 – оптический фильтр, выделяющий сигнальную (рабочую) длину волны, 7 – оптический фильтр, выделяющий опорную длину волны, 8 – фотоприемное устройство

Формула (1) для расчета минимальной обнаружимой концентрации метана, которая определяется уровнем шума приемной системы Pn и соответствует условию $Pr(\lambda_2) - Pr(\lambda_1) = Pn$.

$$N_{\min} = \frac{P_n}{[\sigma(\lambda_1)-\sigma(\lambda_2)]l\mathrm{Pr}(\lambda_2)},$$

где λ_1 – длина волны, соответствующая линии поглощения метана (рабочая длина волны), λ_2 – опорная длина волны, лежащая вне спектра поглощения, l – длина измерительной ячейки с газом, $Pr(\lambda_2)$ – мощность, поступающая на фотоприемник, $\sigma(\lambda_i)$ – сечение поглощения метана на длине волны λ_i.

Длина волоконного тракта рассчитывается по формуле:

$$L = \frac{(P_{\text{ис}}-P_r-\mathrm{A}_{\text{яч}}-\mathrm{A}_{\text{мод}}-\mathrm{A}_{\text{иф}})}{\alpha},$$

где $P_{\text{ис}}$ – мощность, введенная от источника во входное оптическое волокно, дБм, P_r – величина оптической мощности, поступающая на фотоприемное устройство, $\mathrm{A}_{\text{яч}}$ – потери в газовой ячейке, $\mathrm{A}_{\text{мод}}$ – потери в модуляторе, $\mathrm{A}_{\text{иф}}$ – потери в интерференционном фильтре.

Результаты расчетов показывают, что ВОД концентрации метана, работающий на линии поглощения 1332 нм, регистрирует концентрации метана на уровне 0,1 % при длине волоконного тракта около 1 км.

Улучшить характеристики ВОД можно, если перейти на линию поглощения 1667 нм и использовать полупроводниковые источники излучения с мощностью не менее 100 мкВт и фотодиоды с темновыми токами менее 1 нА. Эти изменения повысят значение регистрации концентрации до 0,05 % при длине волоконного тракта около 10 км [3].

Заключение

Выбросы газов достигают нескольких млрд кубических метров ежегодно, что приводит к серьезным экологическим последствиям. Для проведения безопасных работ на промышленных предприятиях необходим регулярный мониторинг объектов с целью своевременного обнаружения утечек газа.

Для решения таких целей необходимо использовать современные волоконно-оптические технологии, которые в сравнении с другими традиционными технологиями экологического мониторинга обладают высокой точностью, стабильностью, защищенностью и удаленной на большие расстояния непрерывной работой.

Список используемых источников

1. Анализатор концентрации метана ВОЛНА-АКМ. URL: http://ufdcom.ru/volna-akm (дата обращения 06.05.2023).

2. Плешков Д. И., Турбин А. И. Детектор метана как элемент системы мониторинга протяженных объектов (СМПО-М) компании «ПетроЛайт» // Русский журнал. 2015. № 1. С. 38–44.

3. Миронов С. А. Волоконно-оптический датчик концентрации метана: расчет основных характеристик // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2007. № 44. С. 72–78.

Статья представлена научным руководителем,
старшим преподавателем кафедры ФиЛС СПбГУТ Сергеевым А. Н.

УДК 004.855.3

И. В. Краснорудский, Р. М. Николаев, А. С. Сумин
(студенты гр. ИКПИ-13, СПбГУТ)

МАШИННОЕ ОБУЧЕНИЕ – КЛЮЧЕВЫЕ АЛГОРИТМЫ, СОВРЕМЕННЫЕ ИНСТРУМЕНТЫ И ВАРИАТИВНОСТЬ ПРИМЕНЕНИЯ

Научно-исследовательская работа посвящена изучению основных алгоритмов машинного обучения и современных инструментов, применяемых в данной области. В работе осуществляется исследование различных алгоритмов, а также анализируются их применение и эффективность.

машинное обучение, ключевые алгоритмы, вариативность применения.

Машинное обучение – это наука о разработке алгоритмов и статистических моделей, которые компьютерные системы используют для выполнения задач без явных инструкций, полагаясь вместо этого на шаблоны и логические выводы [1–3].

Ключевые алгоритмы (рис. 1, см. ниже):

1. Решающие деревья.
2. Логистическая регрессия.
3. Метод опорных векторов.
4. Нейронные сети.
5. Алгоритм *k*-ближайших соседей.
6. Случайный лес.

Ключевые алгоритмы машинного обучения помогают автоматизировать и улучшить процессы принятия решений, оптимизировать бизнес-процессы, повышать точность прогнозов и улучшать пользовательский опыт. Они позволяют извлекать ценную информацию из больших объемов данных, обнаруживать скрытые закономерности и паттерны, делать предсказания на основе имеющихся данных и принимать обоснованные решения. Ключевые алгоритмы машинного обучения становятся неотъемлемой

частью современных технологий и инструментов, обеспечивая новые возможности и преимущества в различных областях, таких как бизнес, здравоохранение, финансы, маркетинг, транспорт и многих других.

Общий принцип работы алгоритмов машинного обучения заключается в обнаружении паттернов и связей в данных и использовании их для принятия решений или предсказания результатов в новых ситуациях.

Рис. 1. Разновидности машинного обучения

Ознакомление с современными инструментами и фреймворками, используемыми в области машинного обучения, является важной частью работы и исследований в этой области (рис. 2, см. ниже). Вот несколько из них:

1. Python. Язык программирования Python широко используется в области машинного обучения благодаря своей простоте, мощным библиотекам и широкой поддержке сообщества разработчиков.

2. NumPy, Pandas. Библиотеки предоставляют высокоэффективные структуры данных и операции для работы с многомерными массивами чисел. Они являются основой для многих других библиотек машинного обучения.

3. Scikit-learn является одной из наиболее популярных библиотек машинного обучения. Она предлагает широкий спектр алгоритмов для классификации, регрессии, кластеризации, а также инструменты для предобработки данных и оценки моделей.

4. TensorFlow – это платформа глубокого обучения, разработанная компанией Google. Он позволяет создавать и обучать нейронные сети с использованием графовых вычислений и обеспечивает высокую производительность.

5. PyTorch – это фреймворк для глубокого обучения, разработанный компанией Facebook. Он предлагает динамическое определение графов вычислений, удобные инструменты для построения и обучения нейронных сетей, а также поддержку GPU-ускорения.

6. Keras – это высокоуровневый фреймворк для глубокого обучения, который обеспечивает простоту использования и быструю разработку моделей. Он может работать поверх TensorFlow, Theano или других библиотек глубокого обучения.

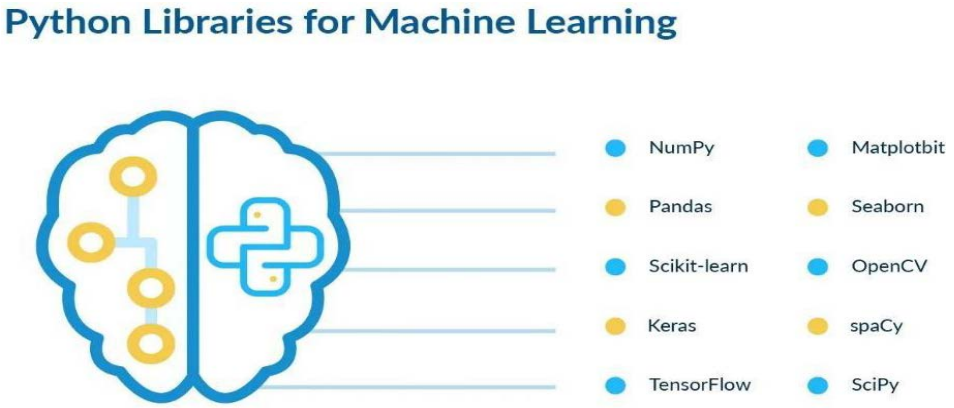


Рис. 2. Разновидности фреймворков и инструментов для разработки.

Эти инструменты и фреймворки предоставляют мощные возможности для разработки, обучения и применения моделей машинного обучения в различных областях и приложениях. Изучение и использование этих инструментов и фреймворков помогает исследователям и разработчикам эффективно работать с данными, реализовывать и тестировать различные алгоритмы машинного обучения, а также создавать интеллектуальные системы и приложения.

Анализ вариативности применения машинного обучения в различных областях, включая обработку естественного языка, распознавание образов, рекомендательные системы и анализ данных.

Вариативность применения машинного обучения в различных областях демонстрирует его широкий спектр возможностей (рис. 3). Ниже представлен анализ вариативности применения машинного обучения в нескольких областях:

1. **Обработка естественного языка** (*Natural Language Processing, NLP*): машинное обучение используется для анализа и обработки текстовых данных. Применение включает распознавание речи, машинный перевод,

суммаризацию текста, классификацию текстовых документов, анализ тональности и другие задачи, связанные с обработкой текста.

2. **Распознавание образов** (*Computer Vision*): машинное обучение применяется для анализа и обработки изображений и видео. Включает в себя задачи: распознавание лиц, классификация объектов, сегментация изображений, детектирование объектов, анализ движения, связанные с обработкой визуальных данных.

3. **Рекомендательные системы**: машинное обучение используется для создания персонализированных рекомендаций для пользователей. Применение включает рекомендации товаров, фильмов, музыки, новостей и других контентов на основе анализа предпочтений и поведения пользователей.

4. **Анализ данных**: машинное обучение применяется для извлечения ценной информации из больших объемов данных. Включает в себя задачи, такие как кластеризация, классификация, прогнозирование данных и анализ временных рядов.

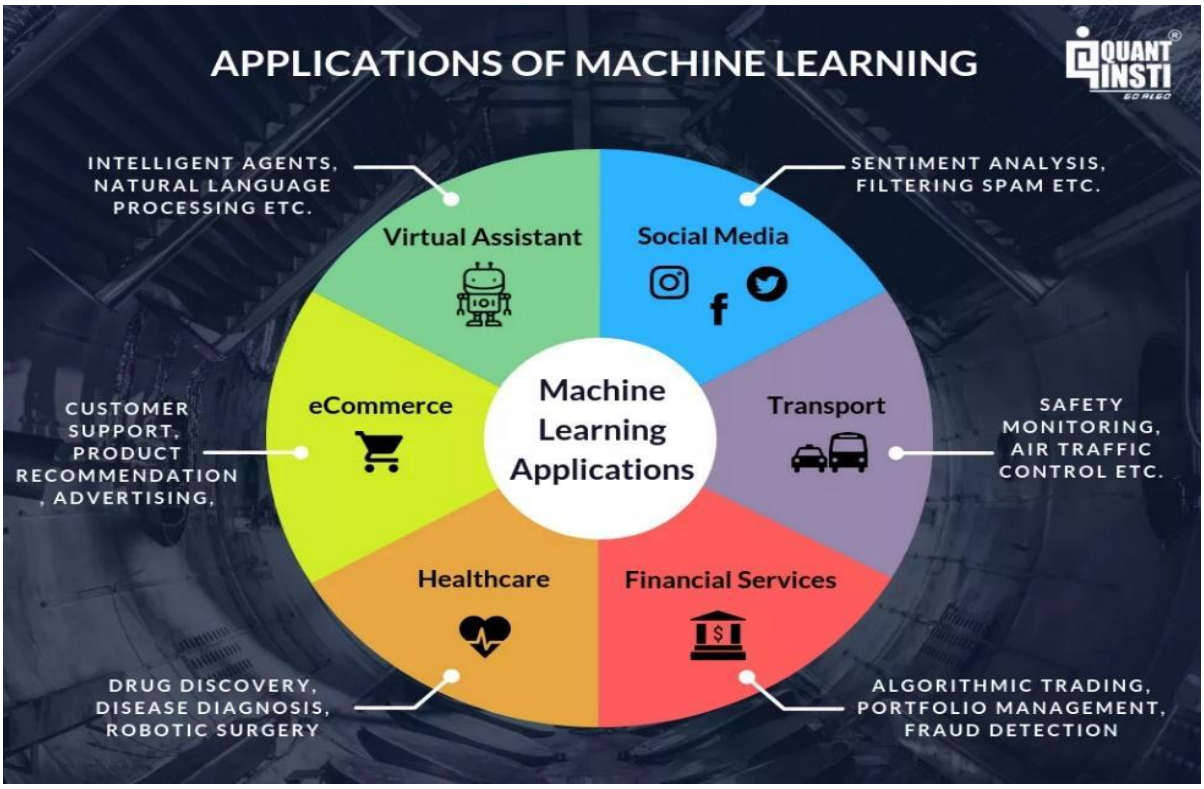


Рис. 3. Вариативности применения машинного обучения

В итоге данной научно-исследовательской работы были получены следующие результаты:

1. Обзор основных алгоритмов машинного обучения, их принципы работы и области применения.

2. Ознакомление с современными инструментами и фреймворками, используемыми в области машинного обучения, включая язык программирования Python и библиотеки TensorFlow, PyTorch и Scikit-learn.

3. Анализ вариативности применения машинного обучения в различных областях, включая обработку естественного языка, распознавание образов, рекомендательные системы и анализ данных.

Список используемых источников

1. Васильев А. Н., Тархов Д. А. Нейросетевое моделирование. Принципы. Алгоритмы. Приложения. СПб. : Изд-во Политехн. Ун-та, 2009. ISBN 978-5-7422-2272-9 URL: https://lib-bkm.ru/13769 (дата обращения 18.05.2023).

2. Галушкин А. И. Нейронные сети. Основы теории. М. : Горячая линия – Телеком, 2010. URL: https://djvu.online/file/QVrG3lB5HBs1N (дата обращения 18.05.2023).

3. Хайкин С. Нейронные сети: полный курс. 2-е изд. М. : Вильямс, 2006. URL: https://djvu.online/file/4gkGpeE96OH9Vcom (дата обращения 05.05.2023).

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.522-004.934

Д. И. Кузин (студент гр. ИКТЗ-15, СПбГУТ)

ГОЛОСОВЫЕ ПОМОЩНИКИ: ВОПРОСЫ БЕЗОПАСНОСТИ, КОНФИДЕНЦИАЛЬНОСТИ И ПУТИ РЕШЕНИЯ

Развитие технологий и широкое распространение устройств, поддерживающих голосовое управление, обеспечили популярность голосовых помощников у пользователей. Актуальными остаются вопросами конфиденциальности и безопасности использования такого типа устройств. В настоящей работе перечислены основные вопросы конфиденциальности и безопасности, а также угрозы, возникающие в результате использования голосовых помощников. Проведен анализ возможных решений приведенных проблем.

голосовые помощники, конфиденциальность, безопасность, пользователь, угрозы, нарушители

Голосовые помощники – приложения, которые основываются на технологии голосового управления, и позволяют пользователю общаться с компьютером или другим техническим устройством, используя естественный голос. Голосовые помощники могут выполнять различные задачи, такие как поиск информации в Интернете, отправка сообщений, исполнение музыки, установка напоминаний и другие задачи, просто по команде голосом. Наиболее распространенные голосовые помощники – *Siri* для *Apple*, *Google Assistant* для *Android*, *Alexa* для *Amazon*, и *Cortana* для *Microsoft*.

Однако согласно проведенным исследованиям более чем у 70 % рядовых пользователей возникают вопросы, связанные с тем, насколько использование подобных приложений безопасно и как, в противном случае, обеспечить безопасность и конфиденциальность при использовании голосовых помощников.

Для того, чтобы определить, какие проблемы безопасности возникнуть при использовании голосового помощника, необходимо в первую очередь рассмотреть алгоритм работы таких приложений.

Процесс функционирования голосового помощника основывается на распознавании голоса, который активируется любой фразой, содержащей название голосового помощника и запрос. В свою очередь, запрос записывается и распознается. Далее подается команда на выполнение того или иного действия. Если для выполнения команды пользователя потребуется подключать какие-либо интернет-ресурсы, то запрос поступает на обслуживающей сервер и обрабатывается. В случае выполнения команд пользователя, не требующих задействования дополнительных ресурсов, такие команды выполняются непосредственно устройством [1] Данный процесс продемонстрирован на разработанной в рамках работы схеме (рис. 1).

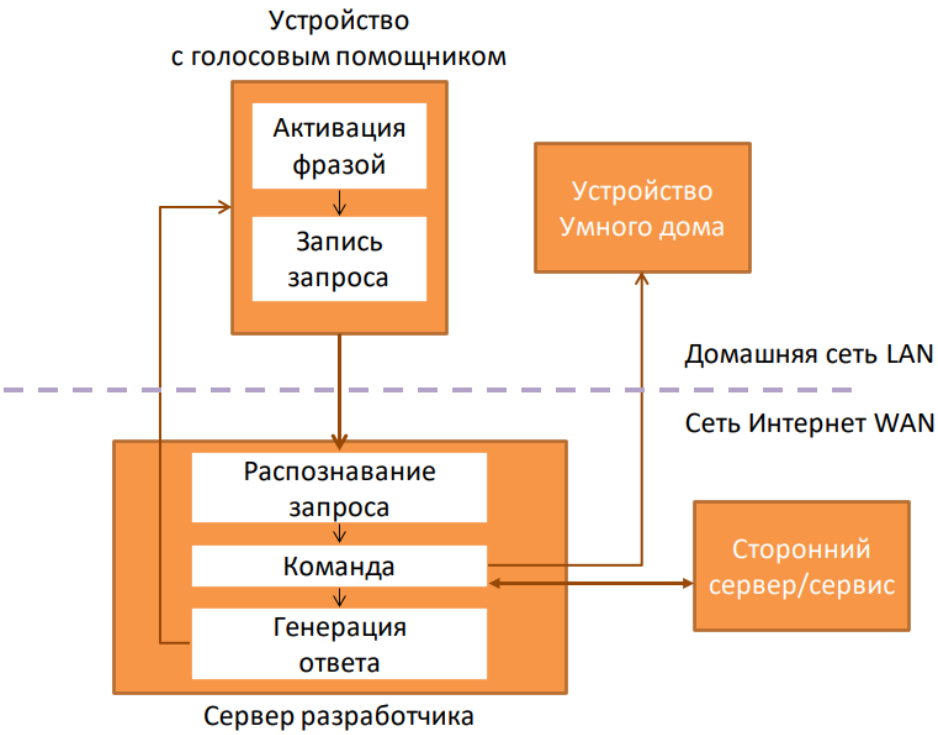


Рис. 1. Схема работы голосового помощника

Далее необходимо определить и классифицировать потенциальных нарушителей, а затем определить их цель. Следует помнить, что основная цель нарушителя: заставить голосового помощника выполнять вредоносные действия без разрешения владельца. В рамках настоящего исследования предпринята попытка систематизации информации и выработки модели

нарушителей и угроз, возможных при использовании голосового помощника, результаты представлены в табличном виде (табл. 1)

ТАБЛИЦА 1 – Модели нарушителей и угроз

Группа	Наименование	Тип	Описание
Частное использование	Другие пользователи устройства	Внутренний	Случайный доступ, одновременные запросы, могут нанести вред по незнанию
	Лица, распространяющие вредоносное ПО и вирусы	Внешний	Обладают достаточными навыками, мотив– кража данных, реже – хулиганство
Публичная система	Технический персонал	Внутренний	Обладают достаточными навыками, знакомы со структурой системы
	Конкуренты	Внешний	Обладают достаточными навыками, мотивы – уничтожение данных, блокирование, нанесение ущерба
	Лица, распространяющие вредоносное ПО и вирусы	Внешний	Обладают достаточными навыками, мотив – кража данных, вывод из строя системы

Одна из главных проблем безопасности связана с возможностью несанкционированного доступа к персональным данным и аккаунтам, которые связаны с использованием голосового помощника. В случае хакерской атаки на систему, которая управляет помощником, злоумышленник может получить доступ к чувствительной информации, такой как финансовые данные, контакты, личные сообщения и другие приватные данные [1].

Кроме того, голосовые помощники могут стать объектом кибератак, возможно, даже с помощью взлома виртуального помощника при помощи звука. Для того, чтобы предотвратить такой сценарий, нужна эффективная система аутентификации и защиты данных, таких как шифрование и механизмы контроля доступа к системе.

Также стоит обращать внимание на то, как голосовой помощник использует информацию, переданную пользователем. Приложение может собирать данные о пользователе, а также записывать и сохранять аудиофайлы, содержащие разговоры людей в своей периферии.

Таким образом, возможно выделить перечень возможных угроз:

– угрозы случайного доступа,
– угрозы утечки акустической (речевой) информации,
– угрозы несанкционированного доступа.

При этом возможно определить основные виды атак:

– атака на подсистему записи запроса (ультразвук, свет/лазер, электромагнитные волны),

– атака на подсистему распознавания запроса (подмена запроса),

– атака на подсистему генерацию ответа (подмена ответа),

– атака на подсистему исполнения команд (подмена команды).

Исследователи из Японии и Мичиганского университета определили способ управлять устройствами Google Home, Alexa от Amazon или Siri от Apple с расстояния в сотни метров, направляя лазерные указки и даже фонари на микрофоны устройств. Голосовой ассистент воспринимает лазерные и ультразвуковые сигналы акустическим приемником (микрофоном) и считывает его как команду [1]. Затем злоумышленник может диктовать данные команды с целью получения какой-либо выгоды. Голосовой помощник улавливает команды и начинает их выполнять. Первой на вход всегда идет команда, уменьшающая громкость голоса ассистента до минимума, так нарушитель скрывает от пользователя внедрение в систему. Далее у злоумышленника возникает возможность использования пользовательского приложения – получение персональной информации, номера карты, доступа в квартиру или к автомобилю.

Следует отметить, что практически все голосовые помощники работают по протоколам HTTP и HTTP, которые являются уязвимы к угрозам [2]. Однако пользователь может снизить риски с помощью следующих действий, необходимых для поддержания безопасности данных:

– настройка прокси-сервера на Wi-Fi маршрутизаторе или любом другом сетевом устройстве, к которому подключен голосовой помощник,

– установка двухфакторной аутентификации,

– принудительное отключение микрофона на устройстве во время бездействия или использование специальных устройств,

– настройка уведомления об активных сессиях с голосовым помощником.

Отдельное внимание стоит уделить рассмотрению проблемы с конфиденциальности данных пользователей. Режим ожидания запросов, который активен постоянно, предполагает постоянное прослушивание окружающей обстановки, что позволяет голосовому помощнику моментально реагировать на команду пользователя. Однако такое прослушивание вызывает у пользователей определенные опасения. В частности, многие пользователи опасаются, что их приватные разговоры могут быть записаны и переданы третьим лицам, таким как рекламодатели, что создает серьезные риски для конфиденциальности и безопасности в целом [2].

Еще одна проблема связана с несовершенствами технологической базы голосовых помощников. Иногда система может допустить ошибку и принять неправильный запрос, что может привести к записи и передаче чувствительных данных, которые могут быть использованы в криминальных

целях. Например, если голосовой помощник ошибочно записывает сводку о банковском счете, злоумышленники могут использовать эти данные для хищения денежных средств [3].

В связи с этим, производители голосовых помощников работают над усовершенствованием системы безопасности – используют современные методы шифрования и аутентификации пользователей, чтобы предотвратить несанкционированный доступ к данным [3]. Также многие производители предоставляют возможность отключить режим постоянного прослушивания, выбрав режим активации только по команде пользователя, что помогает повысить конфиденциальность использования голосовых помощников.

Также в настоящей работе предложены некоторые решения проблем безопасности и конфиденциальности:

– использование альтернативных голосовых помощников, которые гарантируют большую конфиденциальность данных. Некоторые из таких приложений, например, не отправляют данные на серверы компаний-разработчиков,

– установка дополнительных программ для защиты данных, такие как антивирусов, межсетевых экранов и VPN-сервисов.

– внимательное отношение к запросам, которые могут быть использованы против пользователя в будущем. Например, темы личных финансовых вопросов, медицинских проблем или других конфиденциальных данных следует избегать при использовании голосового помощника,

– регулярные проверки настроек голосового помощника и отключение любые опции, которые могут собирать данные без разрешения.

Таким образом, в рамках настоящего исследования выполнен обзор алгоритмы работы голосового помощника, составлена модель нарушителя, выявлены перечень основных угроз и видов атак, даны обширные рекомендации для пользователей. Настоящая работа является основой для дальнейших исследований и разработок в данной тематике.

Список используемых источников

1. Amazon Alexa: What can you do to stop your recordings being kept? // BBC News. 2022. URL: https://www.bbc.com/news/technology-49808208 (дата обращения 18.06.2023).

2. Голосовые помощники: защита конфиденциальности и безопасности // Kaspersky Lab. 2023. URL: https://www.kaspersky.ru/blog/voice-assistants-privacy-security/23751/ (дата обращения 18.06.2023).

3. Голосовые помощники: как защитить свои данные // Forbes 2019. URL: https://www.forbes.com/sites/forbestechcouncil/2019/09/26/voice-assistants-how-to-protect-your-data/?sh=7b5d2e9c2b7d (дата обращения 18.06.2023).

Статья представлена кандидатом технических наук, доцентом Зарубиным А. А.

УДК 004.432.2

У. У. Махмутхажиев, Е. А. Соколов, Г. С. Шейде
(студенты гр. ИКПИ-12, СПбГУТ)

РАЗРАБОТКА IOS-ПРИЛОЖЕНИЯ: ВЛИЯНИЕ QA НА КОНЕЧНЫЙ ПРОДУКТ

В настоящее время разработка мобильных приложений стала важной составляющей бизнеса и технологического развития. Одной из популярных платформ для создания мобильных приложений является iOS, разработанная компанией Apple. В процессе разработки iOS-приложений решающую роль играет не только команда разработчиков, но и квалифицированный специалист по обеспечению качества (QA-тестировщик). В данной статье рассматривается влияние, оказываемое рассмотрим, какое влияние оказывает QA-тестировщиком на конечный продукт разработки iOS-приложения.

разработка приложений, QA, тестирование, iOS-разработка.

Основные этапы разработки iOS-приложения

Разработка iOS-приложения включает несколько основных этапов. Первоначально команда разработчиков проводит анализ требований и проектирование приложения, определяет функциональность и интерфейс приложения. Затем разработчики приступают к программированию, используя язык программирования Swift и интегрированную среду разработки Xcode. После завершения этапа программирования проводится тестирование, которое выполняет QA-тестировщик.

Основная роль QA-специалиста в разработке iOS-приложения состоит в: обеспечении качества iOS-приложения. Он отвечает за выявление и исправление ошибок, проверку соответствия приложения требованиям, а также оценку его функциональности и стабильности. В процессе тестирования проводятся различные виды тестирования, такие как функциональное тестирование, тестирование верстки, тестирование на разных устройствах, нагрузочное тестирование и другие. QA-специалист также обеспечивает проверку безопасности приложения, чтобы убедиться, что оно защищено от взлома и утечки данных.

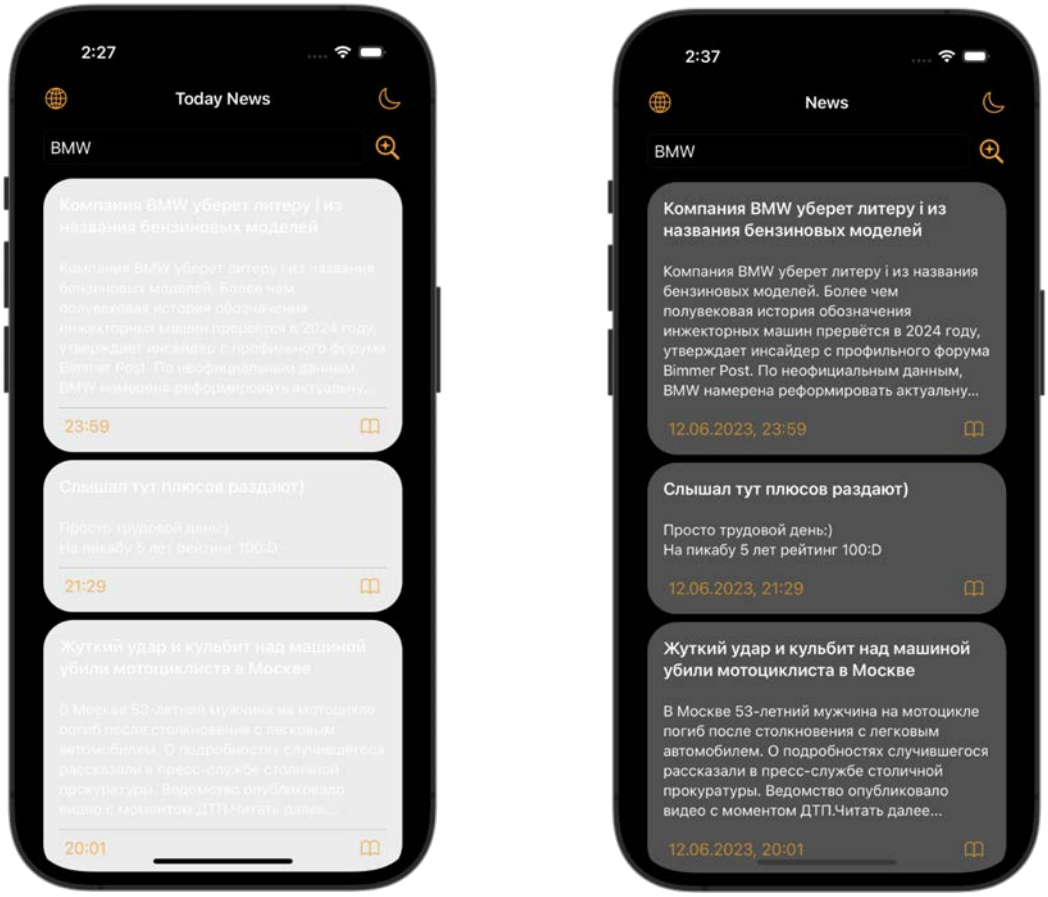
При проведении данного исследования нами было создано мобильное приложение, позволяющее читать новости, собранные, по ключевым словам, и категориям более чем с сотни популярных мировых новостных порталов благодаря задействованию технологии взаимодействия с внешним ресурсом через API. В ходе создания приложения было написано и протестировано несколько версий, от первой «сырой» бета-версии, до конечного

продукта, готового к выходу в релиз. Далее мы рассмотрим влияние тестировщика на конечный продукт с наглядными примерами из процесса создания нашего авторского приложения.

Влияние QA-специалиста на конечный продукт

Профессиональный QA-специалист способен значительно повлиять на конечный продукт разработки iOS-приложения. Его работа позволяет выявить и устранить ошибки и проблемы до выпуска приложения на рынок, что способствует повышению его качества.

Также помогает повысить удобство использования приложения, обнаруживая неполадки в пользовательском интерфейсе, некорректное отображение данных и другие проблемы, которые могут негативно повлиять на пользовательский опыт. В процессе разработки новостного приложения были выявлены одни из самых популярных багов среди всех приложений, а именно некорректное отображение текста и форм в режиме тёмной темы (рис. 1).



а) б)

Рис. 1. Темная тема v. 1 (а), темная тема v. 2 (б)

Благодаря тщательному тестированию повышается надежность и стабильность приложения. Обнаруживаются потенциальные проблемы с производительностью и оптимизируется работа приложения, что позволяет снизить вероятность возникновения сбоев и зависаний. В ходе тестирования одной из версий приложения благодаря нагрузочному тестированию были выявлены ошибки из-за загрузки превью новостей в высоком разрешении, которые впоследствии были решены командой разработки.

Одним из важных заданий QA-специалиста является обеспечение высокого уровня безопасности приложения. Он проводит тестирование на наличие уязвимостей, проверяет аутентификацию и авторизацию, а также осуществляет анализ защищенности системы.

Кроме вышеуказанных задач, QA-тестировщик также выполняет тестирование на разных языках и локализациях, проверяет совместимость с различными версиями iOS и обеспечивает соответствие приложения стандартам качества. В бета-версии нашего приложения были обнаружены ошибки в работе приложения на русском языке, в частности – не локализованы плейсхолдеры в форме поиска новостей (рис. 2).

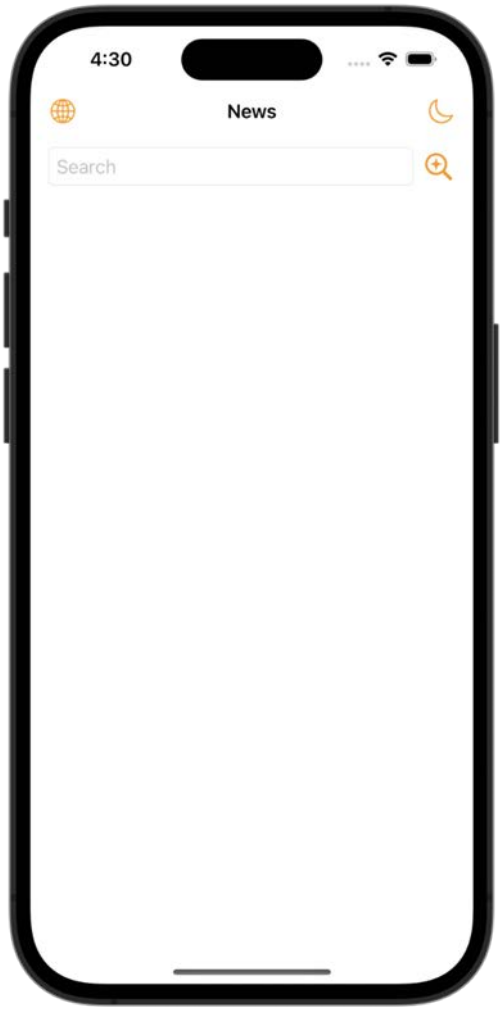


Рис. 2. Ошибки локализации в плейсхолдере

Другим важным аспектом является удобство использования и пользовательский опыт. Он проверяет навигацию, интерфейс пользователя, а также соответствие приложения гайдлайнам и рекомендациям Apple. QA стремится обеспечить интуитивно понятный и легкий в использовании интерфейс, который позволит пользователям без труда осваивать и работать с приложением. Например, он может вернуть продукт в разработку, предложив идею более удачной реализации интерфейса, например, иконки кнопок, их расположение на экране, добавление элементов к общему интерфейсу приложения. При разработке авторского приложения были предложены и внедрены такие изменения, как добавление превью новостей, более интуитивно понятные иконки кнопок и др. Удалось исправить ошибки локализации, ошибки верстки и отображения темной темы, отображения даты публикации новости, а также добавлены превью новостей и автоподбор новостей по региону на главном экране при запуске

приложения. Итоговая версия приложения, которая отправится в релиз выглядит так, как показано на рис. 3.

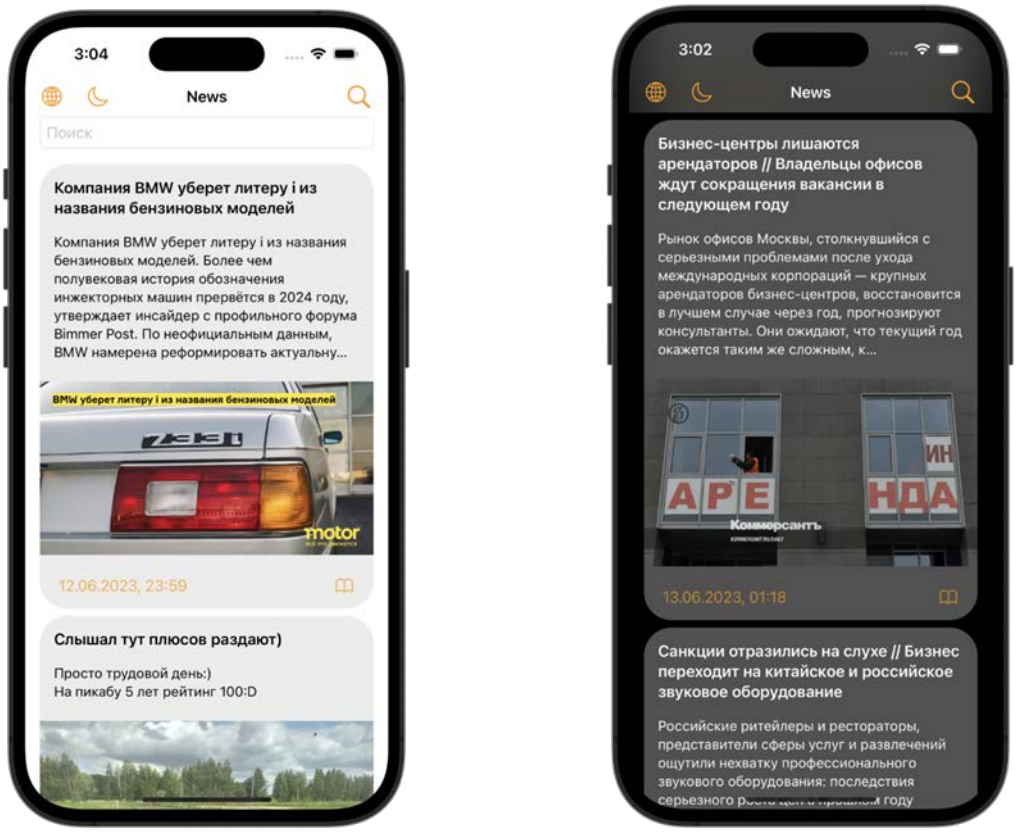


Рис. 3. Вывод новостей по запросу и экран при открытии приложения

Итак, работа QA-тестировщика обеспечивает безопасность, надежность, производительность, совместимость и удобство использования приложения. QA-специалист играет важную роль в процессе разработки, помогая выявлять и исправлять ошибки, формулировать проблемы, а также повышать общую качественную характеристику продукта. Сотрудничество между разработчиками и QA-специалистами является ключевым фактором для достижения успеха на рынке мобильных приложений и удовлетворения потребностей пользователей.

Список используемых источников

1. Документация Apple для разработчиков: Разработка iOS-приложений [Электронный ресурс]. URL: https://developer.apple.com/documentation/ios (дата обращения 16.05.2023).

2. Канер, С., Фолк, Дж., & Нгуен, Х. К. Тестирование программного обеспечения. Практическое руководство. Питер, 2019.

3. Серрано, Ж. Практическая разработка приложений для iOS 14: SwiftUI, UIKit и App Clips. Apress. 2021.

Статья представлена научным руководителем, и. о. заведующего кафедрой ВМ СПбГУТ, кандидатом физико-математических наук Плотниковым П. В.

УДК 681.335

И. Н. Медведев (студент гр. ИКМ-12з, СПбГУТ)

ИСПОЛЬЗОВАНИЕ ОПТИЧЕСКОГО BER-ТЕСТЕРА ДЛЯ ОПРЕДЕЛЕНИЯ КАЧЕСТВА ПРОТЯЖЕННОЙ ОПТИЧЕСКОЙ ЛИНИИ СВЯЗИ

Частота битовых ошибок (BER) – это показатель целостности телекоммуникационного сигнала и эффективный показатель качества линии связи, основанный на количестве или проценте переданных битов, которые были приняты неправильно. Для измерения показателя BER используются специальные программно-аппаратные комплексы BER-Tester (BERT). В данной работе описываются технологии, используемые в анализаторах битовых ошибок (BERT), а также принцип действия BER-тестеров и сферы применения данного оборудования.

BERT, протяженная линия, BER, коэффициент битовых ошибок, ВОЛС.

Для проверки качества линии связи зачастую недостаточно знать лишь физические характеристики оптической трассы и входящих в нее пассивных компонентов. Для определения физических характеристик оптической трассы существует множество различных приборов. В основном используются оптические рефлектометры, измерители мощности, спектроанализаторы и прочие. Эти приборы незаменимы при строительстве оптической линии и при выявлении неисправностей в процессе эксплуатации. Однако, для показательной оценки пригодности линии связи к передаче определенного типа трафика лучше всего подходит BER-тестер (BERT).

Частота битовых ошибок (BER) – это показатель целостности телекоммуникационного сигнала, основанный на количестве или проценте переданных битов, которые получены неправильно. BER является одним из важнейших показателей качества любой цифровой линии связи, в том числе и оптической.

BERT используется для количественной оценки коэффициента битовых ошибок в волоконно-оптических системах передачи данных, Ethernet или любой системы, которая передает данные по сети. В отличие от других методов тестирования, BERT позволяет оценить производительность и надежность передачи по всей оптической линии, включая передатчик, приемник, оптическое волокно и пассивные компоненты. Так, например, не только слишком малые уровни сигнала, но и слишком высокие значения уровня сигнала способны привести к перегрузке приемника и, как следствие, появлению битовых ошибок [3].

Принцип работы BER-тестеров основан на передаче псевдослучайной битовой последовательности – последовательности бит, которая генерируется по определенному алгоритму. Псевдослучайная последовательность

трудно предсказуема и статистически максимально приближена к сигналу, который возможен при использовании сети в работе.

Для выполнения теста частоты ошибок в битах предварительно определенный поток данных, в виде псевдослучайной битовой последовательности, отправляется через вход сетевого канала, затем выходные данные канала на принимающей стороне анализируются для оценки количества обнаруженных ошибок в сравнении с количеством битов, переданных за заданный период времени [1].

Частота ошибок в битах рассчитывается путем деления количества битов, полученных с ошибкой, на общее количество битов, переданных за тот же период времени.

В общем виде схема BER-тестера показана на рис. 1.

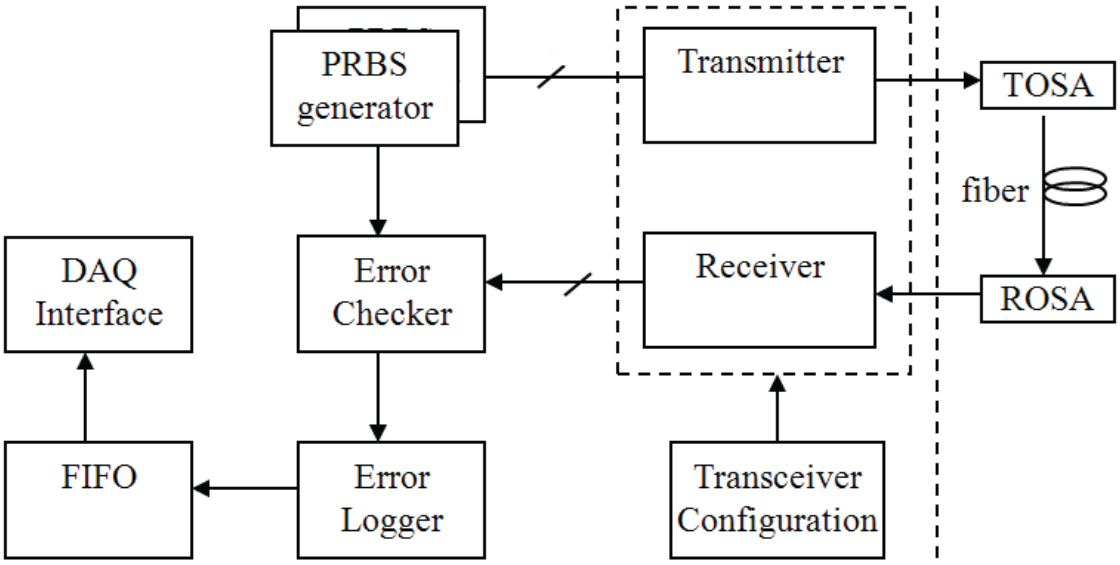


Рис. 1. Принципиальная схема BER-тестера

Процесс измерения битовых ошибок можно разбить на пять этапов:

- генерация псевдослучайной последовательности;
- передача псевдослучайной последовательности;
- прием псевдослучайной последовательности;
- сравнение выходных и входных данных;
- запись результата.

Коэффициент битовых ошибок является статистической величиной, для того чтобы вычислить истинное значение BER потребовалась бы псевдослучайная последовательность бесконечной длины, что не имеет никакого практического смысла. Поэтому коэффициент битовых ошибок определяется с устанавливаемой доверительной вероятностью, выражаемой от 0 до 1. Для вычисления количества бит, требуемого для определения коэффициента BER с заданной вероятностью используется выражение:

$$N_{\text{бит}} = \frac{-\ln(1 - CL)}{BER},$$

где *BER* – это требуемый коэффициент битовых ошибок, а *CL* – доверительная вероятность.

Для большинства оптических систем передачи данных достаточно, чтобы коэффициент битовых ошибок не превышал 10^{-12}, т. е. один ошибочно переданный бит на 10^{12} верно переданных. Проверка на меньшие значения *BER* может быть избыточна, так как может потребовать значительно больше времени на ее проведение, в то время как для большинства приемосдаточных испытаний требуется подтвержденное значение $BER < 10^{-12}$ [2].

Одним из показателей избыточности более точного определения BER является время для статистического измерения количества битовых ошибок, которое можно определить по выражению:

$$t_{\text{изм}} = \frac{-\ln(1-CL)}{B \cdot BER},$$

где *B* – битовая скорость передачи.

В таблице 1 представлены результаты расчетов времени измерения, за которое с вероятностью 95 % *BER* будет ниже заданного значения для разных скоростей и модуляций.

ТАБЛИЦА 1. Зависимость времени измерения BER от скорости передачи

	PCI Express 5Gb/s NRZ	PCI Express 16 Gb/s NRZ	IEEE 802.3 bj 25.78125 Гбод NRZ	IEEE 802.3 bs 26.5625 Гбод PAM-4	IEEE 802.3 bs 53.125 Гбод PAM-4
10^{-15}	~ 166 ч.	~ 52 ч.	~ 32 ч.	~ 31 ч.	~ 16 ч.
10^{-12}	~ 10 мин.	~ 3,1 мин.	~ 1,92 мин.	~ 1,9 мин.	~ 1 мин.
10^{-9}	~ 600 мс	~ 190 мс	~ 120 мс	~ 110 мс	~ 56 мс
10^{-6}	~ 600 мкс	~ 190 мкс	~ 120 мкс	~ 110 мкс	~ 56 мкс

Существует ряд причин, по которым следует проводить BER тестирование на критически важных участках ВОЛС:

Во-первых, основной протокол передачи данных, который используется в сетях до 100G это Ethernet. Протокол Ethernet может маскировать реальный BER, за счет того, что в этом протоколе встроены механизмы проверки и коррекции ошибок. В случае получения ошибочного бита, неправильно принятые пакеты пересылаются повторно, чтобы обеспечить правильность принятия. Однако, формальное отсутствие ошибок на уровне протокола Ethernet (2 уровень модели OSI), не является показателем отсутствия ошибок на физическом уровне (1 уровень OSI). На повторную отправку неправильно принятого пакета расходуется ресурс канала передачи, а учитывая то, что один неправильно принятый бит может привести к пересылке целого кадра длиной до 12000 бит, можно сделать вывод что низкий

BER на физическом уровне может привести к существенному снижению полосы пропускания.

Еще одной из причин необходимости BER-тестирования современных сетей является то, что не все сети строятся по технологии Ethernet, а также не все протоколы обладают возможностью коррекции ошибок. Как один из примеров, широковещательные сети, такие как CaTV, передающие прямой поток данных к пользователям. Битовые ошибки в таком сигнале могут привести к искажению и ухудшению качества передаваемого изображения.

Особо тщательного подхода к передаче видеосигнала требует сфера телемедицины, в которой ухудшения качества недопустимо и может повлечь серьезные последствия.

Существует два подхода к измерению BER, которые дополняют друг друга.

Первый – измерение BER в лаборатории, на стадии проектирования и тестирования потенциальной оптической трассы или ее участка. В лаборатории собирается стенд, на котором с помощью катушек с оптическим волокном моделируется трасса и тестируется оборудование, которое планируется использовать. Для лабораторных испытаний используются специальные лабораторные BERT-установки. Они не требуют портативности и могут стоить дешевле своих портативных аналогов.

Второй подход к измерению BER – это приемосдаточные измерения или же измерения в процессе эксплуатации, с целью проверки причины неисправности линии или ее профилактики. Привезти лабораторный прибор и установить его в таких условиях не представляется возможным, к тому же, может потребоваться модуль loopback, которого для лабораторной установки может быть не предусмотрено. Поэтому существуют портативные BER-тестеры, которые позволяют проводить измерение качества линии на месте ее расположения.

Список использованных источников

1. Волоконно-оптическая техника: современное состояние и новые перспективы / под ред. С. А. Дмитриева, Н. Н. Слепова. М. : Волоконно-оптическая техника : Техносфера, 2010. 607 с. : ил., табл.; 25 см.; ISBN 978-5-94836-245-8.

2. Гайнов В. В., Конышев В. А., Леонов А. В., Лукиных С. Н., Наний О. Е., Скворцов П. И., Трещиков В. Н., Шихалиев И. И., Убайдуллаев Р. Р. Однопролетные оптические линии связи большой протяженности // Прикладная фотоника, 2015. Т. 2. № 1. С. 5–22.

3. Родина О. В. Волоконно-оптические линии связи : практическое руководство. – М. : Горячая линия-Телеком, 2012. 400 с. : ISBN 978-5-9912-0109-4. Текст : электронный. URL: https://znanium.com/catalog/product/360394 (дата обращения: 09.12.2023). Режим доступа: по подписке.

Статья представлена научным руководителем, доцентом кафедры ФиЛС СПбГУТ, кандидатом технических наук Дюбовым А. С.

УДК 004.056

А. Е. Платонов (студент гр. ИБС-03, СПбГУТ)

УГРОЗЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ, СВЯЗАННЫЕ С НЕЗАКОННЫМ ОБОРОТОМ SIM-КАРТ

В последние годы правоохранительные органы и Роскомнадзор фиксируют рост незаконного оборота SIM-карт для мобильных телефонов, что создает угрозы информационной безопасности. Указанные SIM-карты регистрируются у мобильных операторов на подложные данные, по которым нельзя установить их истинную принадлежность. В статье рассмотрены основные угрозы, связанные с отсутствием идентификации пользователей мобильной связи, выделены основные способы незаконного распространения таких карт, описаны меры профилактики и составы административных правонарушений, предписанные статьями 13.29 и 13.10 КоАП РФ.

информационная безопасность, незаконный оборот SIM-карт, административное правонарушение.

Ответственность за нарушение

Правонарушения в сфере незаконного оборота SIM-карт входят в сферу главы 13 Кодекса об административных правонарушениях (КоАП РФ) [1], которая определяет административную ответственность за правонарушения в области связи и информатизации. В частности, статьей 13.29 КоАП РФ предусмотрена ответственность за заключение от имени оператора связи договора об оказании услуг подвижной радиотелефонной связи лицом, не имеющим полномочий от оператора связи на заключение договора об оказании услуг подвижной радиотелефонной связи. Указанная статья предусматривает наложение штрафа на физических лиц от двух тысяч до пяти тысяч рублей, на должностных – от десяти тысяч до пятидесяти тысяч рублей, на юридических лиц – от ста тысяч до двухсот тысяч рублей.

Также, статьей 13.30 КоАП РФ, утратившей силу с 1 марта 2023 года, предусматривалась ответственность за отсутствие в договоре идентификационных данных абонента, таких как: фамилия, имя, отчество, место жительства и паспортные данные, либо предоставление оператору ложных данных.

Основные лица, совершающие правонарушение

Чаще всего, правонарушения по незаконному обороту SIM-карт совершают так называемые «серые дилеры», которые приобретают у мобильных операторов на свою организацию большое количество таких карт и затем их перепродают физическим лицам. Последние зачастую используют такие

симки для сокрытия своей личности при организации различного рода криминальных преступлений. Кроме этого безответственные сотрудники магазинов операторов связи и их дилеров, которым требуется выполнить план новых подключений абонентов, сознательно могут идти на нарушения регламентов, не задумываясь о тяжести возможных последствий.

Статистика

Ниже приведена статистика за 2016 год по количеству изъятых из незаконного оборота SIM-карт, осуществленных территориальными органами Роскомнадзора во взаимодействии с органами МВД. На диаграмме приведено процентное соотношение по основным операторам мобильной связи (рис. 1).

Рис. 1. Статистика по изъятым из незаконного оборота в Северо-Западном ФО SIM-картам в 2016 году

В 2016 году Северо-Западный федеральный округ был лидером по количеству административных правонарушений в сфере незаконного оборота SIM-карт. К сожалению, в прошлом 2022 году цифры оказались значительно весомее – Роскомнадзором доложено об изъятии не менее 152 тысяч подобных сим-карт, выданных с нарушением закона [3].

Методы реализации SIM-карт

Криминальные группировки

Данные группировки закупают SIM-карты у операторов напрямую на специально открытых юридических лиц, так называемых «серых» дилеров. Далее при продаже, договоры между абонентом и оператором связи не заключаются, поскольку карточки зарегистрированы на юридическое лицо. Списки пользователей SIM-карт, в нарушение действующих правил, операторам связи не направляются. При этом, как правило, сотовые компании получают хорошую финансовую прибыль за использование услуг связи «серыми» дилерами и не требуют от них данных о принадлежности карт конкретным лицам. В ходе специально организованных рейдов сотрудниками Роскомнадзора и органов МВД изымаются из незаконного оборота от десятков до нескольких тысяч SIM-карт у «серых» дилеров [4].

Сотрудники ритейла

В данном случае, сотрудником ритейла оператора сотовой связи или же официальным дилером оформляются поддельные договоры на существующих лиц (жертв), которые ранее оформляли на себя SIM-карты и оставили свои персональные данные в договорах на обслуживание. Сотрудник умышленно вводит эти данные в новые договоры, зачастую у других операторов связи, о чем жертвы даже не догадываются. Далее данные карты продаются сотрудниками с завышенной стоимостью доверенным лицам, которые зачастую связаны с криминалом или спецслужбами враждебных государств [5].

Основные каналы реализации

Ниже приведена схема с основными каналами реализации SIM-карт (рис. 3).

Рис. 3. Основные каналы реализации нелегальных SIM-карт

Угрозы от продажи нелегальных SIM-карт

Незарегистрированные SIM-карты являются отличными инструментом для разного рода группировок и преступников, целью которых является сокрытие своей активности и затруднения своей идентификации.

Спецслужбы иностранных государств и террористические группировки

Данные организации используют нелегальные SIM-карты для анонимной коммуникации, координации своих действий, распространению экстремисткой пропаганды, вербовке новых участников, денежных транзакций, закупке оборудования и использования в самодельных взрывных устройствах. самое опасное для общества – использование спецслужбами иностранных государств анонимных SIM-карт для террористических целей. В настоящее время они являются основной угрозой от использования анонимных SIM-карт [6].

Криминальные группировки

Основными целями использования таких SIM-карт являются анонимная коммуникация для продажи наркотиков.

Кроме этого данные группировки могут обманным путем получать деньги от мобильных компаний. Зарегистрированный злоумышленниками за рубежом виртуальный оператор выставляет телекоммуникационным компаниям счета за так называемые платные услуги (благотворительность,

интим по телефону и т. п.) в зависимости от продолжительности соединения. С помощью SIM-карт, оформленных на подложные данные, с территории других государств производится большое количество соединений на данный сервис. Далее виртуальный оператор выставляет счета для оплаты телекоммуникационной компании, с которой у него есть договор на обслуживание. Далее счета на оплату по цепочке проходят по всем телекоммуникационным компаниям, увеличиваясь в размере на сумму услуг каждого из операторов. Последним в цепочке является оператор, зарегистрировавший SIM-карту на подложные данные. При этом счета на оплату формируются всеми операторами в цепочке с большим запозданием по отношению ко времени совершения соединений. Крайний мобильный оператор, несмотря на отсутствие средств на счету SIM-карты с подложными данными, вынужден оплатить выставленные счета следующему оператору в цепочке уже оплатившему услуги соединения из собственных средств. Таким образом, родительская компания не может получить средства по подложным данным и останется в минусе.

Мошенники

Используют SIM-карты для массовой рассылки мошеннических материалов, регистрации поддельных аккаунтов, звонков с рекламой, звонков от лица банков и государственных учреждений с целью ввода жертвы в заблуждение и последующей наживы.

Риски информационной безопасности

С позиции информационной безопасности серые SIM карты также представляют угрозу. Если сотрудник использует такую карту в рабочем телефоне, регистрирует аккаунты и коммуницирует, используя данный номер телефона, то данный канал связи нельзя считать защищенным. Злоумышленники могут восстановить проданную карту и получить доступ ко всем зарегистрированным аккаунтам, восстановив пароли. Также, атакующие могут звонить от имени сотрудника и получать доступ к критической информации используя социальную инженерию [8–10].

Методы борьбы

Основные методы борьбы, автор видит в увеличении количества профилактических проверок, совершенствовании законодательства РФ в части обязанности операторов сотовой связи вести более жесткий контроль за посредниками, реализующими их SIM-карты, и в ответственности за актуальность данных абонентских договоров на предоставление услуг сотовой связи.

Вывод

В данной статье проведен анализ методов применения и каналов реализации нелегальных SIM-карт, рассмотрены основные угрозы, исходящие от неконтролируемого распространения и использования данных SIM-карт, а также были предложены методы по борьбе с данным явлением.

Список используемых источников

1. Федеральный закон от 28.04.2023 № 175-ФЗ «О внесении изменений в Кодекс Российской Федерации об административных правонарушениях», раздел 2, глава 13 [Электронный ресурс]. URL: http://publication.pravo.gov.ru/Document/View/0001202304280023 (дата обращения 08.05.2023).

2.Роскомнадзор, отчет об итогах деятельности за 2016 год [Электронный ресурс]. URL: https://29.rkn.gov.ru/p2582/p5458/news111428.htm (дата обращения 08.05.2023).

3. Чьи сим-карты заблокируют уже в мае 2023 года// LIFE.RU 03.05.2023 [Электронный ресурс]. URL: https://news.mail.ru/society/56067333/?frommail=1&utm_partner_id=736 (дата обращения 08.05.2023).

4. Роскомнадзор, отчеты о деятельности [Электронный ресурс]. URL: https://rkn.gov.ru/plan-and-reports/reports/ (дата обращения: 08.05.2023).

5. Серая зона: как работает рынок левых сим-карт в России// РИА Новости 05.07.2017 [Электронный ресурс] URL: https://ria.ru/20170705/1497891651.html (дата обращения 08.05.2023).

6. Чем опасны серые SIM-карты? // Securitylabs.ru 30.07.2019 [Электронный ресурс]. URL: https://www.securitylab.ru/blog/company/CABIS/346759.php (дата обращения 08.05.2023).

8. Шашкин В. С., Сахаров Д. В., Яковлев В. А., Андрианов В. И. Основные угрозы социальной инженерии в государственных учреждениях // Материалы Всероссийской научно-практической конференции «Национальная безопасность России: актуальные аспекты» ГНИИ «Нацразвитие». Июль 2018. Сборник избранных статей. 2018. С. 36–46.

9. Карев А. С., Бирих Э. В., Сахаров Д. В., Виткова Л. А. Проблемы информационной безопасности в интернете вещей // Интернет вещей и 5G. 2016. С. 66–70.

10. Красов А. В., Сахаров Д. В., Тасюк А. А. Проектирование системы обнаружения вторжений для информационной сети с использованием больших данных // Наукоемкие технологии в космических исследованиях Земли. 2020. Т. 12. № 1. С. 70–76.

Статья была представлена научным руководителем, доцентом кафедры ЗСС СПбГУТ, кандидатом технических наук, доцентом Сахаровым Д. В.

УДК 004

И. Б. Сединкин (студент гр. ИКВТ-21, СПбГУТ)

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И КИБЕРБЕЗОПАСНОСТЬ: ВЗГЛЯД НА БУДУЩЕЕ БОРЬБЫ С НОВЫМИ ВИДАМИ УГРОЗ

В данной работе рассматриваются новые виды угроз в киберпростанстве, которые реализуются с помощью технологий искусственного интеллекта. Определена актуальность проблемы, описываемой в данной статье. Описаны наиболее распространённые типы современных киберугроз, способы их определения, а также меры противодействия им.

киберугрозы, искусственный интеллект, защита данных.

В настоящее время наблюдается активное проникновение технологий искусственного интеллекта (ИИ) в различные сферы нашей жизни, включая киберсреду. Новые способы в сфере кибермошенничества уже стали угрозой национальной безопасности многих стран мира, включая Российскую Федерацию [1]. По итогам 2022 года было отмечено, что каждое четвёртое преступление совершалось с использованием ИКТ [2], и было зафиксировано более полумиллиарда попыток перехода по вредоносным ссылкам. Это свидетельствует, что общий уровень киберпреступности остаётся стабильно высоким.

Цель работы заключается в попытке обобщения основных видов киберугроз с использованием ИИ, включая описание основных признаков и мер противодействия.

На основе изученной литературы выделены наиболее распространенные виды киберугроз:

1 **Вредоносное ПО** (англ. *Malware, malicious software*) – разновидность программного обеспечения, созданного для нанесения ущерба компьютеру и данным, хранящимся в них. Существует множество различных видов вредоносного ПО, каждый из их обладает своими особенностями, по-своему наносит ущерб компьютерам. Однако присутствие такого ПО можно обнаружить по «внешним признакам»: низкая производительность компьютера, большая нагрузка на ЦП, произвольное открывание/закрывание приложений, внезапное отключение компьютера и др. Распространение вредоносного ПО стало более эффективным при использовании ИИ: он значительно понижает вероятность быть обнаруженным аналитиками, что заставляет их использовать альтернативные способы борьбы [3]. В четвёртом квартале прошлого года было совершено 49 % успешных атак на частные лица с при-

менением шпионского ПО, а самыми распространёнными способами его доставки являлись фишиноговые сайты, электронные письма и социальные сети [4].

2 **Дипфейк (**англ. *deepfake, deep learning*) – технология синтеза изображений, аудио- и видеозаписей с помощью искусственного интеллекта. В век информационных технологий человек ежедневно сталкивается с информацией разного формата и содержания. Информация стала производиться в таком огромном количестве, что человек, в связи с его объективными свойствами, уже не способен заниматься её обработкой без участия ИИ. Технологии ИИ активно применяются в обработке различных медиа, включая видео, аудио и изображения (дипфейки). Следовательно, возникла угроза появления заведомо ложной информации, представленной в различных форматах. Практика показала, что человек не способен эффективно отличать реальное изображение, аудио и/или видео от фальшивого без применения дополнительных технологий, и с течением ситуация будет только усугубляться [5]. Как следствие, в 2019 году британская энергетическая компания лишилась 220 000 долларов с применением дипфейковой записи голоса директора компании, что является первым зарегистрированным случаем применения данной технологии в отношении крупных организаций [6].

3 **Информационные атаки** – массированное информационное воздействие, направленное на нарушение целостности, доступности и конфиденциальности информации. Информационные атаки имеют совершенно разную направленность: на государственные учреждения, на частные компании, на физические лица. В век информационных технологий основным «полигоном» для информационных атак становятся социальные сети, являющиеся удобным местом распространения информации. В условиях стремительного роста ИИ-технологий информационные атаки усовершенствовались и требуют принципиально иного подхода к их устранению [5]. В 2022 году количество успешных кибератак на веб-ресурсы организаций увеличилось на 56 %. Если в 2021 году компании становились жертвами 17 % случаев, то в 2022 году – в 22 %. Возросло количество инцидентов во многих отраслях, особенно удар пришелся на госучреждения – количество атак более чем удвоилось и составляет 41 % от всех случаев. [6].

4 **Фишинг** – вид интернет-мошенничества, при котором происходит получение несанкционированного доступа к персональным данным путём перехода по вредоносным ссылкам. Такой вид мошенничества использует массовую рассылку электронных писем, содержащих вредоносные ссылки. Они ведут на поддельные сайты популярных брендов и компаний. ИИ научился создавать дизайн сайта-подделки, практически полностью повторяющий дизайн сайта-оригинала. Как следствие, при первом приближении подделки трудно распознать человеческим взглядом. На таких сайтах посетители добровольно оставляют свои личные данные, которые могут быть

использованы мошенниками в корыстных целях. В 2022 году система «Антифишинг» Касперского выявила и предотвратила более полумиллиарда попыток перехода по фишинговым ссылкам, а также более 378 496 попыток угона Telegram-аккаунтов. Это говорит о том, что проблема фишинга всё ещё является одной из самых острых [1]. Также фишинг является наиболее распространённым способом передачи вредоносного ПО [6].

5 **Недостоверная информация** – информация, не соответствующая действительности, используемая с целью манипуляции общественным мнением, ввода в заблуждение, призыва к противоправным действиям. В виду того, что объём создаваемого контента огромен, человек физически не справляется с его анализом без поддержки ИИ. Как следствие, может быть упущена из виду заведомо ложная и не соответствующая действительности информация. Такая информация с высокой вероятностью появляется в СМИ, чья аудитория активно следит за актуальными новостями и может быть целенаправленно обманута [5]. Команда исследователей CSET (*Center of Security and Emerging Technologies*) продемонстрировала способность ИИ генерировать дезинформацию и влиять на мнения людей. Эксперименты показали, что человек действительно реагирует на информацию, предоставляемую ИИ, и может поменять свою прежнюю позицию [7].

Для противодействия обозначенным видам угроз актуальным является реализация ряда мер:

1 **Использование антивирусного ПО** – одно из наиболее доступных способов бороться с вредоносными программами. Антивирусное ПО – программное обеспечение или их совокупность, направленные на выявление и устранение вредоносных программ и их профилактику. Антивирус сканирует компьютер на наличие вредоносных программ и блокирует их действие, а также занимается восстановлением уже повреждённых файлов. Антивирусы должны постоянно обновляться. Необходимость обновлений вытекает из постоянного усовершенствования вредоносного ПО: их способности обходить защиту систем. Современное антивирусное ПО оснащено поддержкой ИИ для ускоренного анализа влияния вредоносных программ на компьютеры. Согласно исследованию, IT Security Survey 2023 австрийской лаборатории AV-Comparatives, антивирусы Bitdefender, Kaspersky, ESET и Avast входят в четвёрку лучших антивирусов на каждом континенте. Первые два являются наиболее популярными для Android OS [8].

2 **Модерация контента** – обработка информационного контента на платформе с целью исключения запрещённой и/или нежелательной информации. Любой контент нуждается в обработке для обеспечения информационной безопасности на разных платформах. Традиционно такого рода деятельностью занимался и до сих пор занимается человек. Однако в условиях стремительной генерации контента как людьми, так и ИИ, человек перестал справляться с данной задачей самостоятельно без привлечения дополнительных технологий. Кроме того, человек постоянно сталкивается

с недопустимым контентом, который может повлиять на его психологическое здоровье. Искусственный интеллект, несмотря на объективные недостатки при решении данных задач, способен эффективнее справляться с ними [5]. Microsoft разработала сервис Azure AI Content Safety, предназначенный для автоматизированной модерации контента для чат-ботов Bing и Copilot, основанной на нейросетевых технологиях. По словам Microsoft, Azure AI Content Safety защищает от предвзятого, сексистского, расистского, ненавистного, насильственного и саморазрушительного контента. Однако периодически чат-бот Bing выдавал нелицеприятный и оскорбительный контент, что является поводом для скептического отношения к данному сервису. Несмотря на это, продолжается работа по его усовершенствованию [9].

3 **Выполнение процедуры фактчекинга** (англ. «*fact checking*») – проверка сведений на достоверность и соответствие действительности. В эпоху постправды общественное мнение формируется не на основе объективных фактов, а субъективных утверждениях, что порождает дезинформацию, распространяющуюся на огромную аудиторию. С этой целью выполняется значительная работа по её противодействию. Очевидно, что в условиях информационного хаоса этот процесс требует автоматизации с использованием ИИ [5]. Не всегда он способен это делать без участия человека. По мнению Эндрю Дадфилда, главы отдела по автоматизированному фактчекингу в Full Fact: «пока нельзя настолько доверять проверку автоматической системе – сейчас это преждевременно» [10]. Также он добавляет: «Для этой задачи нужно учесть столько тонкостей и выполнить ее с такой точностью, на которую технологии пока не способны». Несмотря на все сложности, имеются значительные успехи в реализации данной идеи.

4 **Популяризация идей социальной кибербезопасности** – новая дисциплина, находящаяся на стыке ИКТ и социологии, целью которой является исследование влияния инструментов манипуляций на человеческое поведение и разработка методов по их сопротивлению. Сравнительно молодая наука, появление которой стало следствием перехода информационных войн на различные интернет-платформы. Получить доступ к персональным данным можно не только путём нахождения уязвимости в компьютерных сетях, но и путём манипулирования человеком, используя его слабости. Человек может оказаться в искусственно созданной мошенниками ситуации, когда он, пребывая в стрессе, перестаёт мыслить критически. Психологически ослабленный, он может решиться на поступки, на которые в привычном состоянии он бы не отреагировал. Социальная кибербезпасность использует методы, основанные на работе в многомерном сетевом анализе, науке о данных, машинном обучении, обработке естественного языка и моделировании на основе агентов, что позволяет сделать основные выводы: какие способы манипуляций использовать, в чей адрес, в чью пользу и как оказывать им сопротивление [5].

Таким образом, можно сделать вывод, что кибрепреступники активно используют технологии ИИ для совершенствования видов мошенничества, и особенно важным становятся вопросы постоянного информирования пользователей о новых видах угроз.

Список используемых источников

1. Отчёт «Лаборатории Касперского» о спаме и фишинге в 2022 году | Securelist: [Электронный ресурс] // Аналитика и отчёты о киберугрозах «Лаборатории Касперского». URL: https://securelist.ru/spam-phishing-scam-report-2022/106719/ (дата обращения 23.06.2023).

2. Число киберпреступлений в России: [Электронный ресурс] // TAdviser – портал выбора технологий и поставщиков. URL: https://www.tadviser.ru/index.php/Статья:Число_киберпреступлений_в_России (дата обращения 23.06.2023).

3. Malicious Uses and Abuses of Artificial Intelligence [Электронный ресурс] // Home | Europol. URL: https://www.europol.europa.eu/cms/sites/default/files/documents/ malicious_uses_and_abuses_of_artificial_intelligence_europol.pdf (дата обращения 23.06.2023).

4. Дипфейки и другие поддельные видео – как оставаться в безопасности | Лаборатория Касперского: [Электронный ресурс] // Защитные решения кибербезопасности кибезбезопасности для дома и бизнеса | Лаборатория Касперского. URL: https://www.kaspersky.ru/resource-center/threats/protect-yourself-from-deep-fake (дата обращения 23.06.2023).

5. Эксперты назвали 12 приоритетных для государства областей применения искусственного интеллекта: [Электронный ресурс] // Научно-технический центр ФГУП «ГРЧЦ» (НТЦ) – НТЦ. URL: https://rdc.grfc.ru/2023/04/ai_priority_area_application/ (дата обращения 23.06.2023).

6. Актуальные киберугрозы: итоги 2022 года: [Электронный ресурс] // Positive Techologies – безопасность, консалтинг, compliance management. URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2022/ (дата обращения 23.06.2023).

7. ИИ может писать дезинформированные тексты и формировать мнение пользователей соцсетей: [Электронный ресурс] // SecurityLab.ru. URL: https://www.securitylab.ru/news/520535.php (дата обращения: 23.06.2023).

8. Итого опроса AV-Comparatives 2023: Популярные антивирусы, ОС и бразуеры: [Электронный ресурс] // Антивирусы и программы безопасности. URL: https://www.comss.ru/page.php?id=11452 (дата обращения 23.06.2023).

9. Microsoft представила сервис модерации контента на основе AI: [Электронный ресурс] // SecurityLab.ru. URL: https://www.securitylab.ru/news/538391.php (дата обращения 23.06.2023).

10. Лиханова Е. Автоматизированный фактчекинг: подходит ли он для соцсетей: // Старший редактор RB.RU. 2020. URL: https://rb.ru/story/auto-fact-checking/ (дата обращения 23.06.2023).

Статья представлена научным руководителем,
старшим преподавателем кафедры ИНиРЯ СПбГУТ Маршевой Н. В.

УДК 004.318

В. В. Сиваков (студент гр. к-631, СПбКТ)

ИССЛЕДОВАНИЕ ПРИЧИН ЗАМЕДЛЕНИЯ ТЕМПОВ РОСТА ВЫЧИСЛИТЕЛЬНОЙ МОЩНОСТИ МИКРОПРОЦЕССОРОВ

В данной статье рассматриваются основные причины замедления темпов роста вычислительной мощности и производительности микропроцессоров, созданных на основе кремниевых транзисторов. Проанализированы причины использования кремния в современных микропроцессорах. Определены особенности внедрения аналогов для замены кремния в вычислительной технике. Представлены некоторые перспективные концептуальные методы решения проблемы замедления темпов роста вычислительной мощности и производительности микропроцессоров.

микропроцессор, кремний, транзисторы, производство, производительность.

Производители новых технологий сталкиваются с проблемой уменьшения размера транзистора [1], которое вызывает ряд технических проблем, таких как увеличение сопротивления и выделение тепла. [2] Эта проблема решается производителями за счет изменения формы затвора и уменьшения других частей транзистора. [3] Однако, необходимо учитывать, что длина канала транзисторов часто больше фактической и каждый производитель считает нанометры по-своему.

Основные проблемы использования кремния в микропроцессорах

● Эффект «узкого канала»

Эффект «узкого канала» проявляется в транзисторах с очень маленькими размерами канала (менее 100 нм), где электроны начинают испытывать сильное взаимодействие с поверхностью канала, вызывая изменение электрических свойств транзистора [4].

● Влияние тепла

Кремниевые транзисторы генерируют большое количество тепла во время работы, что может привести к повреждению их структуры, если они не охлаждены должным образом.

● Неоднородности в производстве

Малейшие отклонения в процессе производства кремниевых транзисторов могут привести к значительным изменениям в их электрических свойствах, что может привести к неисправностям в их работе.

● Низкая подвижность электронов

Подвижность электронов в кремнии относительно низка, что может приводить к низкой скорости работы транзисторов.

● Ограниченная масштабируемость

Транзисторы на основе кремния имеют ограниченные возможности для уменьшения их размеров, что может привести к ограничениям в их использовании в более сложных устройствах.

Все эти проблемы являются актуальными для кремниевых транзисторов и требуют дальнейших исследований и разработок для их решения.

Способы решения проблем

Переход на более оптимизированные архитектуры микропроцессоров

Архитектура микропроцессора влияет на его внутреннюю структуру и инструкции [5]. Существуют разные архитектуры, такие как x86, ARM и Power, каждая со своими преимуществами и недостатками. Выбор правильной архитектуры может повлиять на производительность, эффективность, совместимость и возможности устройства. Важность архитектуры микропроцессоров заключается в определении возможностей и ограничений устройства. Использование ИИ может помочь оптимизировать архитектуру.

Переход на многочиплетную компоновку микропроцессоров.

Компании начинают использовать многочиповые решения вместо увеличения плотности транзисторов на одном чипе. Многочиплетная компановка позволяет разделить функции микропроцессора на несколько чипов: один обрабатывает данные, другой управляет памятью, третий - вводом-выводом. Это увеличивает производительность и масштабируемость, а также уменьшает энергопотребление. Однако многочиповая компоновка может привести к более высокой стоимости и более сложной разработке, и производству микропроцессоров, а также вызвать проблемы с синхронизацией работы между различными чипами. В целом, переход на многочиплетную компоновку микропроцессоров – это важный шаг для улучшения их производительности и функциональности.

Использование аналогов кремния в производстве микропроцессоров

Дисульфид молибдена (MoS2) – полезный материал для производства электронных устройств. Он используется в качестве тонкого пленочного покрытия на кремниевых чипах, чтобы уменьшить истирание и повысить производительность благодаря своим уникальным свойствам, таким как низкий коэффициент трения, высокая термическая стабильность и отличная изоляция. MoS2 также дешев и легко доступен. Он может быть использован для создания транзисторов следующего поколения с очень малыми размерами, что делает его полезным для производства различных электронных устройств. Однако, миниатюрные транзисторы на его основе производить массово пока нельзя. (табл. 1).

Углеродные нанотрубки – тонкие цилиндрические структуры из атомов углерода, которые могут использоваться для создания более эффективных и мощных микропроцессоров. Производство нанотрубок из углерода является сложным и дорогостоящим процессом, требующим дополнительных исследований и разработок для оптимизации и снижения стоимости. (табл. 1). Несмотря на это, потенциальные преимущества нанотрубок из углерода в микропроцессорах стимулируют дальнейшие исследования и разработки.

За последние годы *карбид кремния (SiC)* стал широко используемым материалом для создания полупроводниковых устройств благодаря своим преимуществам: высокой термостабильности, устойчивости к высоким температурам, высокой скорости электронного переноса и устойчивости к радиационному воздействию. (табл. 1). Такие материалы позволяют производить микропроцессоры, работающие на более высоких частотах и с более высокой эффективностью [6]. Кроме того, SiC идеально подходит для космических и военных приложений.

Графен – это слой углерода с уникальными свойствами, такими как высокая электропроводность и прочность. Графен может использоваться в качестве транзисторов и управлять электрическим током на высоких частотах. Его размер позволяет создавать более компактные микропроцессоры. Однако его использование в микропроцессорах вызывает трудности, так как графен не обладает запретной зоной. Существуют различные методы использования графена в микропроцессорах, например, замена кремния графеном в качестве канала в транзисторах или создание графеновых листов для микросхем. Экспериментальные транзисторы с графеном наблюдают слишком большие токи утечки, что делает их непрактичными для использования [7, 8] (табл. 1).

ТАБЛИЦА 1. Сравнение материалов по основным свойствам

Материал	Коэффициент трения	Т режим, °С	Доступность
Дисульфид молибдена	0,05 – 0,07	до +400	Высокая
Углеродные нанотрубки	0,01 – 0,1	до +1000	Очень низкая
Карбид кремния	0,01 – 0,3	до +600	Высокая
Графен	0,03 – 0,4	до +300	Низкая, растёт
Кремний	0,04 – 0,5	до +125	Высокая

В статье рассмотрен вопрос о темпах прогресса в сфере микропроцессоров и решении проблем с ними. Вероятно, выходом из сложившейся ситуации могут стать новые архитектуры, многочиплетные компоновки микропроцессоров, использование аналогов кремния в производстве микропроцессоров. Из данной статьи можно сделать вывод, что поиск замены кремния и решение его проблем продолжается параллельно с его развитием [8].

Список используемых источников

1. В университете Беркли создали транзистор размером в нанометр // Habr URL: https://habr.com/ru/articles/369753/

2. Сметюх Н. П., Степанов Л. Г. Причины замедления темпов роста вычислительной мощности процессоров // Морские технологии: проблемы и решения – 2022 : сборник статей участников научно-практической конференции, Керчь, 25–29 апреля 2022 года. Керчь : ФГБОУ ВО «Керченский государственный морской технологический университет», 2022. С. 68–70. EDN AGRVYJ.

3. Mknews «Конец эпохи кремния. Процессоры будущего» [Электронный ресурс] // pikabu. 2021. URL: https://itigic.com/ru/why-silicon-is-material-from-which-processors-are-made/

4. Денисенко В. Новые физические эффекты в нанометровых МОП-транзисторах [Электронный ресурс] // Компоненты и Технологии. 2009. № 101. URL: https://eposlink.com/ru/catalog/library/elibrary/book/komponenty_i_tehnologii-2147/publication/107072/

5. Johan De Gelas Современные микропроцессоры. Часть первая: секреты высокой производительности [Электронный ресурс] // Ferra. 2004. URL: https://www.ferra.ru/review/computers/s25410.htm

6. Григорьев А. Я., Мышкин Н. К. О смазках вообще [Электронный ресурс] // Химия и жизнь. 2014. № 1. URL: https://elementy.ru/nauchno-populyarnaya_biblioteka/432392/Tverdye_smazki

7. Иванова Е. С. Графен и его применение в электронике [Электронный ресурс] // Материалы XII Международной студенческой научной конференции «Студенческий научный форум». URL: https://scienceforum.ru/2020/article/2018020517

8. Мэтт Миллс. Почему кремний – это материал, из которого сделаны процессоры [Электронный ресурс] // itigic. 2020. URL: https://itigic.com/ru/why-silicon-is-material-from-which-processors-are-made/

Статья представлена научным руководителем, преподавателем СПбКТ Минеевой В. Д.

УДК 342.97

А. С. Третьякова (студентка гр. ИБС-01, СПбГУТ)

ВОПРОСЫ НАРУШЕНИЯ ЗАКОНОДАТЕЛЬСТВА РФ В ГИС ЖКХ

Статья посвящена нарушениям в государственной информационной системе жилищно-коммунального хозяйства и раскрывает эту тему анализ статьи13.19.2 кодекса Российской Федерации об административных правонарушениях, в которой определяется ответственность за неразмещение информации, размещение информации не в полном объеме или размещение недостоверной информации в государственной информационной системе жилищно-коммунального хозяйства. Статья13.19.2 направлена на обеспечение прав и свобод граждан, а также защиту интересов государства в цифровой сфере.

информация, государственная информационная система, законодательство.

Разобраться в вопросах нарушения законодательства РФ в ГИС ЖКХ поможет статья 13.19.2 КоАП РФ Неразмещение информации, размещение информации не в полном объеме или размещение недостоверной информации в государственной информационной системе жилищно-коммунального хозяйства. Данная статья является одной из многих, которые регулируют ответственность за различные виды нарушений в сфере связи, информационных технологий и массовых коммуникаций. Таким образом, эта статья входит в общую систему законодательства, которая направлена на обеспечение прав и свобод граждан, а также защиту государственных интересов в цифровой сфере.

Важно понимать, что нарушения данной статьи могут иметь серьезные последствия, в том числе привести к судебным разбирательствам и штрафным санкциям.

По сути, эта статья предписывает всем участникам процесса размещения информации в государственной информационной системе жилищно-коммунального хозяйства соблюдать законодательные требования к размещаемой информации, ее объему и достоверности.

Например, если информация не размещена в полном объеме, то это может создать трудности для граждан и организаций при получении коммунальных услуг, а также привести к недополучению необходимой информации.

Если же информация, размещенная в государственной информационной системе жилищно-коммунального хозяйства, оказалась недостоверной, то это может привести к нарушению прав и интересов граждан и организаций, а также к неоправданным расходам на получение неверной информа-

ции. В связи с этим, статья 13.19.2 КоАП РФ нацелена на обеспечение защиты прав и интересов граждан и организаций, а также на повышение качества предоставляемых коммунальных услуг.

Государственная информационная система жилищно-коммунального хозяйства является современной электронной платформой, которая предназначена для сбора, хранения, обработки и распространения информации о коммунальных услугах и ресурсах, а также об организациях и предприятиях, которые осуществляют предоставление этих услуг.

В соответствии с действующим законодательством, государственная информационная система жилищно-коммунального хозяйства должна содержать актуальную и достоверную информацию.

Также есть сведения, которые подлежат размещению в ГИС ЖКХ, выделены некоторые из них:

1. Информация об управляющей организации, ТСЖ или ЖСК (носит довольно общий характер, включает адрес, телефон, ИНН и иные подобные сведения).

2. Информация о перечне предоставляемых работ и услуг, их стоимости.

3. Информация об объеме, о качестве коммунальных услуг, предоставленных собственникам и пользователям помещений.

4. Информация о размере платы за жилое помещение.

5. Информация о лицевых счетах, присвоенных собственникам и пользователям жилых (нежилых) помещений в многоквартирных домах.

6. Информация об отчетности ТСЖ/ЖСК.

Согласно статье 13.19.2, для управляющих организаций и ТСЖ предусмотрена ответственность за 4 вида нарушений по размещению информации[1]:

- неразмещение информации;
- нарушения порядка, способов и сроков размещения;
- размещение информации не в полном объеме;
- размещение недостоверных сведений.

Для лиц, которые выполняют функции администраторов общего собрания, предусмотрены следующие санкции: физическим лицам может быть вынесено предупреждение или наложен административный штраф в размере от трех тысяч до пяти тысяч рублей, а должностным лицам – от пяти тысяч до десяти тысяч рублей.

Субъектами административного правонарушения являются соответствующие поставщики информации в ГИС ЖКХ и их должностные лица [2].

Субъективная сторона правонарушения может выражаться как в виде умысла, так и в виде неосторожности.

Для лиц, осуществляющих поставки ресурсов, необходимых для предоставления коммунальных услуг, предоставляющих коммунальные услуги

и/или занимающихся управлением многоквартирных домов, а также для других организаций, которым требуется в соответствии с законодательством Российской Федерации размещать информацию в государственной информационной системе жилищно-коммунального хозяйства, могут быть применены следующие меры: должностным лицам может быть наложен административный штраф в размере от пяти тысяч до десяти тысяч рублей или выдано предупреждение.

Для лиц, ранее подвергнутых административному наказанию за аналогичное административное правонарушение, влечет наложение административного штрафа в размере от пятнадцати тысяч до двадцати тысяч рублей.

Для лучшего понимания темы приведены судебные дела:

1. Верховный суд вынес решение от 5 сентября 2017 г. № 305-АД17-11531 (неисполнение истцом (УО) обязанности по размещению в ГИС ЖКХ, данных о многоквартирных домах – наложен штраф в размере 30 тыс. рублей).

2. Арбитражный апелляционный суд вынес решение от 21 февраля 2018 г. № 09АП-67644/2017 (УО не разместила в системе ГИС ЖКХ информацию о расторжении договора МКД, своевременно не представила заявление о внесении изменений в реестр лицензий города Москвы).

3. (из новостей) Мировой суд оштрафовал председателя ТСЖ города Кизляра за непредоставление сведений в государственной информационной системе ЖКХ. В особенности, не размещена информация о поставщиках коммунальных ресурсов, о перечне предоставляемых услуг по управлению общим имуществом в доме, а также ценах на услуги.

Важно подчеркнуть, что органы государственного жилищного инспектирования (ГЖИ) не имеют постоянной задачи обнаруживать лиц, которые не размещают информацию в ГИС ЖКХ. Обычно проверки управляющих организаций и ТСЖ проводятся на основе жалоб граждан, а не в рамках плановых мероприятий.

Действительно, судебная практика относительно управляющих компаний, ТСЖ и других форм товариществ и кооперативов в отношении статьи 13.19.2, которая касается неразмещения информации, представляет собой обширное поле. Однако возможно заменить административную ответственность и штраф на предупреждение.

Существуют несколько вариантов, которые могут помочь избежать ответственности и получить предупреждение:

1. Регулярно следить за изменениями законодательства в сфере ЖКХ, уделяя внимание также изменениям, касающимся ГИС ЖКХ.

2. Принимать все необходимые меры для соблюдения требований законодательства относительно ГИС ЖКХ и документировать все случаи, когда выполнение этих требований не удалось, чтобы иметь возможность доказать свою невиновность при проверке. Это может включать переписку с технической поддержкой ГИС ЖКХ и записи звонков на горячую линию.

3. Доверить размещение информации в ГИС ЖКХ профессионалам.

4. Вносить информацию в систему в полном объеме и в установленные сроки.

В любом случае, конкретные варианты избежания ответственности и получения предупреждения могут различаться в зависимости от обстоятельств и применяемого законодательства, поэтому рекомендуется обратиться за консультацией к юристу, специализирующемуся в данной области.

В результате рассмотрения статьи 13.19.2 КоАП РФ, можно сказать, что она играет важную роль в регулировании правил размещения информации в государственной информационной системе жилищно-коммунального хозяйства. Эта статья направлена на обеспечение защиты прав граждан и интересов государства в цифровой сфере, а также на предотвращение негативных последствий от неразмещения информации, размещения информации не в полном объеме или размещения недостоверной информации.

Следует помнить, что нарушение данной статьи может привести к серьезным штрафным санкциям, а также к другим негативным последствиям, связанным с судебными разбирательствами и утратой доверия со стороны потребителей.

Поэтому важно соблюдать требования закона, активно развивать свои знания и навыки в области цифровых технологий и информационной безопасности, чтобы минимизировать риски нарушения закона и обеспечить эффективное функционирование государственной информационной системы жилищно-коммунального хозяйства.

Список используемых источников

1. Кодекс Российской Федерации об административных правонарушениях от 30.12.2001 N 195-ФЗ (ред. от 28.04.2023, с изм. от 17.05.2023).

2. Комментарии к статье 13.19.2. URL: https://www.zakonrf.info/koap/13.19.2/?ysclid=li22dglv28465994043 (дата обращения 21.05.2023).

Статья представлена научным руководителем, доцентом кафедры ЗСС СПбГУТ, кандидатом технических наук Сахаровым Д. В.

УДК 543.42

А. В. Фраз (аспирант кафедры ФиЛС, СПбГУТ)

ПРИМЕНЕНИЕ СПЕКТРОАНАЛИЗАТОРОВ В СИСТЕМАХ СВЯЗИ С ТЕХНОЛОГИЕЙ DWDM

Оптические методы передачи информации широко используются и очень востребованы в телекоммуникациях. Одна из перспективных технологий, используемая в настоящее время на транспортных сетях, это технология DWDM (Dense Wavelength Division Multiplexing – плотное спектральное уплотнение), позволяющая передавать большие объёмы данных на дальние расстояния по одному волокну. Для контроля параметров оптических компонентов и оценки качества связи в волоконно-оптических системах с DWDM необходимы анализаторы оптического спектра. В этой статье проанализированы виды измерений, которые проводятся в системах DWDM с применением спектроанализаторов, сформулированы требования к спектроанализаторам для DWDM-систем, выявлены недостатки существующих приборов и намечены пути их преодоления.

DWDM, Dense Wavelength Division Multiplexing, плотное спектральное уплотнение, анализаторы спектра, спектроанализаторы, характеристики DWDM-систем, разрешающая способность, рабочий спектральный диапазон DWDM, спектральные измерения.

Постоянно появляющиеся сервисы и приложения требуют всё большого объёма передаваемого трафика и всё выше скорости передачи данных в телекоммуникациях. Такую задачу решает технология WDM (*Wavelength Division Multiplexing*) – технология спектрального мультиплексирования, которая уже почти 30 лет является основной технологией построения магистральных волоконно-оптических сетей связи. И одна из разновидностей WDM – это технология DWDM (*Dense Wavelength Division Multiplexing*) – плотное спектральное уплотнение, позволяющая передавать большие объёмы данных на дальние расстояния по одному волокну. Расстояние между несущими в DWDM-системах может составлять 25...200 ГГц, в современных сетях наиболее часто используется сетка каналов с шагом 50 ГГц. Для передачи используются прежде всего спектральные диапазоны C (1530...1565 нм), S (1460...1530 нм) и L (1565...1625 нм) [1].

Для контроля параметра оптических компонентов и оценки качества связи в волоконно-оптических системах с DWDM необходимы анализаторы оптического спектра. Одной из основных характеристик, измеряемых анализатором спектра в системе DWDM, является коэффициент оптического отклонения (*Optical Rejection Ratio*, ORR) [2]. ORR характеризует максимальное отношение сигнал/шум, которое анализатор спектра может измерить в заданном диапазоне при максимальном значении сигнала. При использовании волокон с плотно расположенными каналами (ниже 50 ГГц)

важность ORR возрастает. Анализатор спектра с недостаточным ORR не сможет уловить большинство деталей спектрального профиля реального сигнала DWDM.

Следующая важная характеристика – это отношение сигнал/шум (*Optical Signal-to-Noise Ratio*, OSNR). При использовании скоростных систем STM-64/OC-192 (10 Гбит/с) возникнет потребность в измерении больших величин отношения сигнал/шум, также для увеличения быстродействия измерений, потребуется увеличить точность измерений OSNR в 4–5 раз. Минимальный уровень измерений OSNR колеблется в диапазоне от 21 до 35 дБ. Первоклассный OSA гарантирует ORR по крайней мере 50 дБ в пике 0,4 нм.

Диапазон длин волн, который может анализировать спектрометр, играет важную роль. Сегодня многие системы DWDM внедрены в соответствии с рекомендациями G.692 МСЭ. То есть все каналы находятся в зоне C – от 1530 до 1565 нм, именно в этом диапазоне длин волн эрбиевые усилители работают наиболее эффективно. Но в тоже время большим спросом пользуется спектроанализатор, который может анализировать длины волн в следующих диапазонах S и L. Например, такой универсальный анализатор подходит для оптического канала наблюдения (*Optical Supervisory Channel*, OSC), используемого для мониторинга системы DWDM. Таким каналам присваивается одна из следующих длин волн: 1490, 1510 или 1625 нм.

Более высокое разрешение анализатора спектра по длине волны позволяет различать сигналы с одинаковыми длинами волн. В основном это определяется оптическими характеристиками фильтра анализатора спектра. Более узкие фильтры обеспечивают лучшее разрешение анализатора спектра. Для достижения высокого разрешения высококлассные модули спектрального анализа используют двойное пропускание вместе с высококачественными дифракционными решетками.

Высокое значение разрешения при длине волны 0,05 нм обеспечивает хорошую ORR и повышает производительность анализатора спектра при измерениях OSNR для систем с близко расположенными каналами. Такой OSA будет способен анализировать системы DWDM с разносом каналов 50 ГГц (0,4 нм).

Таким образом для спектроанализаторов, работающих в DWDM-системах важны следующие характеристики:

- ORR;
- OSNR;
- спектральный диапазон;
- фильтры;
- разрешающая способность;
- интервалы каналов.

Многие из этих характеристик связаны, поэтому улучшая одну характеристику, можно приди к улучшению и другой характеристики, как например ORR зависит от OSNR, а разрешающая способность зависит от фильтров.

Из современного телекоммуникационного оборудования для систем DWDM следует рассмотреть Спектроанализатор OPM, разработанный компанией T8 [3], который работает в С-диапазоне 1528–1558 нм и имеет канальную сетку с шагом 50 ГГц (рис. 1).

Рис. 1. Спектроанализатор OPM компании T8

А также оборудование компании EXFO, как например FTB-5600 [4], который имеет рабочий диапазон от 1520 до 1580 нм, и FTB-5230S/-OCA [5], в свою очередь который уже работает в диапазоне 1250–1650 нм, что позволяет ему работать не только о всём диапазоне DWDM, но и в диапазоне CWDM. Разрешающая способность у FTB-5230S/-OCA ≤ 0,10 нм, а ширина каналов от 33 до 200 ГГц (рис. 2).

Подводя итог можно сделать вывод, что следует улучшать разрешающую способность спектроанализаторов для систем DWDM и уменьшать ширину каналов для передачи большего количества информации с высокой скоростью и на большие расстояния. Также большим преимуществом устройства будет возможность работы в большом диапазоне длин волн, так как хоть и используются спектроанализаторы работающие с одним диапазонам длин волн DWDM, работа во всём диапазоне DWBЬ и даже диапазон CWDM позволяет использовать спектроанализатор во многих направлениях телекоммуникаций.

Рис. 2. FTB-5230S/-OCA компании EXFO

Список используемых источников

1. Трещиков В. Н., Листвин В. Н. DWDM-системы, четвёртое издание. М. : ТЕХНОСФЕРА, 2021. 420 с.

2. Серегин З. Тестирование DWDM [Электронный ресурс] // Журнал сетевых решений/LAN. 2001. № 02. URL: https://www.osp.ru/lan/2001/02/134650

3. Т8 [Электронный ресурс] // Технические характеристики спектроанализатора ОРМ, производства Т8. Режим доступа URL: https://t8.ru/?page_id=5162 (дата обращения 10.05.2023).

4. EXFO [электронный ресурс] // Технические характеристики FTB-5600, производства EXFO. Режим доступа URL: https://www.exfo.com/en/products/field-network-testing/dispersion-analysis/ftb-5600/ (дата обращения 10.05.2023).

5. EXFO [электронный ресурс] // Технические характеристики FTB-5230S/-OCA, производства EXFO. Режим доступа URL: https://www.exfo.com/en/discontinued-products/ftb-5230s-oca/ (дата обращения 10.05.2023).

Статья представлена научным руководителем, заведующей кафедрой ФиЛС СПбГУТ, кандидатом технических наук, доцентом Былиной М. С.

УДК 004.056.5

С. Г. Баженов, А. В. Ефимова, С. А. Павлюченко,
Л. А. Шумских (студенты гр. ИКПИ-12, СПбГУТ)

КРИПТО-ПРЕСТУПЛЕНИЯ И ТЕХНИЧЕСКИЕ МЕТОДЫ ИХ ПРЕСЕЧЕНИЯ: РАЗВИТИЕ ТЕХНОЛОГИЙ ДЛЯ БОРЬБЫ С КИБЕРУГРОЗАМИ

С развитием технологий и распространением цифровой сферы, киберпреступность и крипто-преступления становятся все более серьезной угрозой. Киберугрозы могут причинить значительный ущерб компаниям, организациям и даже государствам. Однако, также развиваются и технические методы, используемые для пресечения крипто-преступлений. В данной статье мы рассмотрим основные виды крипто-преступлений, а также современные технологические подходы к борьбе с ними.

киберугрозы, фишинг, фарминг, мошенничество, криптовалютные атаки.

1 Крипто-преступления

Крипто-преступления представляют собой преступные действия, осуществляемые с использованием информационных технологий и криптографических методов. Рассмотрим некоторые из них:

- Фишинг и фарминг

Фишинг и фарминг – это виды киберпреступлений, связанных с обманом пользователей с целью получения конфиденциальной информации или доступа к личным данным. Часто злоумышленники создают поддельные веб-сайты или отправляют фальшивые электронные письма, подражающие официальным источникам, чтобы выманить у пользователей логины, пароли и другую личную информацию.

- Вредоносные программы

Распространение вредоносных программ, таких как вирусы, трояны и шпионское ПО, является распространенным способом киберпреступников получить незаконный доступ к системам и данным. Эти программы могут нанести серьезный ущерб, включая кражу личных данных, блокировку системы или даже контроль над ней.

- Киберворовство и мошенничество

Киберворовство и мошенничество включают в себя кражу финансовых данных, кредитных карт и других ценных активов. Киберпреступники могут использовать различные методы, включая кражу и подделку идентификационных данных, для получения незаконного доступа к банковским счетам и финансовым ресурсам жертв.

- Криптовалютные атаки

С развитием криптовалют и блокчейн технологий, киберпреступники также обратили свое внимание на эту область. Криптовалютные атаки включают в себя взлом криптовалютных бирж, майнинг с использованием вычислительных ресурсов других пользователей и другие атаки на криптовалютные системы. Одним из примеров таких атак является 51 %-ная атака, при которой злоумышленник получает контроль над большинством вычислительных мощностей блокчейн сети (рис. 1).

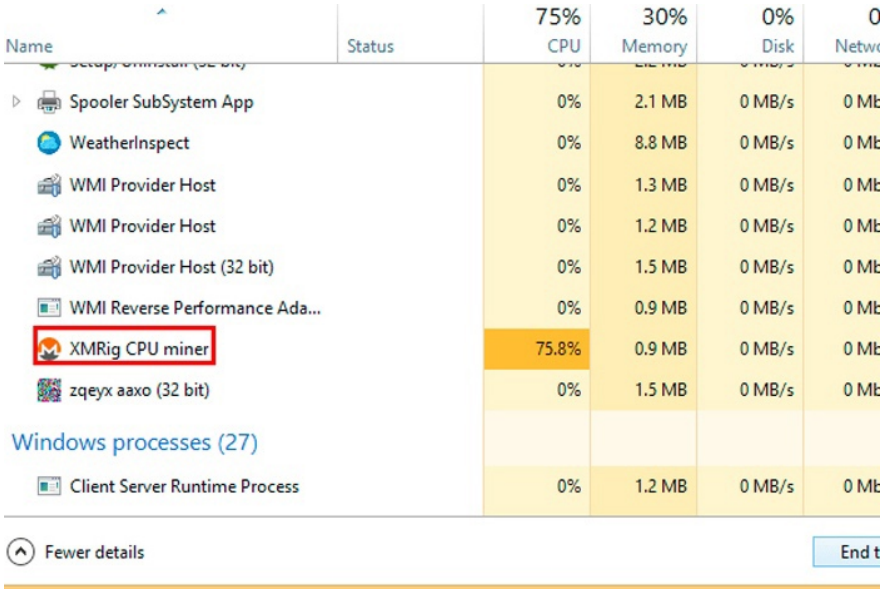


Рис. 1. Пример вредоносной программы использующей ресурсы компьютера для майнинга

2 Технические методы борьбы с крипто-преступлениями

Для пресечения и предотвращения крипто-преступлений разрабатываются и применяются различные технические методы. Рассмотрим некоторые из них:

- Многофакторная аутентификация

Многофакторная аутентификация требует от пользователя предоставить несколько независимых факторов для подтверждения личности, таких как пароль, биометрические данные (отпечаток пальца, сканирование сетчатки глаза) или одноразовый код, полученный через SMS.

- Шифрование данных

Шифрование данных – это процесс преобразования информации в нечитаемый вид с использованием специальных алгоритмов. Шифрование обеспечивает конфиденциальность информации и защиту от несанкционированного доступа. Сильные шифровальные алгоритмы, такие как AES (*Advanced Encryption Standard*), применяются для защиты конфиденциальности данных в ходе их передачи и хранения.

- Системы обнаружения вторжений (IDS)

Системы обнаружения вторжений используются для выявления аномалий и вторжений в компьютерные системы. IDS мониторят сетевой трафик и системные журналы, анализируют их с помощью специальных алгоритмов и обнаруживают подозрительные активности или попытки несанкционированного доступ к системе. После обнаружения потенциальной угрозы IDS может принять соответствующие меры, такие как блокировка доступа или оповещение администратора системы.

- Блокчейн технологии

Блокчейн технологии представляют собой распределенную и непрозрачную систему записи транзакций. Они обеспечивают прозрачность, целостность и безопасность данных. В криптовалютных системах блокчейн используется для проверки и регистрации транзакций, что делает их более надежными и защищенными от манипуляций и подделок (рис. 2).

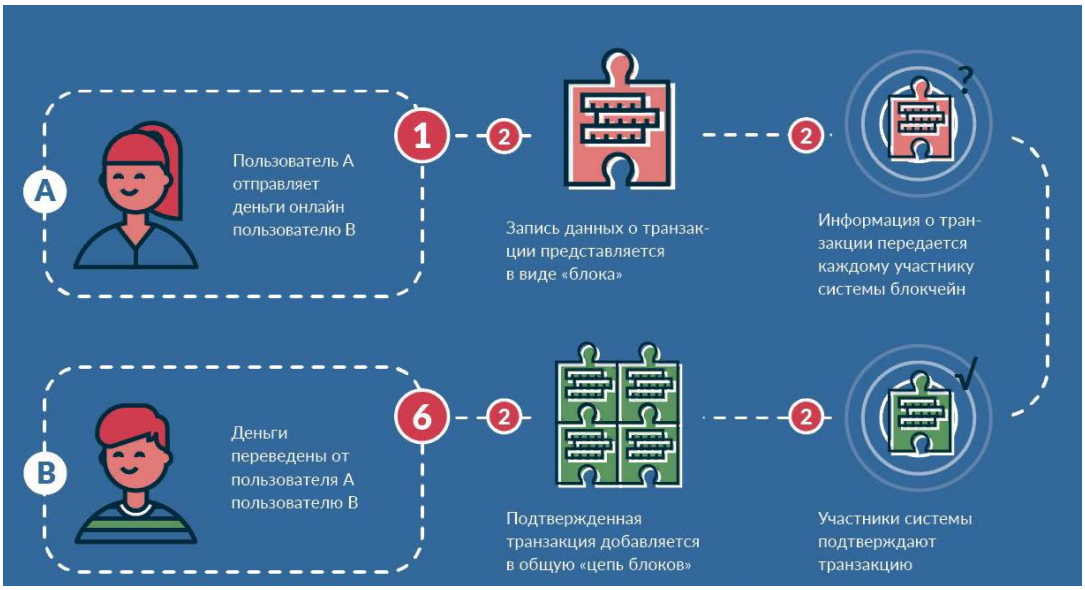


Рис. 2. Схема работы блокчейн системы

- Искусственный интеллект и машинное обучение

Анализ поведения пользователей с помощью алгоритмов машинного обучения позволяет выявить аномальное поведение и своевременно обнаружить потенциальные угрозы. Автоматизированный поиск уязвимостей с применением искусственного интеллекта позволяет сканировать и анализировать системы на наличие уязвимостей, что помогает предотвратить возможные атаки.

Заключение

Крипто-преступления представляют серьезную угрозу в современном цифровом мире, однако развитие технических методов борьбы с ними позволяет эффективно пресекать и предотвращать подобные преступления.

Многофакторная аутентификация, шифрование данных, системы обнаружения вторжений, блокчейн технологии, искусственный интеллект и машинное обучение – все эти средства совместно способствуют созданию надежных систем защиты и обеспечению безопасности цифровых средств передачи и хранения информации. Дальнейшее развитие технологий и совершенствование методов борьбы помогут сделать нашу онлайн-среду более безопасной и надежной для всех ее пользователей. Преодоление криптопреступлений требует постоянного сотрудничества между разработчиками технических решений, правоохранительными органами и пользовательским сообществом для обеспечения безопасности и защиты в сети.

Список используемых источников

1. Авчаров И. В. Борьба с киберпреступностью // Информатизация и информационная безопасность правоохранительных органов. XI межд. конф. М., 2012. С. 191–194.

2. Айков Д., Сейгер К., Фонсторх У. Компьютерные преступления. Руководство по борьбе с компьютерными преступлениями. М. : Мир, 2014. 351 с.

3. Антонос Г. А. Международные изменения права киберпространства // Право и информатизация общества : сб. науч. тр. М. : ИНИОНРАН, 2012. С. 174–186.

Статья представлена научным руководителем,
ассистентом кафедры ЗСС СПбГУТ Катасоновым А. И.

УДК 004.588

М. М. Байзигитов, С. П. Пенькова-Печерская, Д. М. Соловьев
(студенты гр. ИКПИ-13, СПбГУТ)
Ю. В. Одинцова (студентка гр. ИКПИ-14, СПбГУТ)

ВВЕДЕНИЕ В СТАРТАП И РАЗРАБОТКА ЕГО MVP

В наше время все большую популярность приобретает самостоятельное обучение, благодаря доступности средств для удаленной работы и наличию большого количества материалов в интернет. Но, начиная обучение в какой-либо сфере, молодые люди сталкиваются со сложностями поиска качественной информации. Решить данную проблему можно созданием платформы, сотрудничающей напрямую с вузами и систематизирующей большое количество материалов.

онлайн-обучение, MVP, база лекционных материалов, лекции, образование.

Проблематика и основные характеристики проекта

В рамках тренинга МСЭ на тему стартапов в целях получения опыта разработки проектов данного типа требовалось выбрать проблематику, разработать идею проекта и реализовать варианты презентации проекта для дальнейшего питчинга.

Нашей группой было выбрано направление онлайн-обучения и предоставления лекционных материалов не только для учащихся вузов, но и для всех желающих. Была определена следующая **проблематика**:

- отсутствие централизованной базы данных с теоретическими материалами университетов страны;
- отсутствие платформы, сотрудничающей напрямую с вузами в целях предоставления учебных материалов,
- отсутствие развернутой информации по узким направлениям специализаций,
- недостаток учебных и научно-исследовательских материалов для учебного процесса.

Далее в целях **решения** данных проблем были предложены следующие варианты:

- создание платформы, сотрудничающей напрямую с вузами, на которой преподаватели могут выкладывать учебные материалы по различным дисциплинам;
- систематизация большого количества материала на одном интернет ресурсе;
- предоставление качественного модерирования контента на сервисе.

Данный проект в свою очередь способен выполнить следующие **миссии**:

- улучшить образование страны;
- предоставить качественные материалы от ведущих преподавателей;
- предоставить быстрый поиск информационных ресурсов.

Данный проект имеет следующую **целевую аудиторию**: студенты; люди, желающие повысить свою квалификацию; преподаватели;

Способами продвижения данного проекта являются: личная коммуникация, продвижение администрацией университетов, реклама в соц. сетях.

Основными существующими конкурентами станут платформы: openedu, coursera, stepik, studfiles, swiftbook. Преимуществом данного проекта над конкурентами является прямой контакт с вузами, монетизация контента в пользу преподавателей. Основным способом монетизации контента является реклама и система подписок на контент платформы.

Масштабируемость заключается в быстром наборе популярности за счет коммуникации между преподавательским составом и студентами. Таким образом, всё больше вузов будут заинтересованы в совместной с сервисом работе.

Требуемые ресурсы: человеческие – разработка платформы и создание контента на ней, финансовые – аренда, выплаты персоналу. Основные затраты будут касаться разработки платформы, аренды хостинга, реклама, заработные платы причастным.

Риски, связанные с созданием проекта такого рода: нулевой оборот, неокупаемые затраты на рекламу, некачественный контент, низкий уровень заинтересованности в материалах.

Основные KPI

В рамках тренинга была проведена работа над оценкой производительности стартапа. Далее приведены характеристики, подходящие для оценки нашего типа стартапа, а также их описание.

1) SaaS и подписочная бизнес модели

SaaS (программное обеспечение как услуга), например, как Zoom, Adobe, Slack. И подписочная бизнес модель, например, как Нетфликс. Обычно подписка продается бизнесу (B2B сегмент), и в идеале по годовым контрактам. Особенность в регулярности дохода (отметим, что разовый доход не включается в ARR и MRR).

ARR (годовой регулярный доход) – это мера элементов дохода, которые носят повторяющийся характер. Он должен исключать разовые поступления.

MRR – регулярно повторяющийся месячный доход.

1. New MRR – регулярный ежемесячный доход от новых клиентов.

2. Net New MRR – сумма дохода от новых клиентов и от ранее существующих минус отток дохода от ушедших клиентов.

3. Churned MRR – потерянный MRR из-за отказа клиентов продлевать подписку.

4. MRR Churn Rate – показатель оттока MRR. Фиксируется, если клиенты не продлевают подписку или переходят на более дешевый тариф.

5. MRR Growth Rate – процентное увеличение дохода от месяца к месяцу, которое отвечает на ключевой вопрос, насколько быстро растет стартап.

Growth Rate (Скорость роста) – измеряется еженедельно или ежемесячно.

Revenue Churn Rate – показатель оттока дохода.

Revenue Customer Rate – показатель оттока клиентов.

Retention Rate (Customer) – показатель удержания клиентов.

Retention Rate (Revenue) – показатель удержания дохода.

Burn Multiple – коэффициент траты денег по отношению к росту выручки (Net New MRR): чем он выше, тем больше стартап тратит денег для достижений каждой единицы роста, соответственно, чем он ниже, тем лучше.

LTV (lifetime value) – прибыль, которую приносит один клиент за всё время работы с компанией.

CAC (customer acquisition cost) – стоимость привлечения клиента. Рассчитывается как сумма всех затрат на маркетинг и продажи, разделенная на количество новых пользователей.

LTV: CAC (Ratio of Lifetime Value to Customer Acquisition Cost) – отношение ценности жизненного цикла клиента к стоимости привлечения клиента.

Gross margin – результат вычитания стоимости проданных товаров из чистой выручки. Валовая прибыль также может быть выражена в процентах, используется при сравнении стартапов разного размера и разных отраслей.

Также, есть метрики пользовательской активности за определенный период: за день (daily active users, DAU), за неделю (weekly active users, WAU) или за месяц (monthly active users, MAU).

● DAU – это количество уникальных пользователей, которые совершали какое-то действие каждый день (в рамках вашего продукта).

● WAU измеряет количество уникальных пользователей, которые совершали действие каждую неделю (в рамках вашего продукта).

● MAU – это количество уникальных пользователей, которые совершали действие в течение месяца (в рамках вашего продукта).

2) Транзакционная бизнес модель

Облегчайте транзакции и получайте комиссию (*fintech and payments businesses*). Чаще всего разовые транзакции, а не повторяющиеся (Часто большой объем с низкой комиссией (обычно 1–3 %)).

1. *Gross Transaction Value (GTV, валовая стоимость сделки)* – общий объем транзакций.

2. *Net Revenue (Чистый доход)* – комиссия за транзакции (часто измеряется в процентах).

3. *User Retention* – удержание пользователей (% клиентов в 1-м месяце, которые совершают покупку во 2-м месяце и т. д.).

4. *CAC (customer acquisition cost)* – стоимость привлечения клиента. Рассчитывается как сумма всех затрат на маркетинг и продажи, разделенная на количество новых пользователей.

3) Usage-based

Бизнес-модель на основе использования продукта, например, использование данных/количества запросов.

1. Ежемесячный доход (который не повторяется).

2. Скорость роста.

3. Revenue Retention (Удержание дохода) – % выручки от клиентов прошлого месяца в этом месяце.

4. Gross Margin (Валовая прибыль) – доход минус себестоимость проданных товаров (Revenue – Cost of Goods Sold (COGS)).

Архитектура проекта

Для работы с высоконагруженной системой был выбран язык Java с фреймворком Spring. В нём имеются компоненты, позволяющие построить отвечающую всем требованиям систему.

Выбор базы данных пал на PostgreSQL, поскольку PostgreSQL – объектно-реляционная БД. Это означает, что в PostgreSQL можно хранить данные в качестве объектов со свойствами. Соответственно данные можно будет хранить в виде объектов Java. На рис. 1 наглядно изображена схема базы данных.

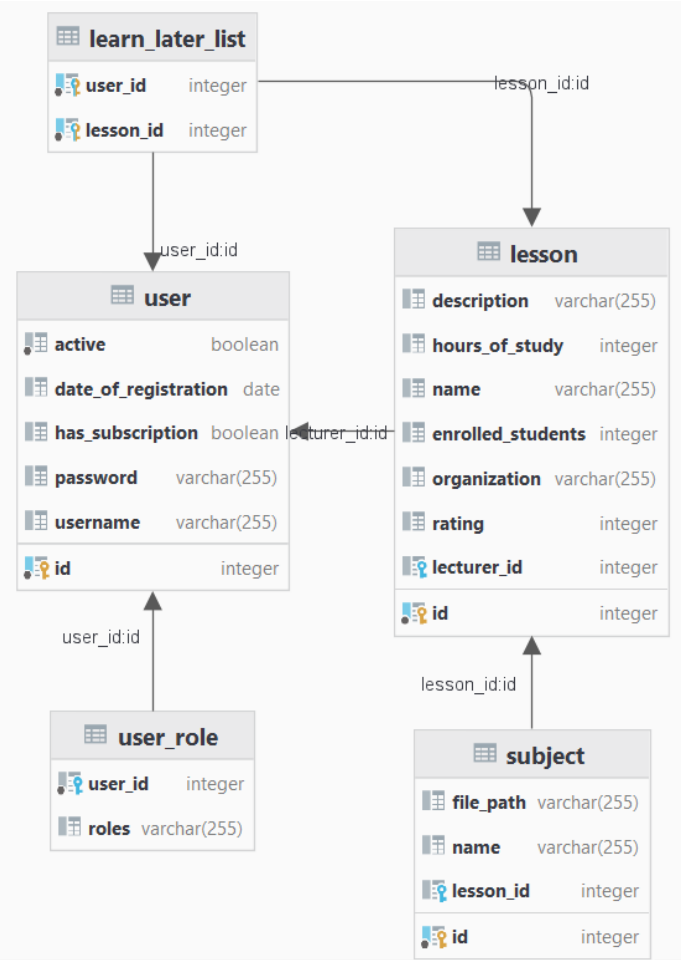


Рис. 1. Схема базы данных

Шаблонизатор Thymeleaf был выбран, поскольку это лёгкий способ связать данные на страницы с данными на стороне бэкенда.

Таким образом в архитектуре проекта используются такие технологии, как: Java, Spring framework; база данных PostgreSQL; Spring security (аутентификации и уровни доступа); Hibernate (связь с базой данных); Hibernate Validator (корректность вводимых данных); Thymeleaf (отображение страниц, шаблонизация).

Список используемых источников

1. Тренинг «Твой телеком-стартап: от идеи до питчинга, или как привлечь инвестора» от Международного союза электросвязи.
2. Новиков Б. А., Горшкова Е. А., Графеева Н. Г. Основы технологий баз данных; под ред. Е. В. Рогова. 2-е изд. М.: ДМК Пресс, 2020. 582 с.
3. Бауэр К., Кинг Г., Грегори Г. Java Persistence API и Hibernate. ДМК Пресс, 2017. 652 с.
4. Уоллс К. Spring в действии. ДМК Пресс, 2015. 752 с. ISBN 978-5-94074-568-6.

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.852

А. А. Баряев, А. А. Курносов, Р. С. Михеев, Д. А. Петров, Е. И. Скоба
(студенты гр. ИКПИ-11, СПбГУТ)

ПРИМЕНЕНИЕ ИССКУСТВЕННОГО ИНТЕЛЛЕКТА В ЗАДАЧАХ ТЕОРИИ ГРАФОВ

Теория графов повсеместно используется в самых различных областях человеческой деятельности: строение сетей связи, маршруты на картах, схемы авиалиний, схемы электроцепей и многое другое. Наиболее востребованной задачей в теории графов является поиск кратчайшего маршрута между клетками лабиринта. Существует множество алгоритмов нахождения кратчайшего маршрута, одним из самых эффективных из них является алгоритм A-Star (A). Учитывая растущую популярность и востребованность нейросетей, актуальным становится вопрос о скорости решения такого рода задач с использованием искусственного интеллекта: можно ли создать нейросеть, решающую задачи быстрее, чем самые эффективные алгоритмы? В статье сформулированы принципы быстрого прохождения лабиринтов с использованием нейросетей. Исследованы преимущества и недостатки использования искусственного интеллекта в прохождении лабиринтов на основании сформулированных в статье принципов.*

искусственный интеллект, графы, линейная алгебра, алгоритм, процедурная генерация, q-learning.

В последнее время все чаще мы слышим новости об искусственном интеллекте. Нет сомнений, что человечество совершило большой скачок в этом направлении за последние годы. Возникает ощущение, что искусственный интеллект является универсальным решением всех проблем и позволяет преодолеть любые трудности. С целью проверки этого утверждения мы приступили к исследованию узкоспециализированной задачи с использованием как классических алгоритмов, так и различных методов искусственного интеллекта. В качестве задачи для сравнения было решено взять задачу из теории графов, а именно задачу прохождения лабиринтов, как достаточно актуальную в наше время, тем более что количество нейросетей, решающих данную задачу, сравнительно мало с количеством классических алгоритмов.

Актуальность данной работы составляет тот факт, что задача прохождения лабиринта имеет гораздо больше применений, чем кажется на первый взгляд. Начиная построением маршрута на карте до ближайшего магазина и заканчивая картой сетей связи, поиском кратчайших маршрутов в сети.

Целью работы является, сравнение возможностей искусственного интеллекта в задачах теории графов с возможностями классических алгоритмов. Выявить, в каких ситуациях рациональнее использовать ИИ, а в каких нет.

В процессе работы была написана программа на PyQt5 в которой есть возможность генерации и прохождения лабиринтов различными способами [1]. Также в программе присутствует возможность собирать различные статистические данные. Все последующие данные были получены из разработанной программы.

Из рис. 1 видно, что самым сложным лабиринтом для прохождения алгоритмом A-Star [1], является лабиринт, сгенерированный алгоритмом итеративной версии поиска по глубине (*backtracking*) [1].

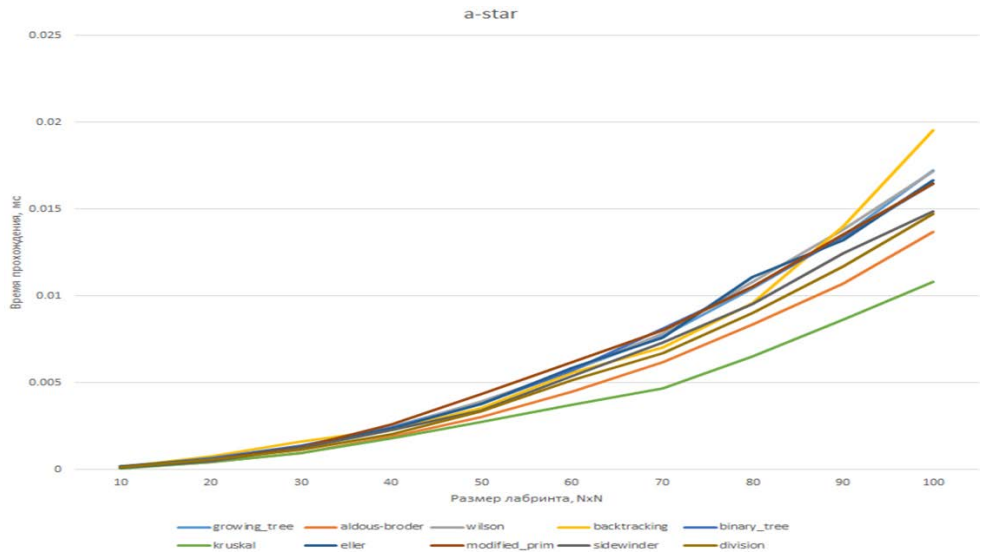


Рис. 1. Зависимость времени прохождения от размера лабиринта для алгоритма A-star

Из рис. 2 видно, что самым сложным лабиринтом для прохождения алгоритмом Дейкстры [1], является лабиринт, сгенерированный алгоритмом бинарного дерева (binary_tree) [1].

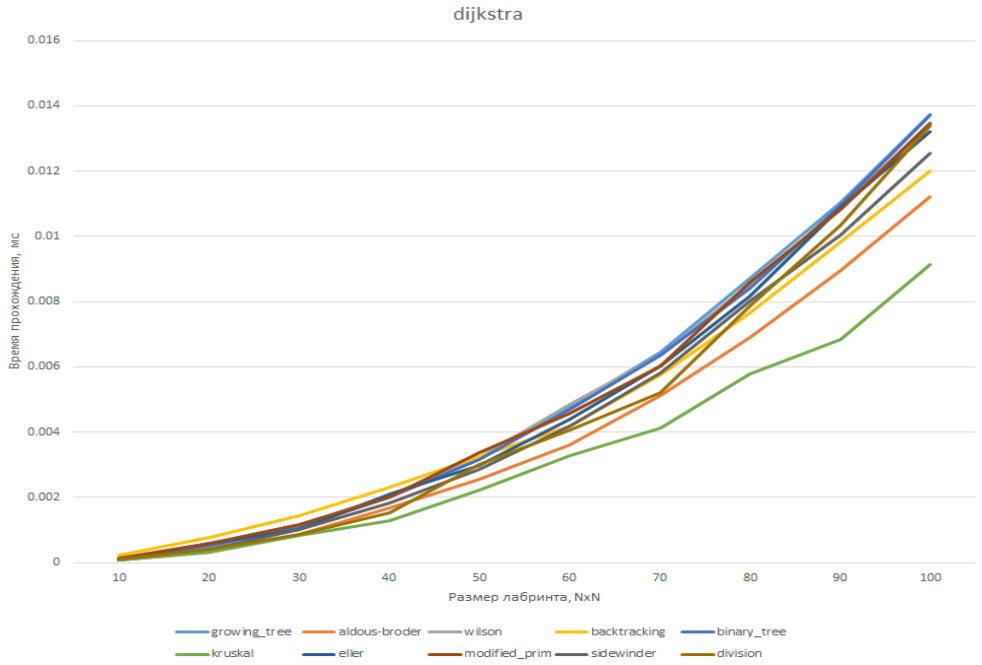


Рис. 2. Зависимость времени прохождения от размера лабиринта для алгоритма Дейкстры

Из рис. 3 видно, что самым сложным лабиринтом для прохождения алгоритмом поиска по ширине [1], является лабиринт, сгенерированный алгоритмом Уилсона (*wilson*) [1].

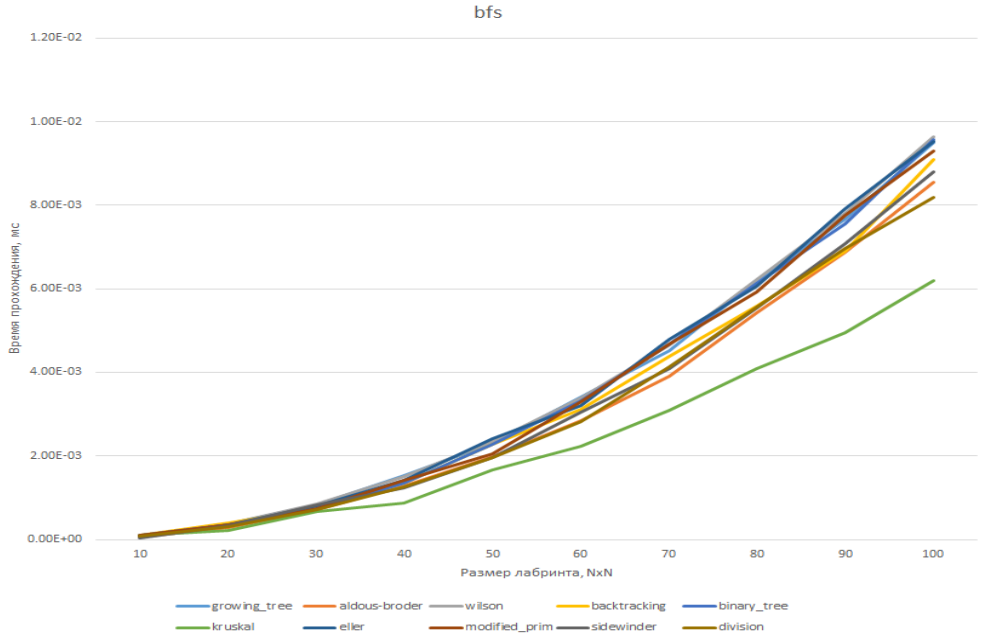


Рис. 3. Зависимость времени прохождения от размера лабиринта для алгоритма поиска по ширине

По общим данным, самым быстрым алгоритмом прохождения оказался поиск по ширине. Самым медленным – A-Star.

В процессе нашей исследовательской работы мы написали искусственную нейронную сеть [2]. Данная искусственная нейронная сеть состоит из 3 полносвязных слоев (каждый нейрон получает входные данные от всех нейронов предыдущего слоя и передает свой результат всем нейронам следующего слоя). Рассмотрим устройство нейронной сети для лабиринта 7 на 7 клеток (рис. 4).

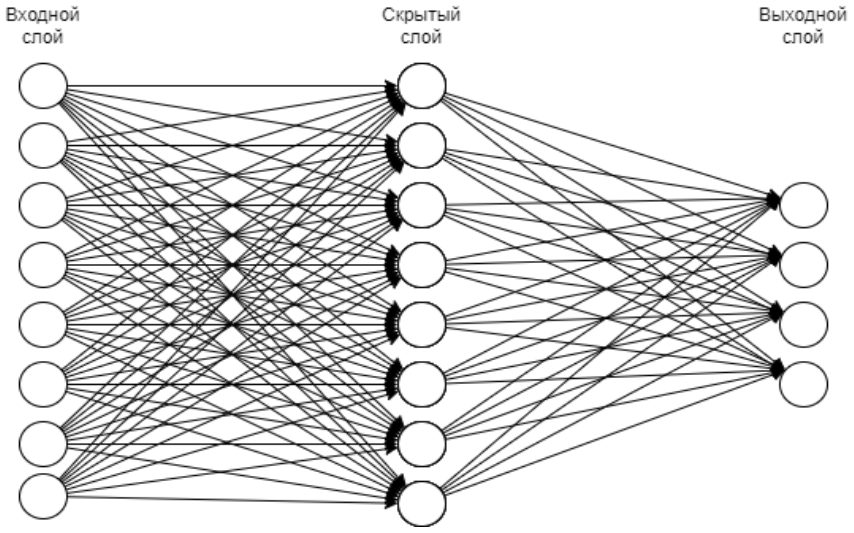


Рис. 4. Устройство искусственной нейронной сети

В первом слое (входном) находятся все ячейки нашего лабиринта. В каждом нейроне содержится значение от 0 до 3: 0 – стены нет, проход; 1 – стена; 2 – начало маршрута; 3 – конец маршрута.

Во втором слое (скрытом) каждый нейрон является переменной, значение которой меняется нейросетью таким образом, чтобы, «пройдя» по этим значениям, можно было дойти от начала до конца маршрута.

Третий слой является выходным. В нем находится 4 нейрона, которые делают вывод: куда идти (вверх, вниз, направо, налево) из определенной клетки.

Процесс обучения искусственной нейронной сети заключается в том, что нейронная сеть в своем скрытом слое должна обустроить карту для прохождения лабиринта. Иначе говоря, если искусственная нейронная сеть находиться в определенной клетке, то она должна пойти в определенном правильном направлении (рис. 5).

На графике (рис. 6) мы видим, что уже обученная нейронная сеть проходит лабиринт в 500 раз дольше нежели любой из приведенных классических алгоритмов. Расчеты производились на ноутбуке MacBook Air на чипе Apple M1 2020 года выпуска.

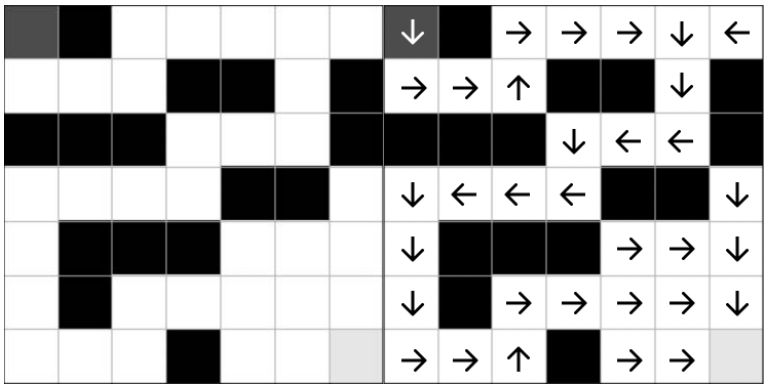


Рис. 5. Карта для прохождения лабиринта с предпочтительными направлениями из каждой клетки

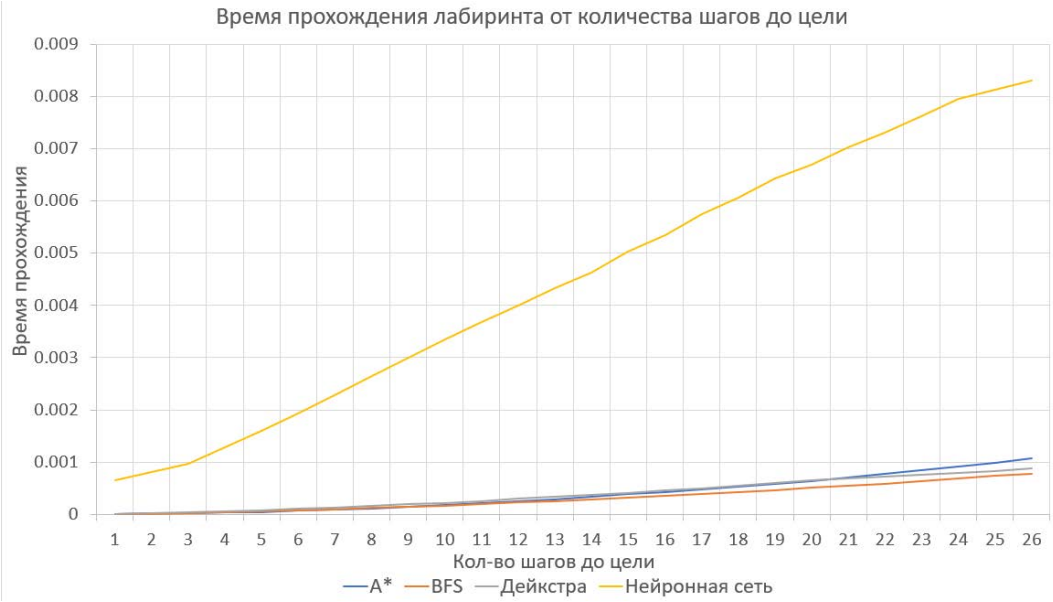


Рис. 6. Зависимость времени прохождения лабиринта от количества шагов до цели

Проведя исследование, мы можем сделать вывод, что использование ИИ для нахождения минимального пути в лабиринте является не рациональным. Помимо того, что ей нужно обучаться продолжительное время, иногда доходящее до нескольких часов (зависит от производительных мощностей), так ещё и после обучения она проходит лабиринт в сотни раз медленные всех алгоритмов, что видно на приведенном графике. К тому же большой проблемой является то, что для каждого лабиринта нейронную сеть нужно обучать заново. Причина заключается в том, что нейросеть не обладает такой архитектурой чтобы эффективно анализировать пространственную метрику.

Список используемых источников

1. Генерация кратчайшего пути в 2D лабиринтах [Электронный ресурс]. URL: https://studfile.net/preview/16380301/

2. Deep Reinforcement Learning for Maze Solving [Электронный ресурс]. URL: https://www.samyzaf.com/ML/rl/qmaze.html#The-Qmaze-Class

Статья представлена научным руководителем, профессором кафедры ПИВТ СПбГУТ, кандидатом технических наук, доцентом Пачиным А. В.

УДК 514.11

А. А. Боклаженко, Е. А. Григорьев, В. В. Ильин, Е. А. Шаров
(студенты гр. ИКПИ-13, СПбГУТ)

ИССЛЕДОВАНИЕ МЕТОДОВ ПРОЦЕДУРНОГО РАЗБИЕНИЯ ПЛОСКОСТИ

В компьютерном моделировании, симуляции местности, тестировании алгоритмов и дизайне текстур широко используются методы процедурной генерации, позволяющие получить псевдослучайные решения поставленных задач. В том числе актуальна проблема разбиения плоскости на псевдослучайные регионы. В статье сформулированы принципы качественного разбиения плоскости, рассмотрены существующие методы решения данной проблемы, а также предложены новые. Исследованы преимущества и недостатки каждого из методов на основании соответствия сформулированным в статье принципам.

плоскость, разбиение, регион, алгоритм, процедурная генерация, шум.

Требуется найти алгоритм разбиения плоскости на регионы на произвольном масштабе. При этом разбиение должно быть:

а) произвольным, без смещений, симметрий и прямых границ в любом из направлений;

б) однозначным и воспроизводимым, то есть не зависеть от порядка генерации;

в) регионы должны быть крупными, с минимальным числом анклавов, в идеале, эти параметры можно контролировать.

Алгоритм также должен быть применим в прикладных задачах, то есть его алгоритмическая сложность должна быть как можно меньше. Также для дискретизации разбиения требуется предварительно разбить плоскость клетчатой сеткой. Регионом будет считаться множество граничащих клеток, полученное в результате разбиения плоскости. Выполнить разбиение – значит предоставить алгоритм, который, принимая на вход клетку (x, y) из некоторого региона, возвращает множество клеток данного региона.

Для однозначной и воспроизводимой процедурной генерации каждой клетке на плоскости должно соответствовать определённое псевдослучайное число. Карта имеет сид, каждая клетка имеет координаты. Используется

хэш-функция, в которую для некоторой клетки (x, y) передаётся в качестве аргумента целая часть от суммы сида и произведений координат клетки на иррациональные коэффициенты. Данный способ позволяет при использовании надёжной хэш-функции и удачном подборе коэффициентов получать значения, не отличимые от белого шума, при том, благодаря иррациональности коэффициентов, исключаются паттерны.

Для генерации текстур и случайных карт на сегодняшний день используются следующие методы: Диаграмма Вороного и шум.

Диаграмма Вороного разбивает плоскость на полигоны, каждый из которых представляет собой множество всех точек плоскости, находящихся ближе к одному из заданных заранее сайтов (вершине), чем к остальным.

Поскольку в диаграмме Вороного по заданному множеству сайтов строятся ребра полигонов, то ее невозможно использовать для решения поставленной задачи напрямую.

Каждая из клеток плоскости обладает псевдослучайным значением. Те из них, чьё значение меньше заданного параметра, становятся сайтами. Используя идею объединения множества клеток по близости к сайтам, можно получить решение задачи следующего вида (рис. 1):

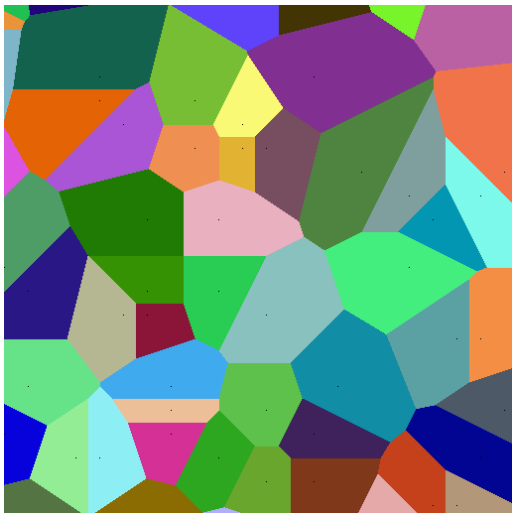


Рис. 1. Разбиение плоскости по Диаграмме Вороного

К достоинствам данного метода можно отнести отсутствие анклавов, к недостаткам – большую временную сложность алгоритма разбиения и простоту получаемых форм: каждый регион представляет собой выпуклый многоугольник.

Шум – математический алгоритм для получения карты высот. Его используют для генерации рельефа и текстур (наиболее используемыми в процедурной генерации являются шум Перлина и симплекс). Этот алгоритм также можно использовать для процедурного разбиения плоскости на регионы по принадлежности к слою (диапазону высот): две соседние клетки принадлежат одному региону если они принадлежат одному слою. Для более разнообразного разбиения можно использовать несколько шумов и рассматривать принадлежность по сочетанию слоёв. Ниже представлено разбиение по 1 и 2 (рис. 2) шумам соответственно.

Достоинства данного метода: ассиметричное разбиение на невыпуклые регионы, относительно небольшое число анклавов (при генерации нескольких шумов).

Недостатки: ресурсозатратность, сглаженность (простота) форм.

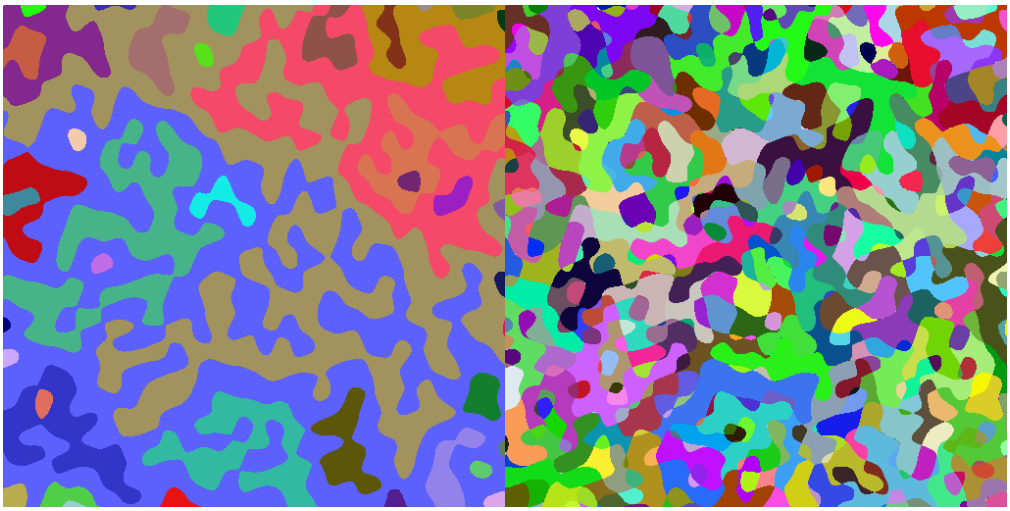


Рис. 2. Разбиение плоскости по одному (слева) и двум (справа) шумам

Рассмотренные методы являются популярными, но сложновычислимыми. Далее в статье предложены новые методы.

Разбиение на регионы по указателю. Каждая из клеток плоскости получает в соответствие целочисленное значение, которое интерпретируется, как указатель вверх, вправо, вниз или влево. Две соседние клетки принадлежат одному региону, если одна из них указывает на вторую (рис. 3).

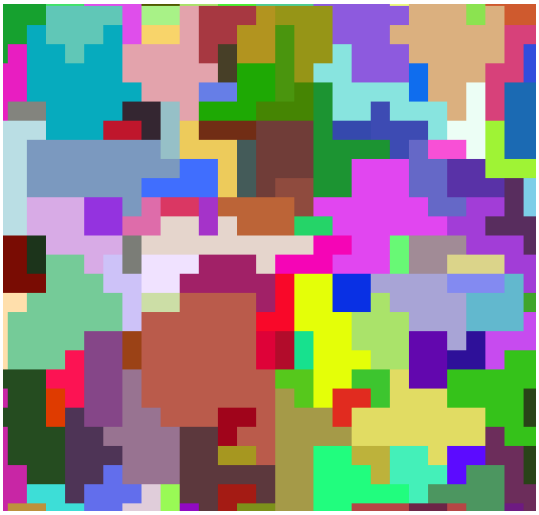


Рис. 3. Разбиение плоскости по указателю

Данный метод даёт разбиение на регионы произвольной формы, минимальный размер 2, средний порядка 7.5. Столь малые значения обусловлены низкой вероятностью, что две произвольные соседние клетки принадлежат одному региону: 7/16.

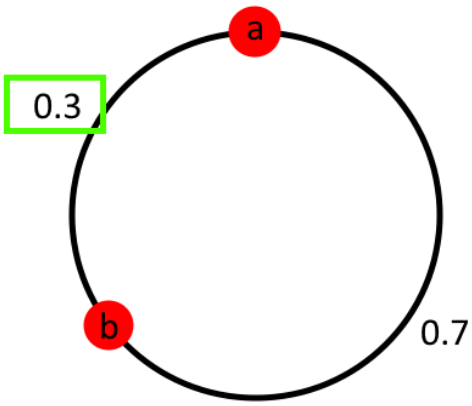


Рис. 4. Визуализация окружности численной разности

Разбиение на регионы с помощью численной разницы. Пусть числа с плавающей запятой расположены в диапазоне [0;1] на окружности таким образом, что 0 совпадает с 1, *a*, *b* являются числами, расположенными на этой окружности (рис. 4). Разницей чисел *a* и *b* будет называться такое число c, которое характеризует длину минимальной дуги от *a* до *b*.

Пусть клеткам на плоскости присвоены значения с плавающей запятой в диапазоне [0;1]. Разница соседних клеток

не должна превосходить некоторой величины *t*, следовательно, вероятность их принадлежности будет равна удвоенному значению этой величины. Возможность определять вероятность даёт преимущество перед разбиением по указателю, позволяя получать регионы больших размеров (рис. 5).

Достоинства данного способа: случайность форм и размеров регионов, быстрота вычисления. Очевидный недостаток – большое число анклавов и маленьких регионов. Для сглаживания карты введено правило: если площадь региона меньше заданной величины Smin, то данный регион присоединяется к тому, к которому самая нижняя из самых правых клеток данного региона примыкает справа. Таким образом карту можно существенно сгладить, но это ресурсозатратно (рис. 6).

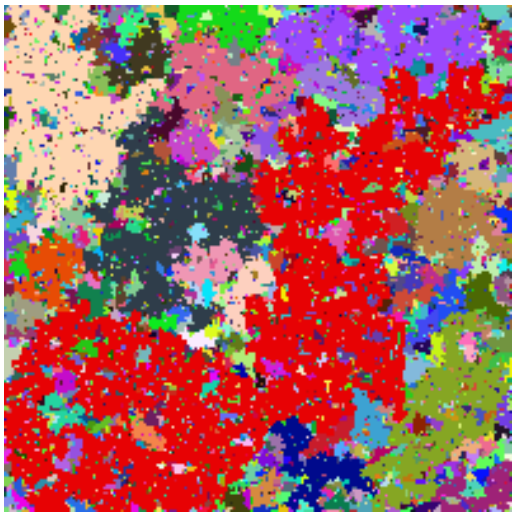


Рис. 5. Разбиение плоскости с помощью численной разницы

Проведённые исследования показали, что наиболее перспективными с прикладной точки зрения методами разбиения плоскости являются шум и разбиение по численной разнице. Наименьшее число анклавов гарантирует шум, наилучшее разнообразие размеров и форм регионов характерно для разбиения по численной разнице.

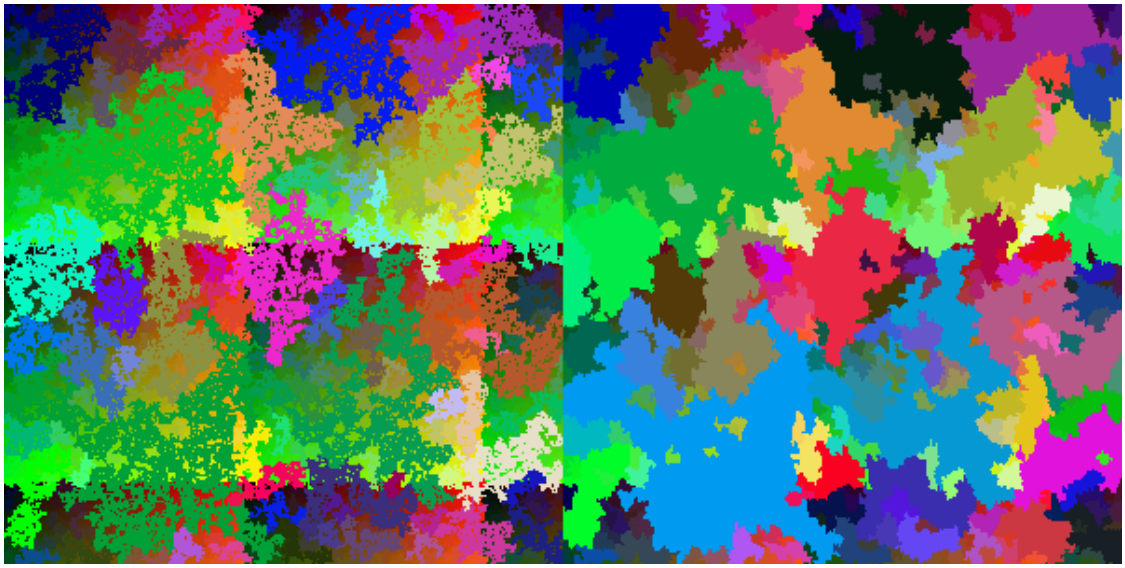


Рис. 6. Разбиение плоскости с помощью численной разницы, сид 2 (слева); сглаживание разбиения (справа)

Список используемых источников

1. Ауренхаммер Ф., Дер-Цай Ли, Клейн Р. Диаграммы Вороного и триангуляции Делоне. World Scientific. ISBN 978-9814447638.s.

2. Шум Перлина [Электронный ресурс]. URL: https://eev.ee/blog/2016/05/29/perlin-noise/ (дата обращения 15.05.2023).

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 658.7

И. Г. Верещагина (студентка гр. БИМ-11з, СПбГУТ)

ИНФОРМАЦИОННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ЛОГИСТИКОЙ НА ПРЕДПРИЯТИИ

В современных условиях организация эффективных логистических процессов на предприятии невозможно без активного применения интегрированных информационных систем. Информационные системы управления логистикой являются связующим звеном между производством, снабжением и сбытом предприятия, и управляют данными, сопровождающими перемещение материальных потоков.

В статье рассмотрено понятие информационных логистических систем, исследованы принципы их работы, достоинства и недостатки. Проведено исследование рынка и сравнительный анализ информационных логистических систем.

предприятие, логистика, информационная система, информационная логистическая система.

В последние годы всё больше предприятий уделяют внимание логистике. Это связано с тем, что на транспортировку, хранение и упаковку товара затрачивается до 35 % от его стоимости. Соответственно, эти издержки влияют на цену товара, которая является одним из важнейших факторов конкурентной борьбы за потребителя. Поэтому для сокращения логистических издержек предприятия уделяют внимание модернизации своей транспортной логистики [1].

Информационная логистическая система – система, работающая с информацией в логистике. Как и любая другая система, информационная система (ИС) должна складываться из последовательно взаимосвязанных элементов и иметь некоторую совокупность интеграционных качеств. Декомпозицию ИС на составляющие элементы можно осуществлять по-разному. Чаще всего ИС делят на две подсистемы:

– функциональную;
– административно-управленческую.

Рассмотрим основные функциональные ИС, применяемых в логистике. Характеристики информационных системы представлены в табл. 1 [2].

Функциональные ИС предназначены больше для создания отчетов, формирования выводов по данным из документов. Так как для логистики необходимы возможности внесения изменений в систему и заведения необходимой информации, подробнее будем рассматривать административно-управленческие информационные системы (табл. 2).

ТАБЛИЦА 1. Основные характеристики функциональных информационных систем, применяемых в логистике

Типы системы	Информационные вводы	Обработка	Информационные выводы
Executive Support Systems (ESS) – Исполнительные системы поддержки выполнения (на стратегическом уровне)	Совокупные данные: внешние и внутренние	Графика, моделирование, интерактивность	Проекции, реакции на запросы
Management Information Systems (MIS) – Управляющие информационные системы	Итоговые операционные данные, данные большого объема, простые модели	Обычные доклады, простые модели, простейший анализ	Резюме и возражения
Decision Support Systems (DSS) – Системы поддержки принятия решений (на управленческом уровне)	Слабоформализованные данные, аналитические модели	Моделирование, анализ, интерактивность	Специальные доклады, анализ решений, реакция на запросы
Knowledge Work Systems (KWS) – Системы управления знаниями	Технические данные проекта, база знаний	Моделирование, проигрывание	Модели, графика
Office automation Systems (OAS) – Системы автоматизации делопроизводства	Документы, расписания	Документы управления, планирование, связь	Документы, графики, почта
Transaction Processing Systems (TPS) – Системы диалоговой обработки запросов (на эксплуатационном уровне)	Транзакции, результаты	Сортировка, список, слияние, модифицирование	Детальные доклады, списки, резюме

Во всех отраслях логистики в последнее время интенсивно внедряются ИС и информационные технологии, что положительно сказывается на оптимизации логистики. Такое внедрение приводит к тому, что логистика становится все более и более оптимизированной.

Согласно исследованию компании Gartner, объем мирового рынка программных систем, предназначенных для управления предприятием, по итогам 2020 года достиг $40 млрд, увеличившись на 4 % в сравнении с показателем годичной давности.

В 2021 году рынок ИС в России перешел к достаточно уверенному росту после шока, вызванного «антиковидными» мерами предыдущего 2020 года. И хотя многие ограничения продолжали сохраняться, ряд проектов, поставленных на паузу в острый период пандемии, были разморожены. COVID-19 для многих компаний стал настоящим вызовом.

ТАБЛИЦА 2. Административно-управленческие информационные логистические системы

Тип системы	Описание
WMS (Warehouse Management System) – Система управления складом	Позволяет управлять складской логистикой в рамках различных технологических процессов (прием и отгрузка товара, внутренние перемещения) в реальном времени
SCM (Supply Chain Management) – Система управления цепочками поставок	Предназначено для автоматизации и управления всеми этапами снабжения предприятия и для контроля всего товародвижения на предприятии
TMS (Transportation Management System) – Система управления перевозками	Позволяет избежать ошибок в планировании и выполнении операций под влиянием человеческого фактора
SFMS (Sales Force Management System) – Система управления продажами	Призвана автоматизировать процесс продаж в различных сферах бизнеса и сделать его наиболее эффективным и прибыльным
PMS (Procurement Management System) – Система управления материальными потоками	Помогает эффективно управлять своими поставками и быстро адаптироваться к изменяющимся экономическим условиям, любой нестабильности в цепочках поставок или обстоятельствам, связанным с риском
MES (Manufacturing Execution System) – Система управления производственными процессами	Предназначена для решения задач оперативного планирования и управления производством, призваны решать задачи синхронизации, координировать, анализировать и оптимизировать выпуск продукции в рамках определенного производства
ERP (Enterprise Resource Planning System) – Система планирования ресурсов предприятия	Система планирования потребности предприятия в трудовых, материальных, информационных ресурсах
MRP (Material Requirement Planning) – Система планирования потребностей в материалах	Гарантирует обеспечение производства материалами и комплектующими изделиями в планируемый период времени

Оправившись от COVID-19, одни руководители решили укрепить лидирующие позиции в отрасли, для кого-то пандемия стал возможностью занять освободившуюся долю рынка, а кто-то увидел в сложившейся ситуации новые перспективы [3].

Исследование рынка за 2023 год показало, что наиболее востребованными являются следующие типы (классы) ИС: ERP, TMS, WMS, YMS, CRM.

В связи с многообразием ИС возникает проблема выбора. Из рис. 1 видно какие типы систем предприятия предпочитают использовать в своей деятельности.

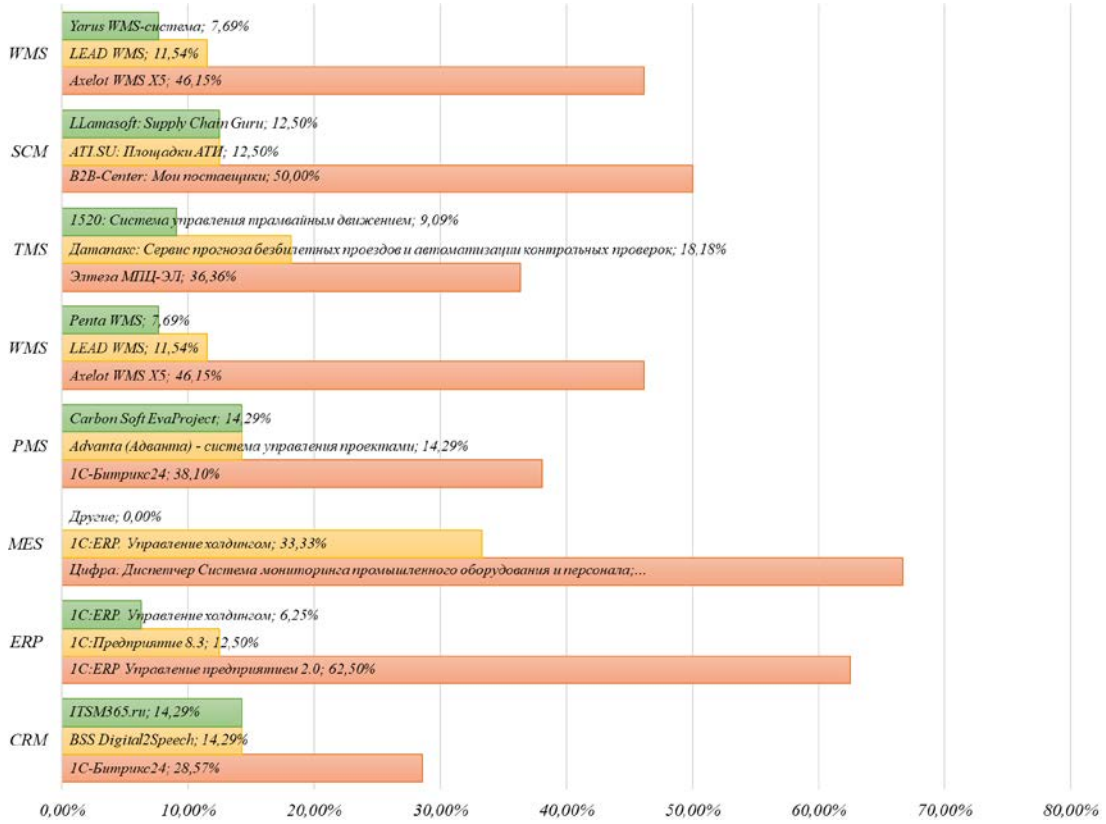


Рис. 1. Востребованность информационных систем в логистике

Поскольку указанные ИС сильно отличаются друг от друга, то проведем сравнительный анализ только самых востребованных, результаты представлены в табл. 3 (см. ниже).

Результаты проведенного анализа позволят предприятию выбрать оптимальную систему по критерию цена/качество, что в свою очередь позволить повысить как эффективность логистических бизнес-процессов, так и эффективность деятельности предприятия в целом.

Список используемых источников

1. Тенденции рынка информационных систем управления транспортной логистикой [Электронный ресурс]. URL: https://corpinfosys.ru/archive/issue-6/59-2019-6-transporttrends (дата обращения 23.05.2023).

2. Неизвестный С. И. Информационные технологии управления логистическими системами : учебное пособие. М. : КНОРУС, 2022. 354 с.

3. Системы управления предприятием (ERP) рынок России [Электронный ресурс]. URL: https://www.tadviser.ru/index.php/Статья:Системы_управления_предприятием_(ERP_рынок_России) (дата обращения 23.05.2023).

ТАБЛИЦА 3. Сравнение ИС для управления логистикой

Параметры сравнения	Информационные системы					
	Axelot WMS X5	B2B-Center: Мои поставщики	Элтеза МПЦ-ЭЛ	1С-Битрикс24	Цифра: Диспетчер Система мониторинга промышленного оборудования и персонала	1С: ERP Управление предприятием 2.0
Online формат	+	+	+	+	+	–
Совместный доступ для нескольких пользователей	+	–	+	+	+	–
Синхронизация с другими ПО	+	–	–	+	+	+
Поддержка ОС	Windows, MacOS, Android	Windows, MacOS, Android	Windows	Windows, MacOS, Android	Windows, MacOS, Android	Windows, MacOS
Обмен файлами	+	+	+	+	+	+
Наличие чата для сотрудников	–	–	–	+	–	–
Визуализация данных	+	–	+	+	+	+
Русскоязычная версия	+	+	+	+	+	+
Создание отчетов	+	–	–	–	+	+
Учет затрат	–	–	–	–	–	+
Прогнозы	–	–	+	–	+	+
Облачное хранение	+	+	+	+	+	+
Наличие пробного периода	–	+	–	+	+	–
Управление складом	+	–	–	–	+	+
Автоматизация основных процессов	+	–	+	+	+	+
Полный выкуп ПО	–	–	+	–	+	+
Покупка лицензии	+	+	–	+	+	+
Управление транспортом	–	–	+	–	+	–

Параметры сравнения	Информационные системы					
	Axelot WMS X5	B2B-Center: Мои поставщики	Элтеза МПЦ-ЭЛ	1С-Битрикс24	Цифра: Диспетчер Система мониторинга промышленного оборудования и персонала	1С: ERP Управление предприятием 2.0
Управление продажами	–	+	–	+	–	+
Управление проектами	–	–	–	+	+	+
Стоимость (руб.)	99 300 ₽/12 мес	19 500 ₽/квартал		159 000 ₽/12 мес	Лицензия от 3 – 5 000 ₽/мес на ед. оборудования	432 000 ₽ выкуп ПО, Аренда от 82 ₽/день, Использовать в «облаке» от 1 288 ₽/мес. за сеанс.

Статья представлена научным руководителем, заведующим кафедрой БИ СПбГУТ, кандидатом экономических наук, доцентом Вольфсоном М. Б.

УДК 004.422.8

И. А. Внуков, Р. А. Малдыбаев (студенты гр. ИСТ-014, СПбГУТ)
А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)
П. В. Матвеев (студент гр. ИСТ-013, СПбГУТ)

РЕАЛИЗАЦИЯ ОСНОВНЫХ КОМПОНЕНТОВ СЕРВИСА ДЛЯ СОПРОВОЖДЕНИЯ ОБРАЗОВАТЕЛЬНОГО ТУРИЗМА

Актуализирована разработка сервиса для сопровождения образовательного туризма. Предложена реализация сервиса по предоставлению образовательного туризма в Санкт-Петербурге. В предложенной реализации рассмотрен функционал формирования персонального образовательного тура из предоставленных сервисом мероприятий для абитуриентов, а также функционал размещения информации о турах и программах, управления бронированиями и расписанием для учебных заведений. Описана логика программно-серверной части, а также функционал взаимодействия с пользовательским интерфейсом. Описаны перспективы развития предложенной реализации.

веб-сервис, веб-разработка, образовательный туризм, образование, организация, образовательная траектория.

В последние годы все больше будущих студентов приезжают в другие города, чтобы посетить открытые мероприятия университетов – дни открытых дверей, подготовительные курсы, олимпиады и другие. Для подобных целей существуют образовательные туры, однако они составлены по заранее сформированным маршрутам и направлены на определенный круг лиц. Кроме того, в широком использовании нет информационных систем, которые позволили бы составлять собственные образовательные маршруты. В связи с этим актуализируется вопрос разработки сервиса по сопровождению образовательного туризма.

Санкт-Петербург известен своими высшими учебными заведениями, многие из которых привлекают абитуриентов со всей России и из-за рубежа. К тому же Санкт-Петербург – источник культурного и исторического наследия, который привлекает тысячи туристов каждый год. Предлагаемый сервис по сопровождению образовательного туризма позволяет выбрать туры, включающие не только мероприятия учебных заведений, но и культурно-развлекательные мероприятия города. Развитие подобной системы позволит абитуриентам получить более глубокие знания о городе и его учебных заведениях, что поможет подготовить будущих студентов к успешной учебе в одном из самых красивых городов России, а также создаст новые возможности для туристической индустрии.

В данной статье рассматривается реализация основных компонентов сервиса для сопровождения образовательного туризма, позволяющего учебным заведениям представлять собственные образовательные события или туры, а пользователям – возможность составления расписания из мероприятий учебных заведений, а также возможность добавления дополнительных культурно-досуговых мероприятий города. Платформа будет полезна не только абитуриентам бакалавриата, магистратуры и аспирантуры, но и школьникам, а также всем желающим получить дополнительное профессиональное образование.

Одним из главных преимуществ данной платформы является возможность создания персонализированных туров. Пользователи сервиса для сопровождения образовательного туризма смогут выбрать интересующие их вузы и мероприятия, добавлять культурно-развлекательные события и составлять собственный маршрут. Такой подход позволит каждому посетителю оптимизировать личные ресурсы и не только ознакомиться с образовательными учреждениями, но и оценить достопримечательности города в соответствии с личными предпочтениями и интересами.

Таким образом, данный сервис способствует развитию образовательной и туристической сфер. Он поможет учебным заведениям расширить контингент иногородних и иностранных студентов, повысить рейтинг

и узнаваемость собственных образовательных программ, а пользователям – эффективнее использовать собственное время, получать полную и актуальную информацию о программах обучения и мероприятиях университетов, минимизировать стресс, вызванный сложностью планирования и бронирования билетов.

Предложенная реализация информационной системы разработана с использованием архитектурного подхода REST API, её общая структура представлена на рис. 1. Данная реализация состоит из следующих взаимосвязанных элементов:

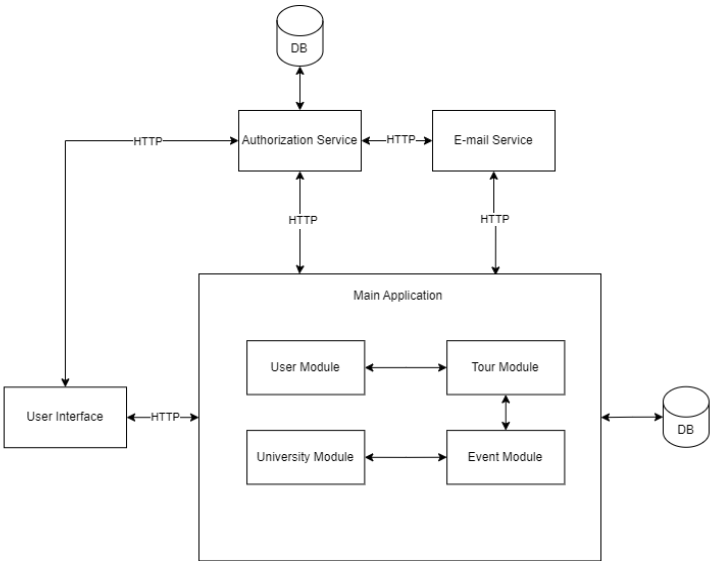


Рис. 1. Общая архитектура предложенной реализации системы

– основное приложение (*Main Application*) – приложение, созданное с помощью веб-фреймворка FastAPI, в которое входят модули, отвечающие за соответствующие сущности: модуль пользователя (*User Module*), отвечающий за управление профилем пользователя, модуль учебного заведения (*University Module*), в котором реализован функционал размещения информации о турах и программах, управления бронированиями и расписанием, модуль мероприятия (*Event Module*), отвечающий за добавление и редактирование информации о мероприятиях, а также связывание их с учебными заведениями и категориями мероприятий, и модуль тура (*Tour Module*), реализующий функционал формирования персональных образовательных туров пользователей;

– сервис авторизации (*Authorization service*) – приложение, созданное с помощью фреймворка Gin Web Framework, отвечающее за регистрацию и авторизацию пользователя в системе и использующее JWT (*JSON Web Token*) для аутентификации и обеспечения безопасного доступа к ресурсам;

– сервис e-mail рассылки (*E-mail Service*) – приложение, созданное с помощью фреймворка Gin Web Framework, отвечающее за отправку электронных писем;

– базы данных (*DB*), основанные на свободной объектно-реляционной системе управления базами данных PostgreSQL;

– пользовательский интерфейс (*User Interface*), разработанный с использованием JavaScript-библиотеки React и фреймворка Tailwind CSS.

База данных основного приложения представлена на рис. 2 и включает в себя следующие ключевые сущности: пользователь системы (*User*); уровень образования пользователя (*Edu_Status*); учебное заведение (*University*); мероприятие (*Event*); категория мероприятия (*Category*); метка для поиска мероприятий (*Tag*); тур, предлагаемый учебным заведением, который состоит из мероприятий (*Tour*).

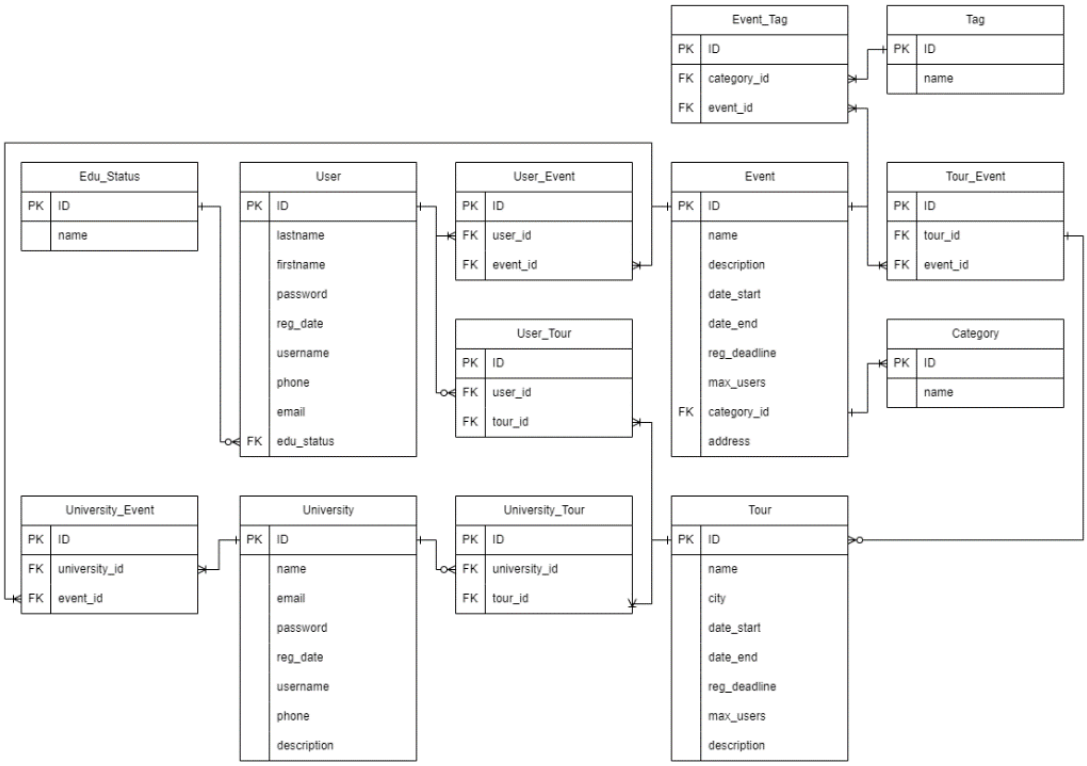


Рис. 2. Структура базы данных основного приложения

Сервис авторизации содержит базу данных, которая включает в себя следующие ключевые сущности: пользователь системы (*User*); права доступа пользователя (*Role*); токен обновления токена доступа JWT (*Refresh_Token*).

Для обновления JWT сервис авторизации использует дополнительные Refresh-токены, которые выдаются пользователю при успешной аутентификации и позволяют обновлять JWT без необходимости повторной аутентификации и ввода своих учетных данных при обновлении токенов.

Сервис авторизации может быть интегрирован с различными приложениями, которые требуют аутентификации пользователей.

Функционал формирования персонального образовательного тура для абитуриентов включает следующие возможности: авторизация и выбор

уровня образования; выбор учебного заведения для просмотра туров и программ; просмотр детальной информации о турах и программах; выбор мероприятий и формирование персонального тура; подтверждение бронирования и оплата [1].

Функционал размещения информации о турах и программах для учебных заведений включает следующие возможности: добавление администратором информации о турах и программах; управление бронированием мест на мероприятиях; редактирование расписания мероприятий и обновление информации; уведомление пользователей о новых мероприятиях.

Пользовательский интерфейс страницы с полным списком проводимых мероприятий данной предложенной реализации представлен на рис. 3.

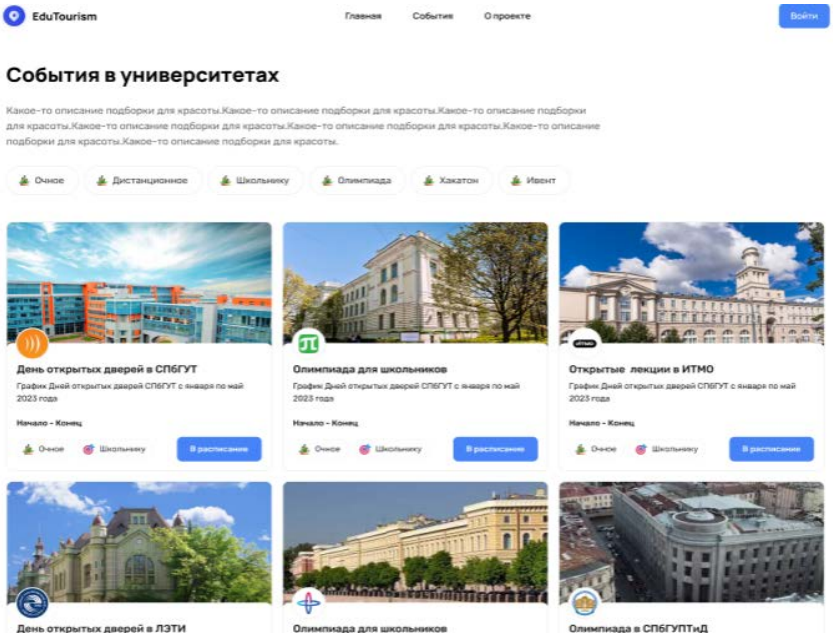


Рис. 3. Пользовательский интерфейс с полным списком проводимых мероприятий

Функция добавления нового тура представлена на рис. 4, функция приема данных одного из событий представлена на рис. 5.

```
@router.post("/users/{user_id}/tours")
def add_tour_to_user(user_id: int, tour_id: int, db: Session = Depends(get_db)):
    user = db.query(User).filter(User.id == user_id).first()
    if not user:
        raise HTTPException(
            status_code=404, detail={"status": "error", "message": "User not found"}
        )

    tour = db.query(Tour).filter(Tour.id == tour_id).first()
    if not tour:
        raise HTTPException(
            status_code=404, detail={"status": "error", "message": "Tour not found"}
        )

    user_tour = User_Tour(user_id=user_id, tour_id=tour.id)
    db.add(user_tour)
    db.commit()

    return {"status": "success", "message": f"Tour {tour_id} added to user {user_id}"}
```

Рис. 4. Фрагмент кода добавления нового тура пользователю

```
import axios from "axios";
import { useEffect, useState } from "react"
import { base_url } from "../axios_config";

export type event = {
    id: number,
    name: string,
    description: string,
    max_users: number,
    date_start: Date,
    date_end: Date,
    reg_deadline: Date,
    city: string,
    category: number[]

};

export const getEvents = () => {
    const [event, eventChange] = useState<Array<event>>();
    useEffect(()=>{
        axios.get(base_url + 'events').then(
            responce => eventChange(JSON.parse(responce.data))
        )
    }, []);
    return event;
}
```

Рис. 5. Фрагмент кода приема данных по событию

Создание сервиса для сопровождения образовательного туризма на основе современных подходов анализа и моделирования информационных систем является важным шагом в развитии туризма и образования. Сервис будет способствовать повышению качества образовательных маршрутов и удобства их использования для туристов, а также внедрению новых форм образовательного процесса. Кроме того, данное решение позволит повысить рейтинг образовательных программ и учебных заведений и укрепить конкурентные позиции на рынке туризма, расширить туристический потенциал города. Таким образом, разработка такого сервиса имеет высокий потенциал для развития туризма и образования и является перспективным направлением деятельности в ближайшем будущем.

Список используемых источников

1. Жаранова А. О., Капитоненко В. В., Пелих Д. А. Формирование концептуального подхода к разработке систем в области образовательного туризма // Актуальные проблемы инфотелекоммуникаций в науке и образовании. XII Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб. : СПбГУТ, 2023. Т. 4. С. 424–429.

Статья представлена доктором технических наук, профессором Птицыной Л. К.

УДК 004.65

Д. И. Восьмаков (студент гр. ИКПИ-14, СПбГУТ)
В. В. Уласик (студентка гр. ИСТ-261, СПбГУТ)

ПРОТОТИП МИКРОПРИЛОЖЕНИЯ ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ О НАУЧНЫХ ПУБЛИКАЦИЯХ

Предложена концепция микроприложения, автоматизирующего процесс концентрации информации о публикациях отдельных авторов, авторских коллективов, структурных подразделений и организаций. Рассмотрены принципы функционирования программного решения, предложен вариант реализации системы на основе распределённой архитектуры.

микроприложение, распределенная система, интероперабельность, наукометрия.

В настоящее время остро стоит проблема автоматизации работы с документами, создания отчётов, хранения больших массивов данных. Так, в образовательных и научных учреждениях Российской Федерации работа по формированию отчётной документации о результатах научной деятельности отдельных авторов, структурных подразделений и организаций выполняется вручную с минимальной степенью автоматизации. Данный процесс является рутинным и отнимает существенную часть рабочего времени у научно-педагогических работников, а также сотрудников, отвечающих за формирование отчетов.

Эта проблема может быть решена путём создания интероперабельных информационных систем, допускающих глубокую интеграцию в единую киберфизическую среду постиндустриального общества. Как и любая составляющая киберфизической среды, микроприложение хранения и обработки информации должно строится на трех базовых принципах: агентности, информационного самообслуживания и управляемой информационной открытости [1]. Интероперабельность подразумевает сквозное информационное взаимодействие и функциональную совместимость систем вне зависимости от их внутреннего устройства.

Целью создания микроприложения является сокращение времени на формирование отчетов о публикационной активности, а также на включение информации в портфолио за счет реализации одного из базовых принципов информационных технологий – DRY (*don't repeat yourself* – «не повторяйся») [2]. При этом каждый участник (физическое или юридическое лицо), представленный в системе агентом, имеет возможность распоряжаться информацией в рамках указанной лицензии (управляемая информационная открытость).

Исходя из перечисленных выше базовых принципов построения системы разработан следующий вариант реализации программного решения. Система функционирует по принципам, схожими с принципами, на основе которых построены распределённые системы управления версиями [3]: у каждого из пользователей имеется своя реплика информации о публикациях; синхронизация между несколькими экземплярами программы выполняется путём импорта или экспорта информации о публикациях в другие системы. По умолчанию пользователь работает с локальным набором данных в своей системе. В этом заключается научная новизна решения.

Микроприложение должно обеспечить добавление, редактирование и удаление информации о научных публикациях, а также экспорт информацию в формате, удобном для обмена между несколькими развёрнутыми копиями системы. Микроприложение должно иметь функционал по фильтрации и отображению списка библиографических источников, вывод подробной информации о каждом источнике и копирование библиографического описания в рекомендованном формате для вставки в список цитирования.

Прототип микроприложения хранения и обработки информации о научных публикациях состоит из микросервиса ASP.NET Core и микрофронтенда, реализованного на Blazor. В роли объектно-реляционного преобразователя выступил Entity Framework Core. В рамках студенческой научно-исследовательской работы предполагается на базе данного прототипа создать интероперабельное микроприложение, которое может быть интегрировано в любые киберсреды, построенные на базе принципов агентности, информационного самообслуживания и управляемой информационной открытости.

В дальнейшем на основе данных принципов и на базе отработанной технологии планируется создать семейство программных продуктов для учета личных достижений и результатов научно-исследовательской деятельности, включая участие в конкурсах, прохождения курсов повышения квалификации и стажировок, свидетельств о государственной регистрации.

Список используемых источников

1. Верхова Г. В., Акимов С. В. Интеграция локальных интероперабельных киберсред виртуальных организаций в единую киберсреду постиндустриального общества // Волновая электроника и инфокоммуникационные системы : сборник статей XXIV Международной научной конференции. Санкт-Петербург, 2021. С. 34–39.

2. Haoyu W., Haili Z. Basic Design Principles in Software Engineering // 2012 Fourth International Conference on Computational and Information Sciences, Chongqing, China, 2012, pp. 1251–1254, doi: 10.1109/ICCIS.2012.91.

3. Москалева Ю. П., Сейдаметова З. С. Централизованные и распределенные системы управления версиями как учебная платформа // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. № 3 (17). С. 112–116.

4. Эванс Э. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем : пер. с англ. М. : Вильямс, 2011. 448 с.

Статья представлена научным руководителем, доцентом кафедры ИСАУ СПбГУТ, кандидатом технических наук, доцентом Акимовым С. В.

УДК 004.93'12

Э. Э. Гайбалиев (студент гр. ИСТ-013, СПбГУТ)
А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПОИСКА МУЗЫКИ ПО РИТМУ

Рассмотрены существующие сервисы по поиску музыкальных композиций, описаны их существенные недостатки. Актуализировано развитие информационных систем для поиска музыки. Представлена построенная диаграмма прецедентов для демонстрации функционала системы поиска музыки по ритму. Описан алгоритм работы поиска музыки по ритму. Определен спектр технологий для разработки системы поиска музыкальных композиций. Приведены перспективы развития информационной системы поиска музыки по ритму.

проектирование информационной системы, поиск музыки, ритм, распознавание музыки.

В настоящее время музыка стала неотъемлемой частью жизни каждого человека. Ее слушают везде: в машине, на работе, во время занятий спортом и даже перед сном. Однако музыкальные предпочтения у каждого человека индивидуальны, и для того, чтобы найти музыку, которая максимально соответствует нашим настроениям и предпочтениям, необходимо потратить много времени на поиск.

Около 225 миллионов человек по всему миру ежемесячно ищут музыку с помощью сервиса Shazam [1]. Однако у данного сервиса есть существенное ограничение: необходим хотя бы небольшой фрагмент искомой песни. Систем для поиска музыки лишь по ритму практически не существует. Из всех протестированных систем с подобным функционалом наиболее корректную работу показала лишь одна – Ritmoteka. Однако и у нее есть существенные недостатки: маленькая библиотека музыки и, соответственно, низкий процент успешного поиска. Таким образом, разработка системы для поиска музыки по ритму приобретает высокую актуальность ввиду востребованности и отсутствия конкуренции.

Основной функцией системы является поиск музыки на основе введенного ритма. Ритм – организация музыки во времени. Основу ритма составляют соотношения длительностей (звуков и пауз). Ритм может быть введен

тремя способами: с помощью клавиатуры, с помощью микрофона, а также голосом. Первый способ подразумевает собой настукивание пользователем ритма по клавише. Второй – ввод ритма через микрофон посредством использования подручных предметов, будь то стук по столу или использование музыкальных инструментов. Третий способ – ввод ритма с помощью голоса, который рассчитан для людей, обладающих определенными вокальными навыками.

После ввода ритма пользователь может ознакомиться с выдачей: просмотреть текст песни, воспроизвести композицию. Если в выдаче пользователь нашел искомую песню, он может отметить её, после чего песня добавится в его историю поиска. На случай, если песни не оказалось в выдаче, пользователю предложена кнопка «Песня не нашлась».

В истории поиска хранятся композиции, отмеченные пользователем при поиске. Пользователь по-прежнему может ознакомиться с текстом песни или воспроизвести ее. Если песня попала в историю по ошибке, она также может быть удалена.

Для более наглядной демонстрации функционала системы на рис. 1 представлена диаграмма прецедентов.

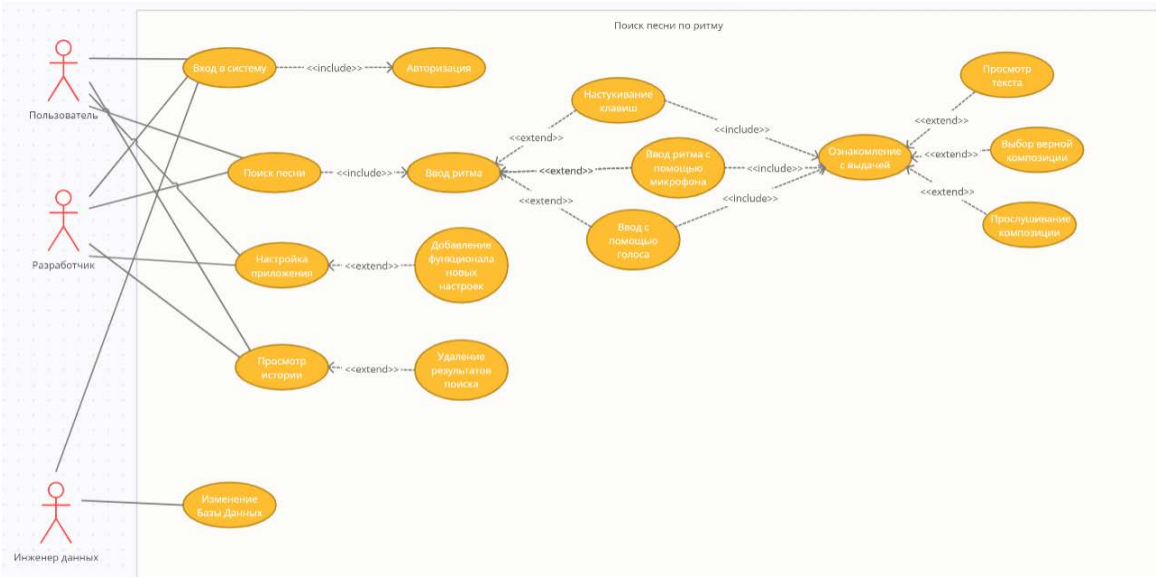


Рис. 1. Диаграмма прецедентов информационной системы поиска музыки по ритму

Состояния системы поиска в момент ввода ритма представлено на рис. 2.

Алгоритм поиска музыки работает следующим образом. Подразумевается, что процедура поиска начинается с инициации со стороны пользователя – нажатием кнопки поиска. После этого пользователь вводит ритм любым доступным способом. Ритм обрабатывается, система осуществляет запрос в базу данных, где хранятся все обработанные ритмы аудиозаписей,

производится поиск наиболее похожих композиций. Из базы данных система получает выдачу песен, которую передает на ознакомление пользователю. Если искомая песня оказалась в выдаче, пользователь отмечает ее, песня добавляется к нему в историю поиска, в базе данных создается соответствующая запись. Если песня не нашлась, пользователь может нажать на кнопку «Песня не нашлась», после чего в базе данных также создастся соответствующая запись, это поможет в будущем с обучением алгоритма поиска.

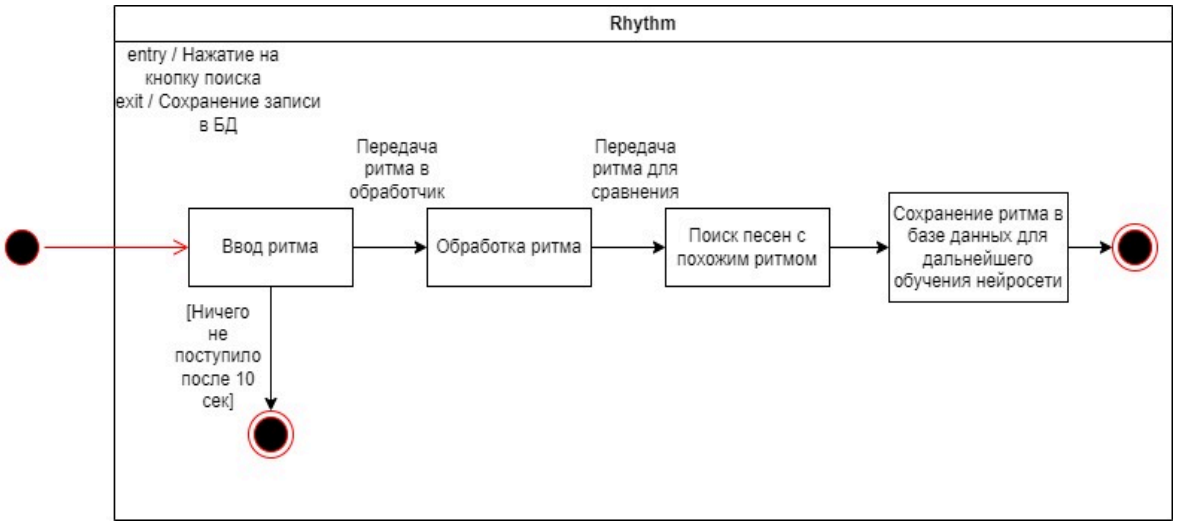


Рис. 2. Состояния системы поиска ритма

Диаграмма деятельности для прецедента «Поиск песни» представлена на рис. 3 (см. ниже).

Для разработки системы поиска музыки по ритму может использоваться следующий набор технологий: для пользовательского интерфейса HTML, CSS, JavaScript (*React.js* или *Vue.js*); Python Django для разработки веб-приложения; Python (*Kivy* или *Beeware*) для backend-части мобильного приложения.

Для реализации алгоритма поиска музыки планируется использовать библиотеку librosa языка программирования Python для анализа музыки и аудио [2]. С ее помощью обрабатывается ритм музыки, который хранится в базе данных в виде числовых массивов, где каждое число соответствует временному промежутку удара. Введенный пользователем ритм также обрабатывается в числовой массив. Поиск музыки в таком случае представляет собой сравнение этих массивов и выдачу наиболее похожих. Однако пользователи могут вводить ритм в отличном от оригинала темпе, особенно когда дело касается мало знакомой песни. Таким образом, должны сравниваться не сами значения в массивах, а относительное расстояние между ними.

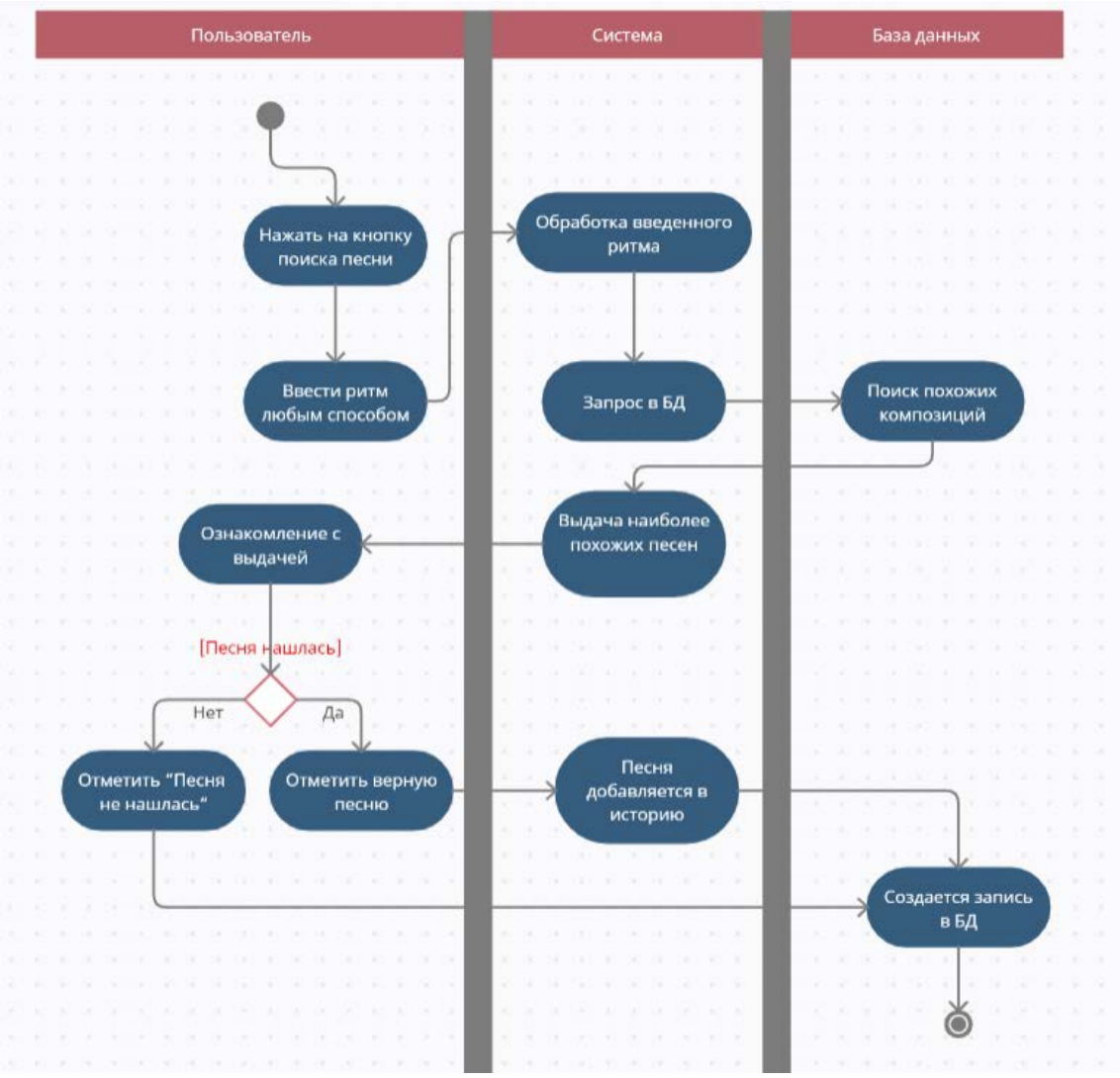


Рис. 3. Диаграмма деятельности для прецедента «Поиск песни»

У системы поиска музыкальных композиций по ритму есть большой потенциал в связи с высокой актуальностью и технологической новизной. В дальнейшем возможна реализация полноценного сервиса с экосистемой, профилями пользователей, возможностью добавления в друзья, просмотром истории поиска других пользователей и т. д. Система также может быть интегрирована в крупные музыкальные сервисы. Например, компания Яндекс успешно внедрила в свой музыкальный сервис функцию распознавания музыки, работающую по принципу Shazam'а, и потенциально заинтересована в развитии подобных технологий.

В перспективах развития предложенного сервиса возможно улучшение алгоритмов распознавания ритма: разработчики могут использовать новые методы и технологии машинного обучения и искусственного интеллекта для улучшения точности распознавания ритма и снижения ошибок, а также формирование персональных плейлистов, рекомендаций музыки на основе предыдущих поисков и учета предпочтений пользователя.

Список используемых источников

1. Shazam Statistics and User Count (2023): [Электронный ресурс] // DMR, 2023. URL: https://expandedramblings.com/index.php/shazam-statistics/ (дата обращения 12.05.2023).

2. McFee,B., Cai Y., McVicar M. librosa documentation [Электронный ресурс] // California: librosa, 2021. 245 p. URL: https://librosa.org/doc/main/ (дата обращения 12.05.2023).

Статья представлена доктором технических наук, профессором Птицыной Л. К.

УДК 004.67

С. В. Гончаров, О. И. Широпатин (студент гр. ИСТ-241м, СПбГУТ)

СОЗДАНИЕ МОДЕЛЕЙ И ИССЛЕДОВАНИЕ ТОПОЛОГИЧЕСКИХ ОПЕРАЦИЙ В POSTGIS

Статья посвящена созданию пространственных моделей и исследованию топологических операций в геоинформационной системе PostGIS. Она охватывает основные принципы построения пространственных моделей и применения топологических операций для анализа геоданных. Также в статье подробно рассмотрены возможности PostGIS для работы с геоданными. Статья представляет интерес для специалистов в области геоинформационных систем и всех, кто работает с геоданными. В ней описываются конкретные примеры применения топологических операций, что позволяет читателю лучше понимать, как эти операции могут быть использованы в решении практических задач. Также статья содержит подробное описание принципов работы с пространственными данными в PostGIS, что поможет читателю эффективнее использовать эту геоинформационную систему в своей работе.

геоинформационная система, пространственные данные, топологические операции, postgresql, postgis, de-9im.

PostGIS [1] – это открытое программное обеспечение, которое добавляет поддержку географических объектов в базу данных PostgreSQL.

По сути, PostGIS представляет из себя библиотеку для хранения географических данных.

В данной работе для взаимодействия с этими географическими данными используется географическая информационная система QGIS, предназначенная для просмотра, анализа и редактирования географических данных. С помощью QGIS пользователи получают возможность работать с различными типами пространственных данных, включая картографические слои, снимки спутников, данные дрона и многие другие.

Одной из особенностей QGIS является возможность интеграции с различными источниками данных, включая базы данных, сервисы веб-карт

и геопространственные стандарты. Пользователи могут импортировать данные из различных форматов, таких как PostGIS, Shapefile, GeoJSON, KML, и другие.

В качестве примера взаимодействия QGIS и PostGIS была создана таблица с координатами точек «Бермудского треугольника» (с помощью функции ST_MakePoint и указанных координат внутри нее). После выполнения функции координаты преобразуются в шестнадцатеричный формат (типа geometry), после чего они готовы для использования в среде QGIS. Пример изображен на рис. 1.

	id integer	location geography
1	1	0101000020E6100000D6C743DFDD0D54C0A8E0F08288C83...
2	2	0101000020E61000003C2EAA45448450C033333333337332...
3	3	0101000020E6100000F29716F5493350C0AE47E17A14264040

Рис. 1. Таблица с добавленными точками

Отображение данных точек в QGIS после импортирования из БД изображено на рис. 2. С помощью функции Points to Path создаются вектора, по которым создается полигон с помощью функции Lines to Polygon. Далее с помощью встроенной функции Field Calculator высчитывается площадь данного полигона. Исходные размеры выдаются в м^2, поэтому полученное число делится вручную на миллион в конструкторе формул. Подобное вычисление возможно благодаря изначальному добавлению карты мира OpenStreetMaps, поверх которой выполняются топологические операции. Также добавлено отображение площади данного треугольника.

Рис. 2. Отображение треугольника в QGIS

Взаимодействие с PostGIS возможно также посредством предварительной загрузки данных из, например, текстового файла (формата CSV), а затем выгрузки этих данных в базу данных PostgreSQL с библиотекой PostGIS. В таблице 1 представлен пример форматирования подобного файла, в котором содержатся названия городов с их координатами.

ТАБЛИЦА 1. Пример формата файла .csv с городами

Город	Широта	Долгота
Нью-Йорк	40,728333	–73,994167
Лос-Анджелес	34,033333	–118,266667

Полный список, задействованный в работе, содержит 25 городов. Их отображение представлено на рис. 3.

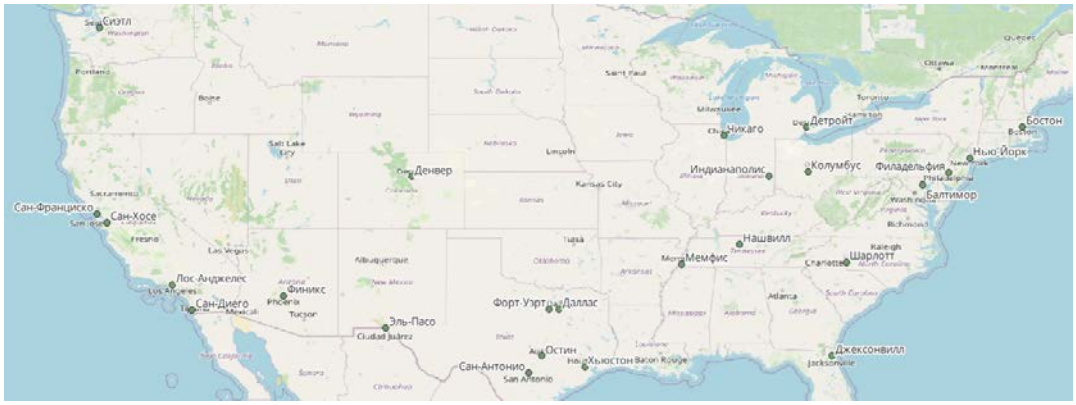


Рис. 3. Точки городов, отображенные на карте

Слой с городами, который был добавлен через файл можно экспортировать в базу данных как SQL дамп. Стоит отметить, что в этом дампе сохраняются как инвертированные координаты широты и долготы, так и координаты в шестнадцатеричном формате (типа *geometry*). Содержимое таблицы после загрузки дампа представлено на рис. 4.

Для определения пространственных отношений объектов применяется топологическая модель DE-9IM (“Dimensionally Extended 9-Intersection Model”) [2], которая является фреймворком для проведения топологических операций подобного вида.

В качестве рабочего примера применения этой модели используется карта Санкт-Петербурга с такими объектами, как зеленые зоны и окружные зоны метро радиусом 500 метров. Нужно с помощью наложения полигонов найти те станции, в пределах которых находятся зеленые зоны.

	ogc_fid [PK] integer	wkb_geometry geometry	field_1 character varying	field_2 double precision	field_3 double precision
1	1	0101000020E6100000CCF09F6EA07F52C07DCC07043A5D44...	Нью-Йорк	40.728333	-73.994167
2	2	0101000020E610000009FB761211915DC05470784144044140	Лос-Анджелес	34.033333	-118.266667
3	3	0101000020E61000009A99999999E955C03333333333F34440	Чикаго	41.9	-87.65
4	4	0101000020E61000007061DD7877D757C00000000000C03D40	Хьюстон	29.75	-95.366667
5	5	0101000020E6100000E256410C74CA52C0D7A02FBDFDF943...	Филадельфия	39.953056	-75.163333
6	6	0101000020E61000003C2EAA4544045CC09A99999999B94040	Финикс	33.45	-112.066667
7	7	0101000020E6100000684293C4929F58C0B3B45373B96D3D40	Сан-Антонио	29.428611	-98.493333
8	8	0101000020E61000066666666664A5DC0EC51B81E855B4040	Сан-Диего	32.715	-117.1625
9	9	0101000020E6100000D7A3703D0A3358C05C8FC2F528644040	Даллас	32.7825	-96.7975
10	10	0101000020E6100000C5AA4198DB775EC0DFC2BAF1EEA642...	Сан-Хосе	37.304167	-121.872778
11	11	0101000020E61000009A999999996954C0F1B8A81611513E40	Джексонвилл	30.316667	-81.65
12	12	0101000020E6100000F1BBE9961D4A55C04D4BAC8C46E243...	Индианаполис	39.767778	-85.158056
13	13	0101000020E6100000C4D155BABB9B5EC012F6ED2422E242...	Сан-Франциско	37.766667	-122.433333

Рис. 4. Представление в базе данных

Для определения отношений полигонов необходимо узнать конкретную матрицу, что можно осуществить с помощью следующего запроса: *SELECT ST_Relate('POLYGON (...)', 'POLYGON (...)');*. Узнав матрицу двух накладывающихся полигонов, можно найти все остальные накладывающиеся регионы запросом: *SELECT stations.* FROM stations JOIN areas ON ST_Intersects(areas.geom, stations.geom) WHERE ST_Relate(areas.geom, stations.geom, '212101212');*. На рис. 5 изображены регионы (зоны метро; зеленые зоны), в том числе найденные с помощью этого запроса (накладывающиеся).

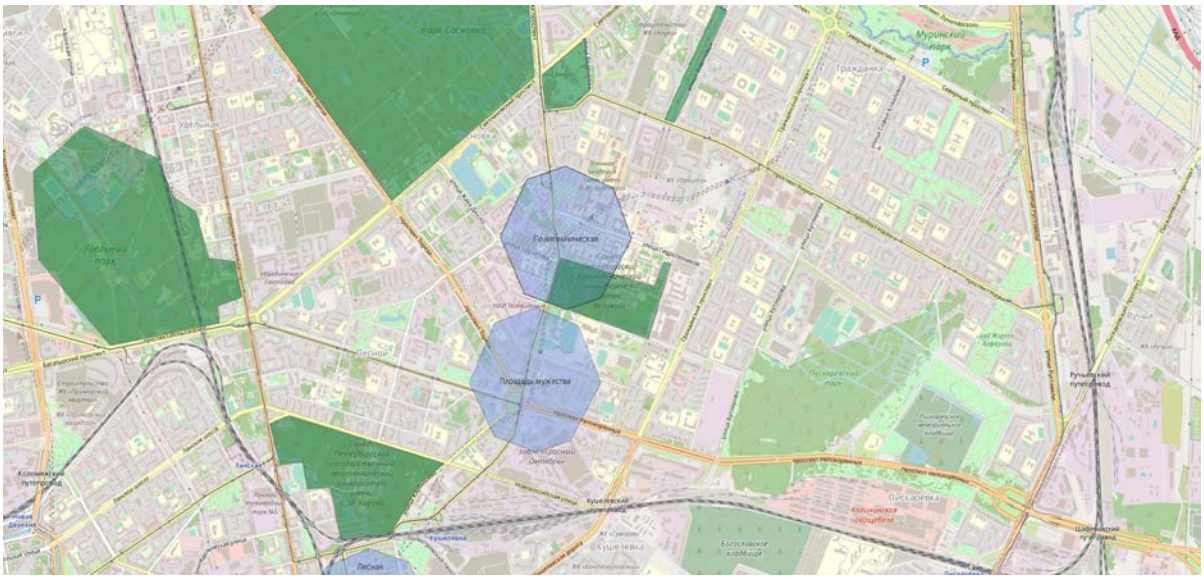


Рис. 5. Отображение регионов

Таким образом, PostGIS представляет из себя гибкое решение для работы с геоинформационными системами, по факту являясь библиотекой для хранения географических данных в PostgreSQL.

Список используемых источников

1. PostGIS [Электронный ресурс] / Wikipedia. URL: https://en.wikipedia.org/wiki/PostGIS
2. Dimensionally Extended 9-Intersection Model [Электронный ресурс] / URL: https://postgis.net/workshops/postgis-intro/de9im.html

Статья представлена научным руководителем доцентом кафедры ИСАУ СПбГУТ, кандидатом технических наук, доцентом Акимовым С. В.

УДК 004.432.2

И. И. Горбунов, К. А. Емельянов, П. В. Кириков, М. А. Ленкин, Р. А. Межуев (студенты гр. ИКПИ-12, СПбГУТ)

JAVA: СОВРЕМЕННОЕ ПРИМЕНЕНИЕ И КЛЮЧЕВЫЕ НАПРАВЛЕНИЯ ПРИ РАЗРАБОТКЕ

На сегодняшний день Java является одним из самых популярных языков программирования, используемых для разработки различных приложений. В настоящей работе рассматривается современное применение Java и ключевые направления разработки приложений с помощью данного языка: начиная с android-разработки и приложений для персонального компьютера, заканчивая большими корпоративными системами. В рамках статьи будут рассмотрены наиболее актуальные технологии, фреймворки и инструменты, применяемые в разработке.

java, jvm, spring, enterprise-приложения, android, javaFx, swing, MVC.

Java – высокоуровневый императивный объектно-ориентированный язык программирования со статической типизацией, созданный в 1995 году компанией Sun Microsystems. Он имеет ряд особенностей, которые делают его уникальным и популярным среди разработчиков, самая интересная из них – парадигма программирования WORA (англ. «*Write Once, Run Anywhere*» – напиши один раз, запускай где угодно).

Java-код компилируется в байт-код, который может быть выполнен на любой платформе, на которой установлена виртуальная машина Java (*Java Virtual Machine*). Это означает, что разработчики могут написать код один раз и запускать его на различных платформах без необходимости перекомпиляции и переписывания приложения для каждой платформы.

Из этой особенности вытекают еще несколько немаловажных преимуществ Java:

- Безопасность: использование виртуальной машины Java помогает обеспечить безопасность приложения, так как байт-код проходит проверку на безопасность перед запуском на JVM.
- Управление памятью и улучшенная производительность: JVM контролирует доступ к памяти и автоматически уничтожает неиспользуемые объекты, а также поддерживает многопоточность.

Настольные приложения

Разработка настольных приложений имеет большое значение в современном мире. Удобство использования, функциональность и эстетичность приложения очень важны для повседневного пользования. Именно для этого и создаются такие инструменты как JavaFX и Swing, которые описаны в статье.

JavaFX

JavaFX – это набор графических и мультимедийных пакетов, который позволяет разработчикам проектировать, создавать, тестировать, отлаживать и развертывать расширенные клиентские приложения, которые работают согласованно на различных платформах.

Фреймворк обладает мощными возможностями в создании графически богатых пользовательских интерфейсов. JavaFX предоставляет функции для добавления в приложение различных визуальных эффектов и анимации, 3D-графики. Одно из ключевых преимуществ JavaFX заключается в том, что он обеспечивает высокую степень переносимости, что позволяет программистам работать на разных операционных системах без необходимости внесения изменений в код.

Структура приложения на JavaFX может выглядеть следующим образом:

- Главный класс приложения: здесь происходит инициализация элементов пользовательского интерфейса и запуск приложения.
- Классы модели: содержат данные приложения и логику их обработки.
- Классы контроллера: отвечают за обработку событий, полученных от пользовательского интерфейса, и взаимодействие с моделью.
- Классы представления: отвечают за отображение данных из модели и обновление пользовательского интерфейса.
- FXML-файлы, отвечающие за расположение графических элементов.

Swing

Swing – это легкий инструментарий с графическим интерфейсом, который имеет широкий спектр виджетов для создания оптимизированных оконных приложений.

Структура Swing-приложений обычно включает следующие компоненты:

- Класс-наследник JFrame – это основное окно приложения. Внутри этого класса определяются все компоненты GUI, такие как кнопки, текстовые поля, метки и т. д.
- Панель содержимого – это контейнер, который содержит все компоненты GUI, расположенные на основном окне приложения. Обычно используется класс-наследник JPanel.
- Меню – это набор команд, доступных пользователю для выполнения операций в приложении.
- Размещение компонентов – это процесс определения расположения компонентов на панели содержимого. Для этого используются менеджеры размещения, такие как BorderLayout, GridLayout и т. д.
- Обработчики событий – это методы, которые вызываются при возникновении определенного события, например, щелчка мыши или нажатия клавиши.
- Диалоговые окна – это специальные окна, которые появляются при необходимости запросить у пользователя какие-то данные или сообщить о каком-то событии. Для создания диалоговых окон можно использовать классы JOptionPane или JDialog.

Android-разработка

В данный момент Android разработка является крайне актуальной из-за широкого распространения смартфонов. Java является одним из наиболее популярных языков программирования для разработки приложений под Android.

Структура Android-приложений на Java может быть различной, но обычно она включает следующие основные компоненты:

- Activities (Активности) – компоненты, которые управляют пользовательским интерфейсом приложения и его взаимодействием с пользователем.
- Services (Сервисы) – компоненты, которые выполняют фоновые задачи, не связанные непосредственно с пользовательским интерфейсом.
- Broadcast receivers (Приёмники сообщений) – компоненты, которые принимают оповещения от системы или других приложений.

- Content providers (провайдеры контента) – компоненты, которые предоставляют доступ к данным приложения через интерфейсы, связанные с базами данных или файловой системой.
- Layouts (интерфейсы пользователя) – это набор элементов пользовательского интерфейса, таких как кнопки, поля ввода, списки и т. д., которые можно настраивать и располагать на экране.
- Resources (ресурсы) – это файлы, такие как изображения, тексты, звуки и другие элементы приложения.
- Manifest (манифест) – это файл, который содержит информацию обо всех компонентах приложения, его разрешениях и других настройках.

Эти компоненты могут быть организованы в соответствии с конкретными потребностями приложения, например, одна активность может использовать несколько разных интерфейсов пользователя, а сервис может запускаться отдельно или вместе с активностью. Также возможно использование дополнительных библиотек и инструментов для разработки и оптимизации Android-приложений на Java.

Enterprise-приложения

Enterprise-приложения – это приложения, разработанные для управления сложными бизнес-процессами и обеспечения масштабируемости и надежности в больших организациях. Они используются для автоматизации бизнес-процессов, управления базами данных, интеграции с другими системами или сервисами. Бизнес-приложения на Java обычно создаются с использованием таких технологий, как Spring Boot, Spring Security, Hibernate, Maven, Docker, SQL и так далее.

Важным аспектом структуры enterprise-приложений является модульность. Часто приложения разбиваются на отдельные модули, которые могут быть разработаны и развернуты отдельно друг от друга. Это позволяет более гибко управлять приложением и обеспечивать его масштабируемость. Ниже представлена и описана структурная схема такого приложения (рис. 1).

Обычно, выделяют следующие слои: слой представления, слой бизнес-логики, контроллер и слой доступа к данным. Каждый слой выполняет свои задачи и взаимодействует с другими слоями через определенные интерфейсы.

Слой представления обеспечивает взаимодействие пользователей с приложением. Этот слой может содержать различные элементы интерфейса, такие как веб-страницы, формы, графические элементы и т. д. В приложениях на Java часто для создания представлений используют такие библиотеки как JSF и Thymeleaf.

Контроллеры – это часть архитектуры, которая обрабатывает входящие запросы и управляет логикой обработки запросов и формирования ответов. Общение между контроллером и клиентом происходит по протоколу

HTTP. Контроллеры могут обрабатывать различные виды запросов, тела запросов, заголовки.

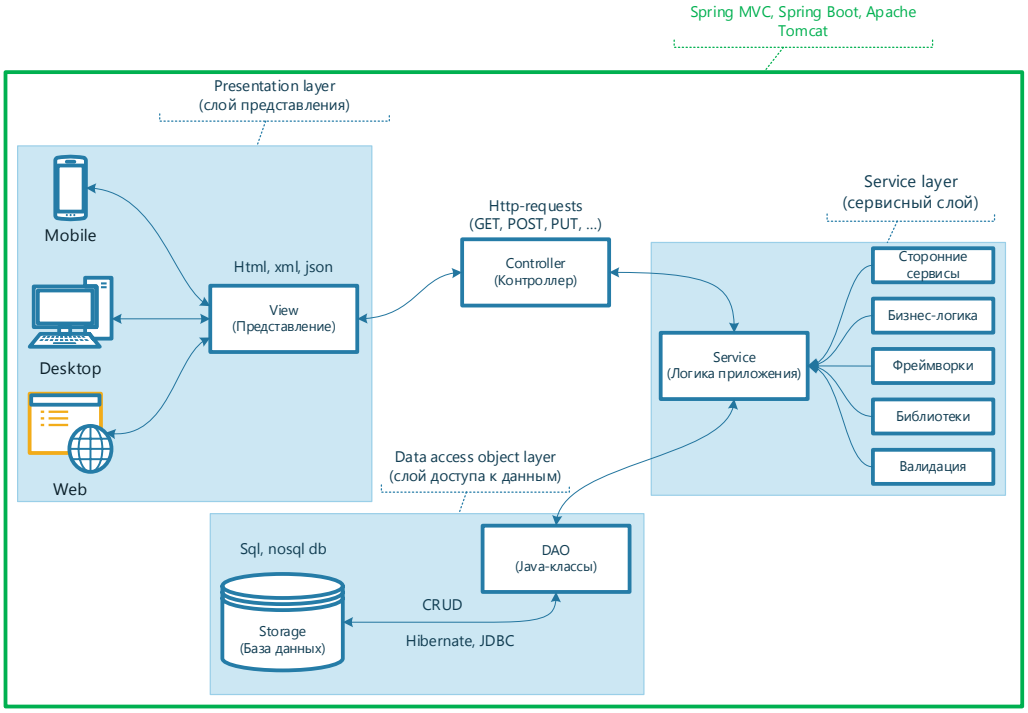


Рис. 1. Упрощенная структура enterprise-приложения

Слой бизнес-логики (сервисный слой) который выполняет бизнес-операции и обрабатывает данные, полученные от слоя представления. В этом слое выполняются операции, связанные с бизнес-логикой приложения, такие как валидация данных, расчеты, интеграция с другими сервисами.

Слой доступа к данным – это слой, который обеспечивает доступ к данным, используемым в приложении. Этот слой может использовать различные технологии для работы с базами данными, такие как JDBC, JPA и Hibernate.

Заключение

Результатом статьи стало понимание того, какие технологии и инструменты можно использовать для создания конкретного приложения. Описанные возможности и технологии помогут разработчикам выбирать наиболее подходящие решения и строить эффективные приложения, соответствующие современным стандартам и требованиям пользователей.

Таким образом, современная Java широко используется для создания как десктоп-приложений, так и мобильной разработки, веб-приложений, сервисов, или бизнес-процессов. Java позволяет создавать универсальные и надежные приложения, которые могут работать на разных операционных системах и устройствах.

Список используемых источников

1. Динеш Р. Spring. Все паттерны проектирования. СПб. : Питер, 2019.320 с.
2. Baeldung: spring tutorials [Электронный ресурс]. URL: https://www.baeldung.com/ (дата обращения 02.05.2023).
3. Документация Java [Электронный ресурс]. URL: https://docs.oracle.com/en/java/index.html (дата обращения 10.05.2023).
4. Infiraise. Best java framework for desktop applications [Электронный ресурс]. URL: https://www.infiraise.com/what-is-the-best-java-framework-for-desktop-applications/ (дата обращения 17.05.2023).

Статья представлена научным руководителем,
ассистентом кафедры ЗСС СПбГУТ Катасоновым А. И.

УДК 004.514

А. А. Грибовский (студент гр. ИСТ-941, СПбГУТ)

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА НА БАЗЕ ТЕХНОЛОГИИ BLAZOR WEBASSEMBLY

В статье представлены результаты разработки библиотеки компонентов для создания модульного пользовательского интерфейса с использование технологии Blazor WebAssembly. Разработанная библиотека компонентов Blazor создана с единых методологических позиций с использованием магистрального принципа построения систем. Компоненты имеют унифицированные API, что позволяет на их основе выполнять комплексирование достаточно сложных пользовательских интерфейсов. Внедрение данной библиотеки будет способствовать стандартизации веб-приложений, предоставляет единый набор инструментов и паттернов элементов правления, которые применимы в различных проектах и командах разработчиков.

пользовательский интерфейс, библиотека компонентов, интерфейс, WebAssembly, Blazor, UX/UI, .NET, C#, ASP.NET Core.

Пользовательский интерфейс в информационных системах является важным элементом, который реализует взаимодействие человека и технической системы [1]. В процессе проектирования интерфейсов выделяются две составляющие – UX и UI, которые играют центральную роль в создании современных информационных пользовательских продуктов. Эффективные UX/UI, основанные на современных тенденциях, улучшают взаимодействие с продуктом и способствуют достижению целей информационного взаимодействия. Решение задач по качественной реализации эффективного интерфейса требует обеспечения качества его составляющих.

Процесс разработки графического пользовательского интерфейса (GUI) является достаточно трудоемкой задачи и требует достаточно высокого уровня разработчиков, поэтому настоящее время существует необходимость в создании унифицированных компонентов, которые позволяют эффективно разрабатывать пользовательские интерфейсы с применением единообразно оформленных элементов управления.

Создание библиотеки унифицированных компонентов Blazor будет способствовать снижению временных и финансовых затрат на разработку пользовательского интерфейса цифрового продукта. Повторное использование компонентов предоставляет возможность внедрения новых функций и взаимодействий с ранее реализованным функционалом, исключив существенную часть временных затрат благодаря отсутствию необходимости написания программных модулей с нуля, что ускоряет процесс разработки и позволяет оперативно реагировать на изменения потребностей пользователей и требований к продукту.

Разработанная библиотека компонентов Blazor создана с единых методологических позиций с использованием магистрального принципа построения систем. Компоненты имеют унифицированные API, что позволяет на их основе выполнять комплексирование достаточно сложных пользовательских интерфейсов. Внедрение данной библиотеки будет способствовать стандартизации веб-приложений, предоставляет единый набор инструментов и паттернов элементов правления, которые применимы в различных проектах и командах разработчиков. Положительными эффектами внедрения компонентного подхода станет повышение эффективности коллективной работы, обеспечение согласованности и единообразия в разработке веб-приложений.

Выбор Blazor платформы обусловлен ее высоким потенциалом. Данная платформа предназначена для создания интерактивного веб-интерфейса на стороне клиента с использованием технологии Blazor WebAssembly [2, 3] и предоставляет следующие основные возможности:

– создание многофункциональных интерактивных пользовательских интерфейсов на языке C#, вместо традиционно используемого JavaScript;

– совместное использование серверной и клиентской логик приложений .NET Core;

– отображение пользовательского интерфейса в виде HTML-страницы с CSS для широкой поддержки браузеров, в том числе для мобильных устройств;

– интеграция с современными платформами контейнеризации (Docker);

– создание классических и мобильных приложений с помощью .NET и Blazor.

Blazor WebAssembly позволяет разработчикам создавать веб-приложения с использованием C# и .NET без необходимости использования подключаемых модулей браузера или дополнительной среды выполнения. Технология обеспечивает безопасную изолированную среду выполнения, что делает его предпочтительным инструментом для создания веб-приложений. WebAssembly представляет собой портативную виртуальную машину, которая запускается в браузере пользователя и реализует изолированной среды выполнения кода. Технология WebAssembly во многом похожа на ассемблер, и ее часто называют браузерным ассемблером. WebAssembly позволяет транслировать код в набор низкоуровневых инструкций, что обеспечивает возможность создания высокопроизводительного программного обеспечения, запускаемого непосредственно в потоке браузера без необходимости установки каких-либо плагинов. Blazor WebAssembly поддерживает несколько языков .NET, включая C#, F# и Visual Basic. Данные языки используются для написания кода, который запускается непосредственно в браузере, что обеспечивает быстрое взаимодействие с пользователем.

Пример графического пользовательского интерфейса, разработанного с помощью созданной библиотеки компонентов представлен на рис. 1.

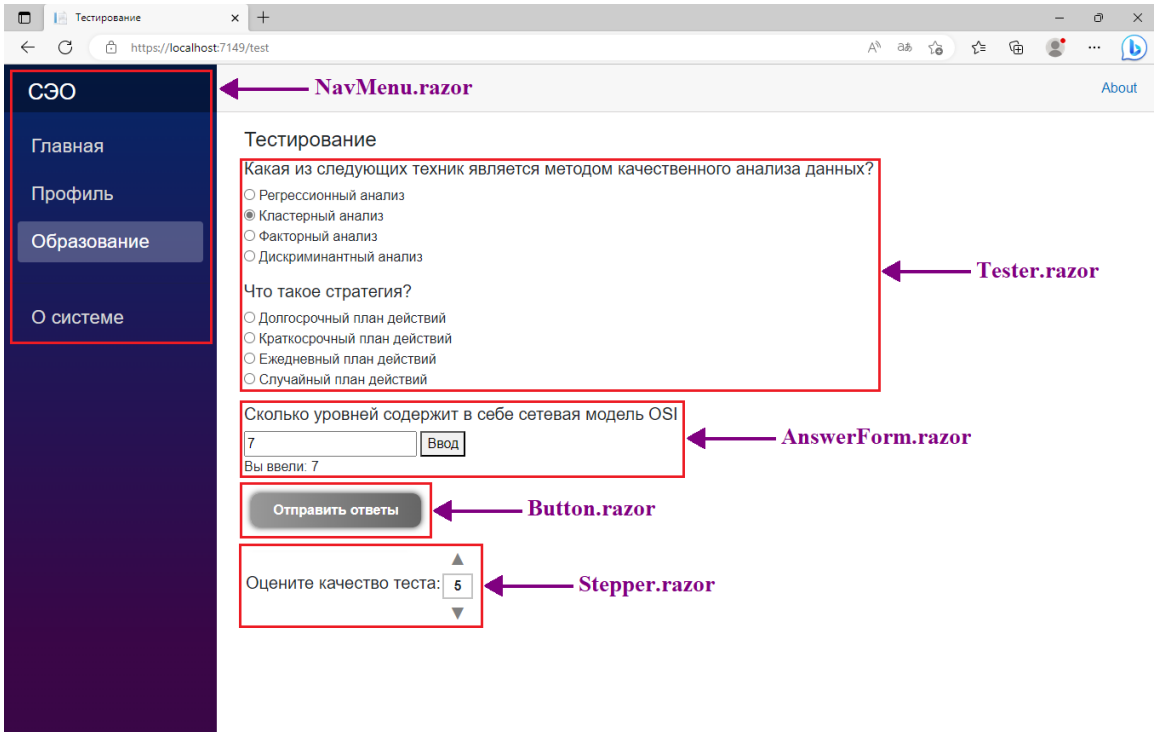


Рис. 1. Пример графического пользовательского интерфейса, созданного с применением разработанной библиотеки компонентов

В левом краю экрана отображается меню приложения, включающее в себя номенклатуру страниц проекта и методы доступа к ним, реализованные с помощью компонента NavMenu.razor. Основное поле страницы содер-

жит в себе наименование страницы, вопросы, кнопки выбора варианта ответа, реализованные с помощью компонента Tester.razor, поле ввода с кнопкой подтверждения, реализованные с помощью компонента AnswerForm.razor, и кнопку отправки данных, реализованную с помощью компонента Button.razor, кроме того, поле содержит в себе элемент оценки теста обучающимся в виде компонента Stepper.razor. Разметка страницы реализована средствами CSS и HTML в рамках технологи Razor.

Разработанная библиотеки компонентов Blazor предоставляет возможность создания графического интерфейса веб-приложения с иерархической модульной структурой, в который каждый компонент выполняет возложенную на него конкретную функцию. Созданные компоненты обладают способностью интеграции с различными технологиями и фреймворками.

Список используемых источников

1. ГОСТ Р ИСО 9241-161-2016 Эргономика взаимодействия человек–система. Часть 161. Элементы графического пользовательского интерфейса, 2016.

2. WebAssembly Specification Release 2.0 (Draft 2023-04-24) [Электронный ресурс]. URL: https://webassembly.github.io/spec/core/_download/WebAssembly.pdf (дата обращения 16.05.2023).

3. Жизненный цикл компонента Razor ASP.NET Core [Электронный ресурс]. URL: https://learn.microsoft.com/ru-ru/aspnet/core/blazor/components/lifecycle?view=aspnetcore-6.0 (дата обращения 16.05.2023).

Статья представлена научным руководителем, заведующей кафедрой ИСАУ СПбГУТ, доктором технических наук, профессором Верховой Г. В.

УДК 004.356.2

Н. С. Доманцевич, В. А. Проскуряк, А. А. Саидов
(студенты гр. ИКПИ-12)

3D-ПРИНТЕР: ПЕРСПЕКТИВЫ РАЗВИТИЯ И КЛЮЧЕВЫЕ ТЕХНИЧЕСКИЕ ПРИНЦИПЫ

Статья посвящена такому техническому устройству как трехмерный принтер (далее 3D-принтер), она раскрывает технические принципы его работы, информирует о различных технологиях 3D печати и повествует о будущих перспективах данного технического устройства.

3D-принтер, CAD-программа, программа-слайсер, трехмерные объекты.

Основной принцип работы 3D-принтера включает следующие этапы

1. В начале процесса на компьютере используется специальная CAD-программа для создания модели объекта.

2. После завершения моделирования, объект сохраняется в специальном формате и передается программе-слайсеру, которая поставляется в комплекте с устройством. Программа-слайсер разбивает модель на тонкие слои, причем толщина каждого слоя зависит от возможностей конкретного 3D-принтера и выбранных настроек.

3. Затем каждый слой преобразуется в двоичный командный код, который передается устройству. Устройство использует этот код для точного позиционирования и осуществления движения печатающей головки, которая наносит материал на соответствующие координаты слоя.

4. Процесс нанесения материала повторяется слой за слоем, пока не будет сформирован полноценный объект.

5. В результате этих шагов 3D-принтер позволяет создавать трехмерные объекты, точно воспроизводя модель, которая была создана в программе CAD.

Устройство 3D-принтера

Базируется на общей схеме, которая позволяет ему работать в трех измерениях, осуществляя линейное движение. Эти устройства оснащены высокоточными шаговыми двигателями и контроллером, который управляет последовательностью движения этих двигателей. Автоматизированная система перемещает печатающую головку и контролирует выдавливание материала в нужный момент, например, расплавленной пластмассы (рис. 1). Пошагово формируется фигурка, которая изначально задана в программе. Этому процессу лежит в основе принцип работы «картезианского робота», который способен перемещаться в соответствии с координатами *X*, *Y* и *Z*, известными также как декартовы координаты [1, 2].

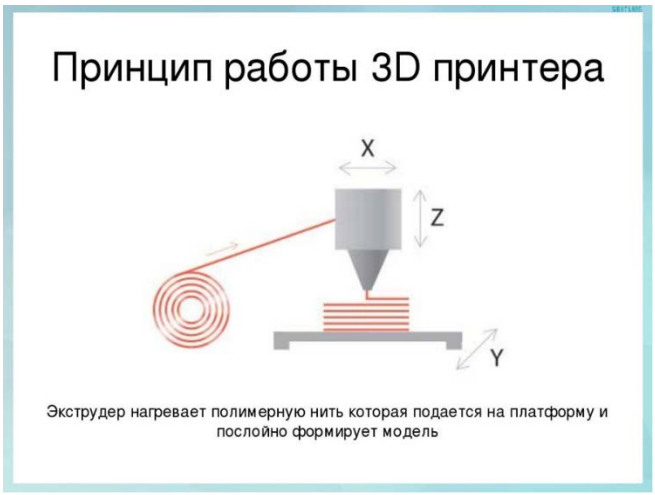


Рис. 1. Схема работы 3D-принтера

Компоненты 3D-принтера включают

1. Экструдер: Этот компонент является одним из ключевых и сложных элементов принтера, который часто улучшается в новых моделях. Экструдер состоит из термальной головки и привода, который выдавливает пластиковую нить. Процесс работы следующий: катушка с нитью загружается в принтер, привод разматывает и выталкивает нить, подводя ее к термальной головке (также называемой камерой). Головка обычно выполнена из нагреваемого алюминиевого элемента, который расплавляет нить. Расплавленный материал выдавливается через отверстие печатающей головки в полужидком состоянии.

2. Линейный двигатель: Скорость печати и долговечность 3D-принтера зависят от типа линейного двигателя. Для каждой координатной оси используется отдельный гладкий стержень, который работает совместно с подшипниками. Подшипники могут быть изготовлены из различных материалов, таких как пластик, сталь, бронза и другие. Бронзовые подшипники являются наиболее сложными в калибровке во время сборки, но они обычно менее шумные.

3. Фиксаторы: Они необходимы, чтобы предотвратить выход линейных приводов за пределы рабочего поля. Они не влияют на функциональность устройства, но их наличие делает печать более точной и аккуратной. Встречаются модели с оптическими или механическими фиксаторами.

4. Платформа: Она представляет собой поверхность, на которой создается готовая модель. Ее размер обычно составляет от 100 до 200 квадратных миллиметров. Производители часто делают платформу подогреваемой, чтобы предотвратить появление трещин или разрывов на модели, обеспечить хорошую сцепку между слоями и между первым слоем и платформой. Платформы обычно изготавливаются из алюминия или стекла, материалов с хорошей теплопроводностью.

Существует несколько различных технологий 3D-печати

- Стереолитография (SLA): в этой технологии основным материалом является смесь жидкого полимера с реагентом, который отвердевает пластик. Ультрафиолетовый лазер применяется для полимеризации смеси в нужные моменты. Фигура создается на подвижной платформе, связанной с небольшим лифтом, который перемещает заготовку вверх или вниз на один слой. Лазерная луч проходит через полимер и останавливается на местах, где необходимо отвердение. Затем лифт поднимает или опускает заготовку для создания следующего слоя.
- Выборочное лазерное спекание (SLS): в металлообработке применяется технология SLS. В этом случае используется композитный порошок, состоящий из частиц размером 50–100 мкм. Порошок равномерно наносится слой за слоем, а затем лазер «запекает» его. Эта технология экономична

и практически не оставляет отходов, по сравнению с традиционными методами, такими как резка, литье, фрезеровка и сверление.

• Многоструйное моделирование: Это уникальная разработка компании 3D Systems, похожая на стандартную струйную печать в обычных принтерах. В этом процессе используется несколько десятков или даже сотен сопел, расположенных на печатающей головке. «Чернила» нагреваются, и слоями наносятся на рабочую поверхность, где они отвердевают при комнатной температуре.

Это только основные и наиболее распространенные методы, но на самом деле существует множество более редких и узкоспециализированных вариантов, таких как УФ-облучение через фотомаску (SGC), склеивание пленок, склеивание порошков, ламинирование листовых материалов (LOM) и другие.

3D-печать имеет огромный потенциал и перспективы в различных областях

• Строительство: 3D-печать домов или контурное строительство представляет собой будущее строительной индустрии. Китай и правительство Дубая уже начали экспериментировать с 3D-печатью зданий. В Дубае уже построено первое 3D-печатное офисное здание, а планируется также печать целого квартала. Недавно в Европе был создан первый напечатанный дом на 3D-принтере.

• Электроника: 3D-печать электроники представляет будущее производства цифровых устройств. В настоящее время проводятся исследования свойств графена и его применения в 3D-печати. Создание графенового аккумулятора на 3D-принтере с неограниченным сроком эксплуатации представляет огромный потенциал в этой области.

• Автомобильная и аэрокосмическая промышленность: 3D-печать широко применяется при разработке автомобилей, самолетов и спутников из-за способности точно воспроизводить сложные элементы. На Международной космической станции уже используется 3D-принтер, и есть успешные результаты печати автомобилей.

• Фармацевтическая промышленность: 3D-печать находит применение в производстве таблеток и других лекарственных средств. Некоторые лекарства, такие как таблетки от эпилепсии, уже изготавливаются по специальной методике 3D-печати, что позволяет постепенное высвобождение активных веществ и упрощение приема медикаментов.

• Пищевая промышленность: 3D-принтеры для печати пищи начинают использоваться в кафе и ресторанах. Хотя эта область 3D-печати находится на ранних стадиях развития, она имеет большой потенциал для создания индивидуально настроенных блюд и экспериментов с текстурой и вкусом пищи.

Список используемых источников

1. Принципы работы устройств. URL: https://principraboty.ru
2. https://3ddevice.com/

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.5

Н. С. Ершова, Р. С. Ноянов (студент гр. ИКБ-14, СПбГУТ)

ТЕЛЕГРАМ-БОТЫ, КАК СРЕДСТВО АВТОМАТИЗАЦИИ ЭЛЕМЕНТОВ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Технологии развиваются с каждым днем все быстрее, в наше время практически все сферы деятельности переходят в цифровой формат, и образование не является исключением. В России одной из целей современного образования является его информатизация, активно внедряются современные информационные технологии в целях повышения качества и эффективности обучения. В данной статье рассматривается, как Телеграм-боты способствуют автоматизации элементов процесса образования в России.

информационные технологии, автоматизация, боты, образование, эффективность, электронные средства обучения.

Ежедневно в учебных заведениях люди используют информацию на физических носителях, таких как печатные издания книг или бумажные материалы лекций, а также проходят множество тестирований на листах бумаги или в тетрадях. Такой подход с книжными переплетами и бумажными рукописями бесспорно широко распространен и удобен, но лишь на небольшие объемы материала. Как преподавателям, так и ученикам носить с собой на занятия тяжелые учебники и множество бумаг крайне затруднительно. Экологическая составляющая физических носителей является немаловажным фактором, повлиявшим на попытку уменьшения их использования, по данным статистики, для создания бумаги ежегодно вырубают около двухсот двадцати миллионов деревьев в России [7].

Переход на электронные средства обучения и образовательное программное обеспечение, позволят автоматизировать процесс обработки и проверки полученных данных, контроль учащихся и сбор информации об обучении [1]. Одним из плюсов такого подхода является возможность сортировки информации, что позволяет сократить время на ее ручную обработку, и, следовательно, освободить время на получение новых знаний для

учеников и уменьшить нагрузку на преподавателей. В эпоху цифровых технологий, электронные образовательные ресурсы становятся все более доступными и удобными для различных категорий студентов. Немаловажным показателем их распространения становится беспрепятственное получение информации, учебных материалов, видео лекций и других образовательных ресурсов. Они могут быть предоставлены всем желающим, кто имеет право на их получение, что особенно важно для тех, кто живет в отдаленных регионах страны или не может позволить себе оплату обучения в традиционных учебных заведениях. В цифровых образовательных средах управление и ограничение доступа реализовано путем проведения авторизации, для этого используются логин и пароль, сертификаты, биометрические данные и другие методы, также важно разделение пользователей по группам с разными уровнями доступа и возможностями.

На сегодняшний день Телеграм – один из самых популярных мессенджеров в мире, поддерживающий множество функций, применяемых пользователями для решения различного рода задач [4]. Тенденцией данной системы стали боты. По своей сути они являются программами внутри мессенджера, которые управляются текстовыми командами в чате по принципу «вопрос – ответ» [3] или работают на основе рассылки, на рис. 1 изображена концепция работы Телеграм-бота.

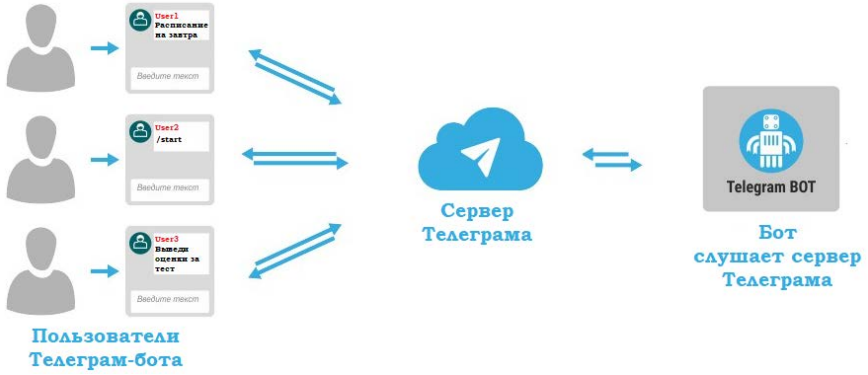


Рис. 1. Принцип работы Телеграм-бота

✓ Боты в Телеграмм, являясь программой, способны выполнять функции, которые задаст разработчик, они позволяют автоматизировать различные процессы в мессенджере. Они могут быть использованы для упрощения и оптимизации различных задач, в том числе и в образовательном процессе. К их функционалу, который уже активно используется, относятся сбор данных, формирование статистики об учащихся, поддержка обратной связи, массовая рассылка информации, формирование тестов с выводом правильных ответов и результатов их прохождения, предложение дополнительных заданий, информации и ссылок на её источники на основе обработанных данных, одним из значимых возможностей ботов является коммуникация

с пользователем в режиме реального времени, боты могут иметь базовый набор команд, позволяющий мгновенно давать ответ на запрос [2].

✓ Существующие боты, применяемые в образовании, служат для уведомления о расписании занятий, отправки домашних заданий, напоминании о сдаче работ, автоматической проверки заданий, выставлению оценок, предоставления индивидуальной обратной связи, примеры таких ботов – @QuizBot в Телеграмме, который позволяет создавать тесты с несколькими вопросами и множественным выбором вариантов ответов. В Российских ВУЗах также применяются боты, например, для быстрого доступа к расписанию занятий, его парсинга и вывода в удобном для пользователя формате, в СПбГУТ за это отвечает Телеграмм-бот @BonchGUTBot [5], в СПбПУ @polyrobot в ВКонтакте, также существуют боты, помогающие абитуриентам, в СПбПУ это @SPbPUbot [6], а @gutka_sut является интеллектуальным ботом-справочником для студентов СПбГУТ.

✓ Мы приступили к разработке Телеграмм бота, который предоставляет возможность создания и прохождения тестов с выбором ответа, для начала работы с ботом необходимо пройти процедуру регистрации, которая разделяет пользователей на студентов и преподавателей. Преподавателям доступны создание, изменение и удаление тестов, формирование результатов прохождения конкретного теста. Доступ к режиму преподавателя ограничен паролем. Режим студента позволяет проходить тесты и просматривать свои баллы за него, доступ к тесту происходит по его идентификатору (в данном боте это название теста), который сообщает преподаватель. Для написания бота мы используем Telegram Bot API, язык программирования Python, а в качестве среды разработки PyCharm, который поддерживает систему контроля версий и совместную разработку.

Список используемых источников

1. Бариева А. А. Внедрение современных информационных технологий в образовательный процесс // Актуальные вопросы современной педагогики: материалы VI Междунар. науч. конф. (г. Уфа, март 2015 г.). Уфа: Лето, 2015. С. 228–230. URL: https://moluch.ru/conf/ped/archive/148/7414/ (дата обращения 23.04.2023).

2. Богушевич Е. Чат-боты – будущее образования. Педсовет [Электронный ресурс]. URL: https://pedsovet.org/article/cat-boty-budusee-obrazovania (дата обращения 20.04.2023).

3. Гринзовский И., Скворцов В. Как самостоятельно сделать дота в «Телеграме» без навыков программирования. Тинькофф журнал [Электронный ресурс]. URL: https://journal.tinkoff.ru/guide/howto-telegram-bot/ (дата обращения 23.04.2023).

4. Смирнов С. Самые популярные мессенджеры в 2022-2023 году: подробно о выборе пользователей в России и в мире. Интернет-Маркетинг [Электронный ресурс]. URL: https://sergeysmirnovblog.ru/socseti/samie-populyarnie-messendzheri-v-rossii-i-v-mire.html (дата обращения 16.04.2023).

5. «BonchGUTBot»: просмотр расписания. Моё образование [Электронный ресурс]. URL: https://moeobrazovanie.ru/publikacii/novosti/187867.html (дата обращения 01.05.2023).

6. Запущен телеграм-бот для абитуриентов Политеха. Поступление 2023 [Электронный ресурс]. URL: https://www.spbstu.ru/abit/events/zapushchen-telegram-bot-dlya-abiturientov-politekha/?ysclid=lgwlv7axly435438920 (дата обращения 01.05.2023).

7. Лес [Электронный ресурс]. URL: https://greenpeace.ru/projects/forest/ (дата обращения 14.04.2023).

Статья представлена научным руководителем, доцентом кафедры ИКД СПбГУТ, кандидатом педагогических наук, доцентом Шияном А. А.

УДК 004.9

А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)
М. Ю. Кужлев (студент гр. ИСТ-022, СПбГУТ)

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ДИСТАНЦИОННОГО МОНИТОРИНГА ЗДОРОВЬЯ ПАЦИЕНТА

Приведена актуальность внедрения автоматизированных систем дистанционного мониторинга здоровья пациентов в медицинскую сферу. Проведен анализ использования технологических средств для медицинского учета пациентов. Выявлены основные недостатки существующих систем мониторинга здоровья пациентов. Разработана архитектура информационной системы, предназначенной для совершенствования медицинского учета населения. В процессе проектирования информационной системы разработаны диаграммы при помощи графического языка объектного моделирования Unified Modeling Language.

дистанционный мониторинг здоровья, автоматизированный мониторинг здоровья, информационная система, модель системы, проектирование, UML-диаграммы.

Дистанционный мониторинг здоровья имеет огромное значение в сфере телемедицины поскольку показывает текущее состояние пациента и позволяет оценить его состояние. Целью процесса наблюдения за пациентом является четкий анализ состояния, а также своевременное обнаружение отклонений от нормы в жизненных показателях.

Дистанционный мониторинг увеличивает частоту контактов между провайдером и получателем медицинской помощи при хронических неинфекционных заболеваниях, что уменьшает необходимость встречи лицом к лицу с врачом и необходимость госпитализации. Это особенно важно, если из-за больших расстояний и отсутствия транспорта амбулаторное посещение затруднительно. Дистанционный мониторинг обеспечивает непрерывность лечения и повышает компетентность, что может вести к снижению затрат системы здравоохранения, так как с помощью мониторинга

не только повышается качество ухода за хроническими больными, но и снижается длительность госпитализаций [1].

Современный подход к дистанционному анализу немыслим без автоматизированных систем оценки здоровья пациента. Однако существующие на сегодняшний день системы оценки здоровья пациента имеют ряд недостатков, основные из которых: нехватка карты с отображением последних перемещений пациента, а также отсутствие иерархического разделения врачей в системах – все врачи имеют доступ ко всем пациентам, что увеличивает риск путаницы и утечки данных. Помимо указанных функциональных недостатков, на необходимость подобных систем указывает почти полное отсутствие выбора медицинских контор, предоставляющих дистанционный мониторинг, а также нехватка врачей, особенно высококвалифицированных узких специалистов. Предлагаемая система может использоваться для профессиональной коммуникации с более опытными коллегами из других городов [2].

Вследствие этого современное общество ставит перед собой задачу внедрения автоматизированного дистанционного мониторинга здоровья пациента для обеспечения безопасности и здоровья населения. Важной составляющей мониторинга является своевременное реагирование, впоследствии чего повышается вероятность своевременного оказания необходимой медицинской помощи пациенту. Таким образом, особую важность приобретает разработка системы способной точно и быстро сообщить врачу, что с пациентом что-то случилось и ему необходима медицинская помощь.

Архитектура системы автоматизированного дистанционного мониторинга здоровья пациента представлена на рис. 1 (см. ниже).

Существует множество способов контроля за состоянием пациента, но мало кто уделяет внимание контролю местоположения пациента, однако это необходимо, поскольку может помочь быстро среагировать и направить нужные службы на место происшествия и спасти человеку жизнь.

Стоит обратить внимание на то, к каким пациентам врачи получают доступ – в системе предусмотрено три роли: врач 1 уровня (главный врач), врач 2 уровня (врач), врач 3 уровня (интерн). Главный врач имеет доступ ко всем пациентам, может добавлять новых врачей и интернов, а также может назначать врачам новых пациентов. Врачи имеют возможность переназначать пациентов на других врачей либо же интернов. Также врач имеет возможность назначать пациенту лекарства и редактировать данные о пациенте. Интерн имеет возможность видеть только тех пациентов, которых ему назначили, назначать им лекарства и редактировать их данные. Помимо врачей, пользователями системы являются пациенты. Пациент приходит на первую консультацию лично, в процессе которой заполняются все необходимые данные, такие как: принимаемые лекарства, история о болезнях и личные данные. Затем пациент получает личные наручные часы, которые собирают информацию о его здоровье и передают данные в зашифрованном

виде на сервер, после чего они попадают в базу и отображаются в анкете пациента. Пациент, в свою очередь, имеет возможность просматривать свои показатели в личном кабинете, в том числе принимаемые лекарства.

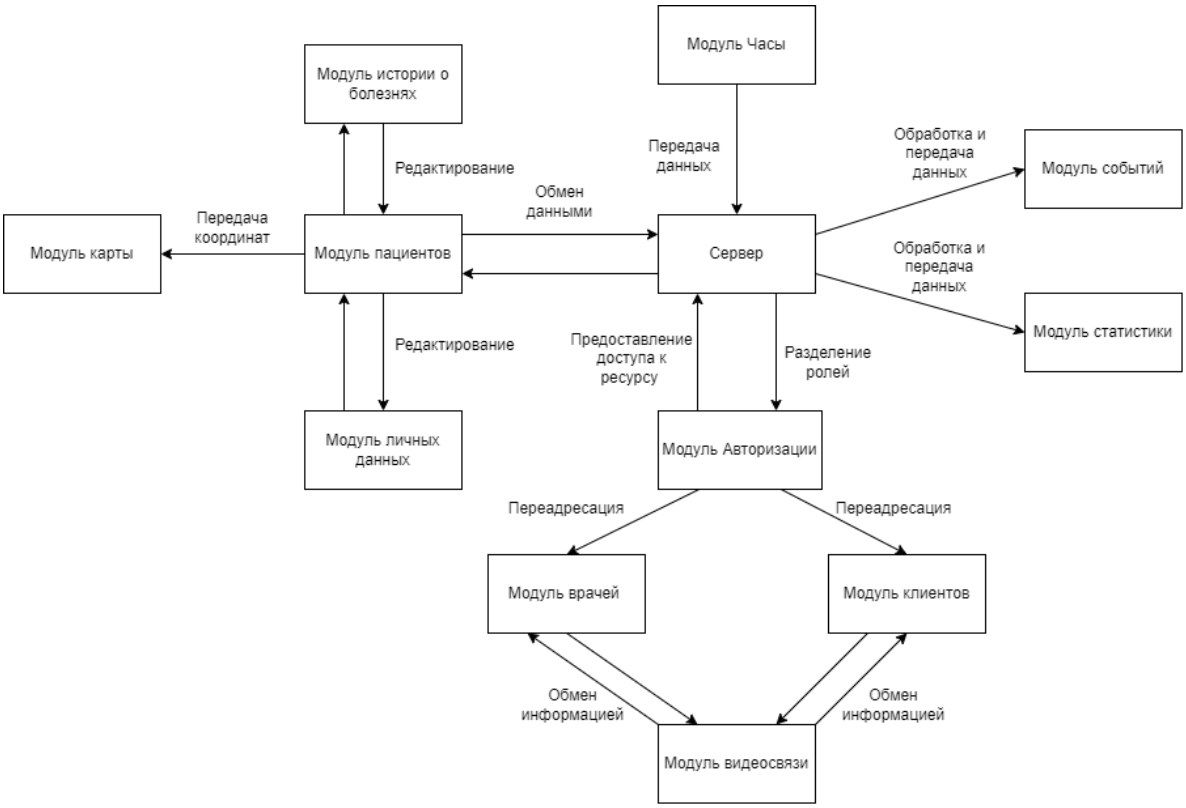


Рис. 1. Модульная схема ИС дистанционного мониторинга пациента

Также для всех участников реализован вход в систему при помощи JSON Web Token, а для того, чтобы не заходить в систему несколько раз в течение времени, добавлен Refresh Token, который обращается на сервер и получает новый JSON Web Token для пользователя. Данная система является безопасной и легкой в разработке.

Диаграмма прецедентов, отражающая основные функциональные возможности системы, представлена на рис. 2 (см. ниже).

Преимуществами подхода, при котором врачи делятся на различные уровни являются: улучшенная безопасность личных данных пациента, улучшение оказания услуг с уменьшением количества пациентов у одного врача, а также уменьшение рисков ошибки, поскольку интерну можно назначить отдельных пациентов, тем самым выбрать для врача наиболее подходящих пациентов по его уровню знаний.

Процесс загрузки данных со смарт-часов имеет следующий алгоритм. В первую очередь часы собирают данные о пациенте: пульс, температуру, сожжённые калории, ночные подъемы, заряд, сигнал, местоположение и другое. После этого указанные данные отправляются на сервер, где система их обрабатывает. Происходит сохранение данных: пульс, нажатая

кнопка о помощи, заряд часов и сигнал в таблицу «Часы». Затем сохраняются данные о местоположении пациента в соответствующую таблицу. Далее проверяется статус кнопки SOS: если она нажата, у врача появляется оповещение в крайнем верхнем углу экрана. Аналогичное оповещение приходит врачу в случае, если пациент упал или у него слишком высокий или низкий пульс в зависимости от болезней.

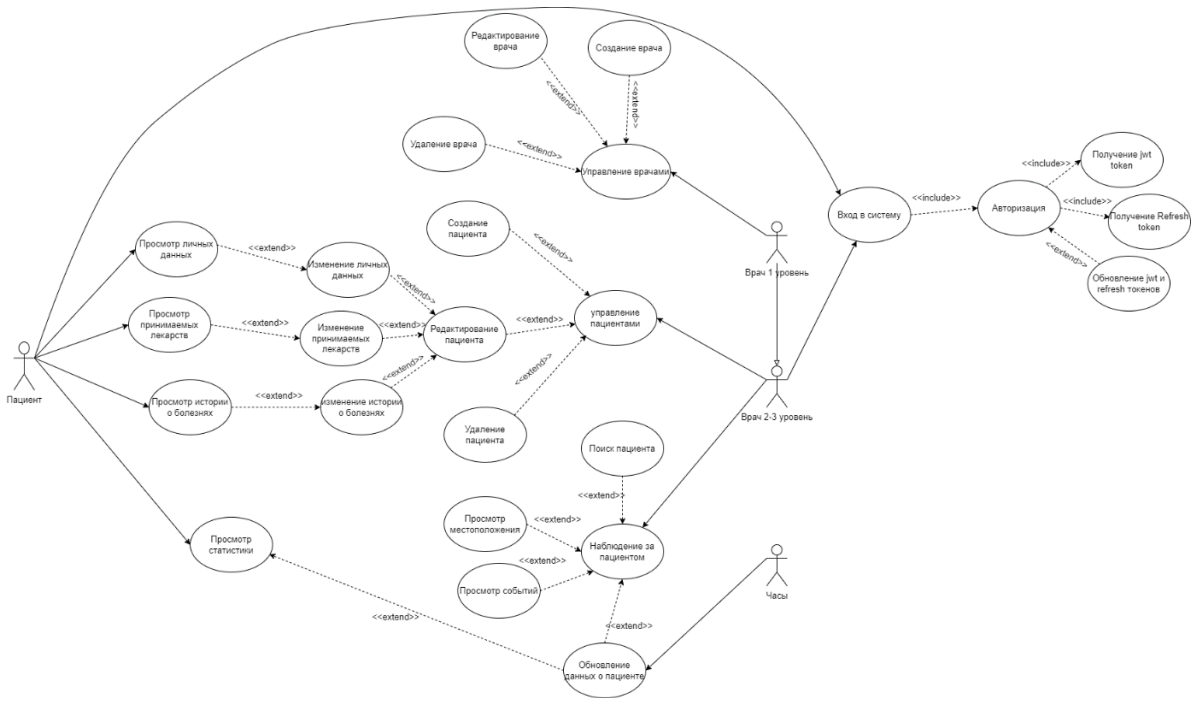


Рис. 2. Диаграмма прецедентов ИС дистанционного мониторинга пациента

Диаграмма классов является одним из важных инструментов в объектно-ориентированном программировании и используется для моделирования классов, их свойств, методов и взаимодействия между ними. Общая диаграмма классов, отображающая взаимодействие всех классов системы, представлена на рис. 3 (см. нже).

Детальнее рассмотреть состояния экземпляра сущности «Пациент» можно на диаграмме состояний (рис. 4, см. ниже).

Таким образом, предложен проект информационной системы автоматизированного дистанционного мониторинга здоровья пациента, способствующей совершенствованию медицинского учета населения. С помощью такой системы можно существенно улучшить качество медицинского обслуживания и снизить затраты на лечение, повысить уровень контроля за здоровьем пациента и увеличить вероятность своевременного выявления проблемных моментов.

В перспективах развития представленной работы планируется внедрение более расширенного функционала для удобства пациентов, а также интеграция модуля автоматического анализа отклонений от стандартных показателей пациента с применением искусственного интеллекта.

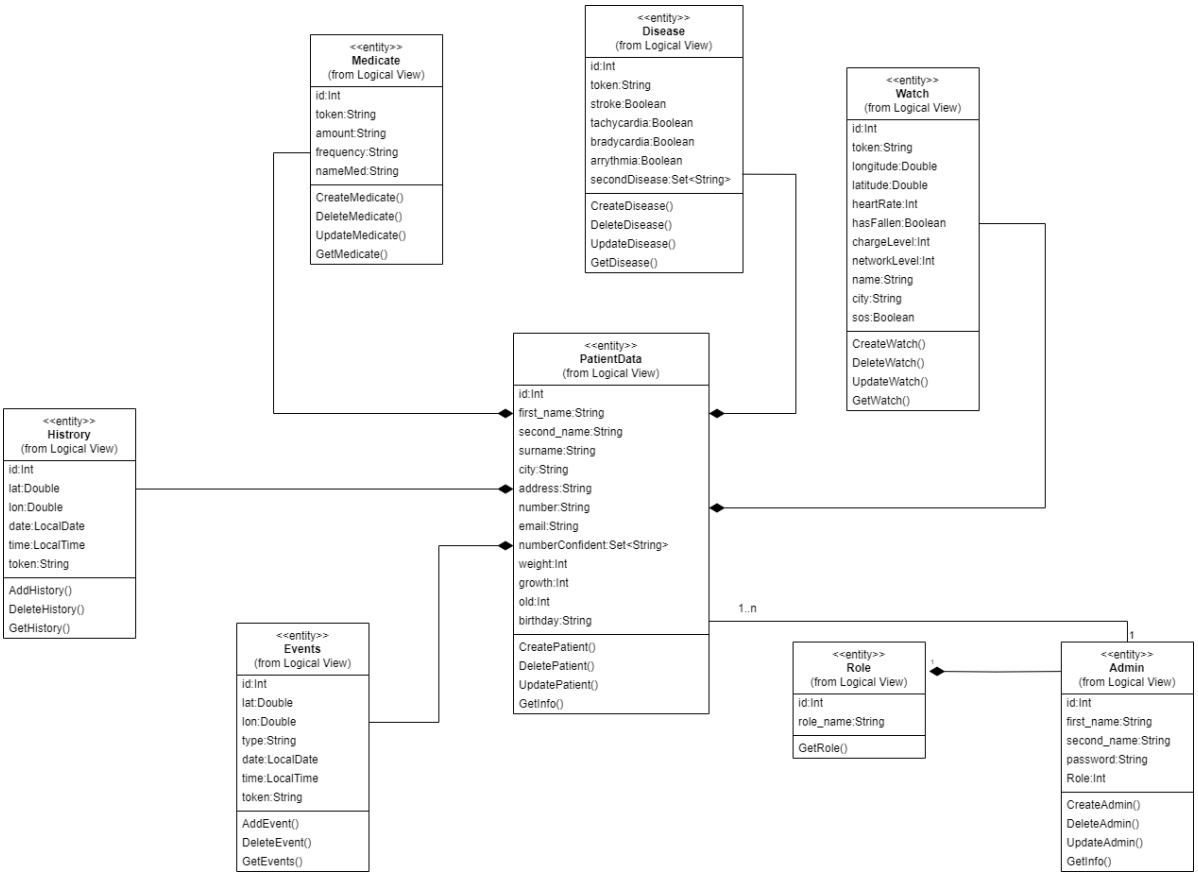


Рис. 3. Диаграмма классов ИС дистанционного мониторинга пациента

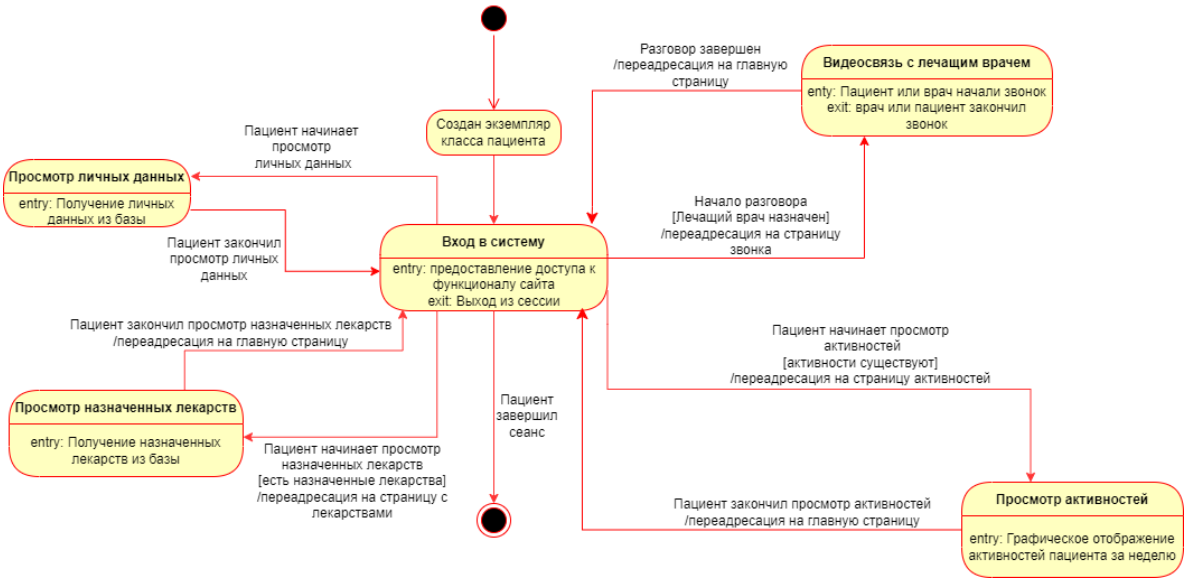


Рис. 4. Диаграмма состояний для объекта «Пациент»

Список используемых источников

1. Воробьев П. А., Воробьев А. П., Воробьев М. П., Краснова Л. С., Копошилова О. С., Везикова Н. Н., Барышева О. Ю. Электронное здравоохранение: проект MeDiCase для помощи пожилым жителям удаленных поселков Карелии // Проблемы стандартизации здравоохранения. 2017. Ч. 2. № 9–10.

2. Владзимирский А. Телемедицина. Как использовать цифровой прогресс для здоровья, когда все на удаленке. СПб. : Питер, 2019. 300 с.

Статья представлена доктором технических наук, профессором Птицыной Л. К.

УДК 004.2

А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)
М. Д. Сагиданов (студент гр. ИСТ-022, СПбГУТ)

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПАНЕЛИ УПРАВЛЕНИЯ IPTV

Рассмотрены различные аспекты проектирования информационной системы панели управления Internet Protocol Television. Представлена актуальность разработки программной платформы для развёртывания системы панели управления. Разработана архитектура информационной системы, учитывающая технические и функциональные аспекты использования системы оператора услуг. Информационная система призвана усовершенствовать организацию процессов управления и наблюдения в сфере представления услуг интернет-телевидения. Рассматриваемые в статье бизнес-процессы формализованы с помощью графического языка объектного моделирования Unified Modeling Language.

проектирование, информационная система, панель управления IPTV, администрирование, UML-диаграмма.

Актуальность разработки информационной системы панели управление IPTV основывается на нескольких современных тенденциях. В первую очередь, это увеличение количества телевизоров и других устройств с возможностью выхода в интернет со стабильным соединением и достаточной скоростью. В это же время, люди привыкают к тому, что имеют возможность получить доступ к тому видеоконтенту, который хотят. Не стоит забывать и о тех, кто эмигрировал в другую страну, но всё ещё нуждается в просмотре телеканалов на родном языке. Технология IPTV позволяет достичь передачи видеоинформации по протоколу TCP/IP и удовлетворить потребности людей при выборе видеоконтента, в том числе в просмотре каналов телевидения любой страны [1].

Соответственно, рост популярности IPTV приводит к появлению на рынке новых поставщиков данных услуг. Им, в свою очередь, необходима информационная система, которая позволит надёжно и удобно управлять комплексом IPTV. Главный недостаток существующих систем – отсутствие возможности встраивания панели управления отдельно от всей

экосистемы, что может не соответствовать требованиям некоторых операторов.

Комплекс IPTV состоит из нескольких крупных модулей: сервер с трансляциями каналов, базы данных с каналами, категориями, тарифами и клиентами [2]. Редактирование и настройка каждого модуля по-отдельности займёт много времени, а это приведёт к финансовым потерям, чего необходимо избегать любому поставщику услуг. Следовательно, требуется система с единственной точкой входа в комплекс, чтобы затем получить возможность настраивать всю систему с помощью одного интерфейса. Эта же система обязана обладать интуитивно понятной визуальной оболочкой, чтобы быть эффективным средством управления в руках администратора. Эти задачи ставит перед собой панель управления IPTV. Другими словами, это инструмент для быстрой и надёжной работы со сложным комплексом IPTV для людей, которые имеют низкий уровень технической подготовки.

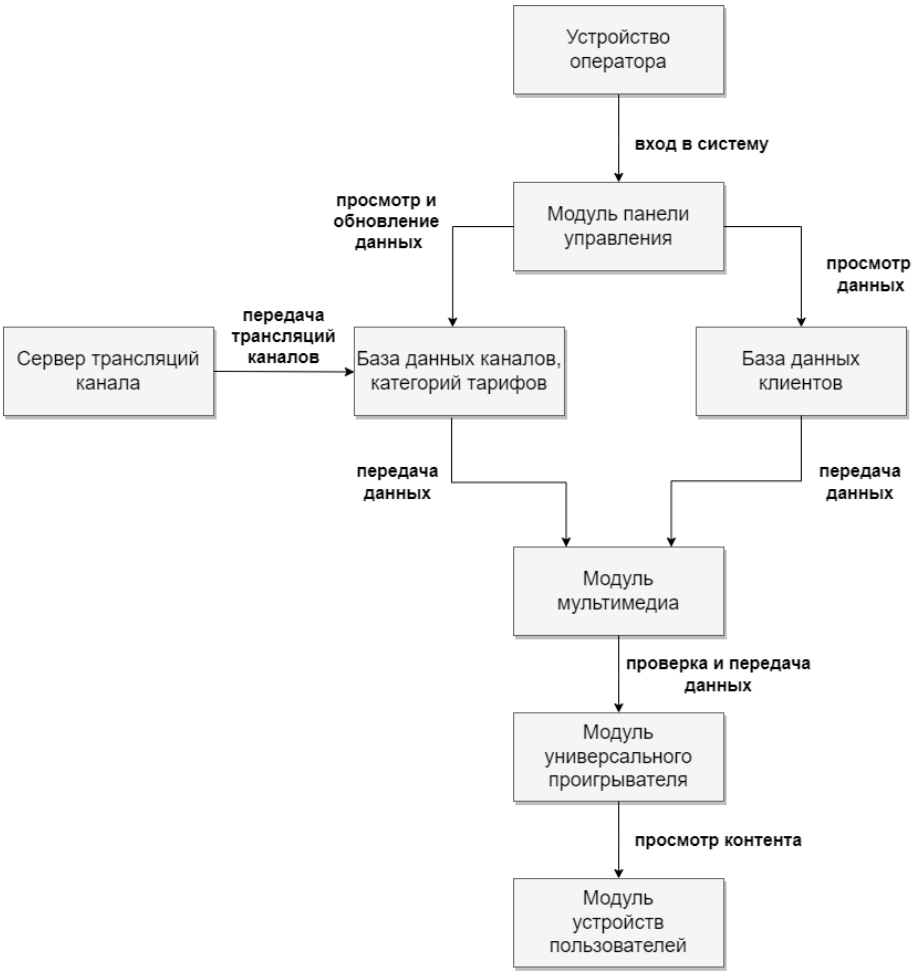


Рис. 1. Схема взаимодействия модулей системы панели управления IPTV

Для наглядного представления архитектуры системы и места, которое в ней занимает панель управления, разработана модульная схема информационной системы. Каждый модуль, представленный на рис. 1, выполняет

соответствующую задачу, после выполнения которой передаёт необходимую информацию следующему модулю. Панель управления имеет доступ к просмотру и настраиванию двух важных модулей, содержащих в себе каналы, категории, тарифы и информацию о клиентах. Внесённые изменения далее применяются во всех остальных модулях. Например, администратор изменил тариф клиенту, после сохранения клиенту будет доступен другой набор категорий.

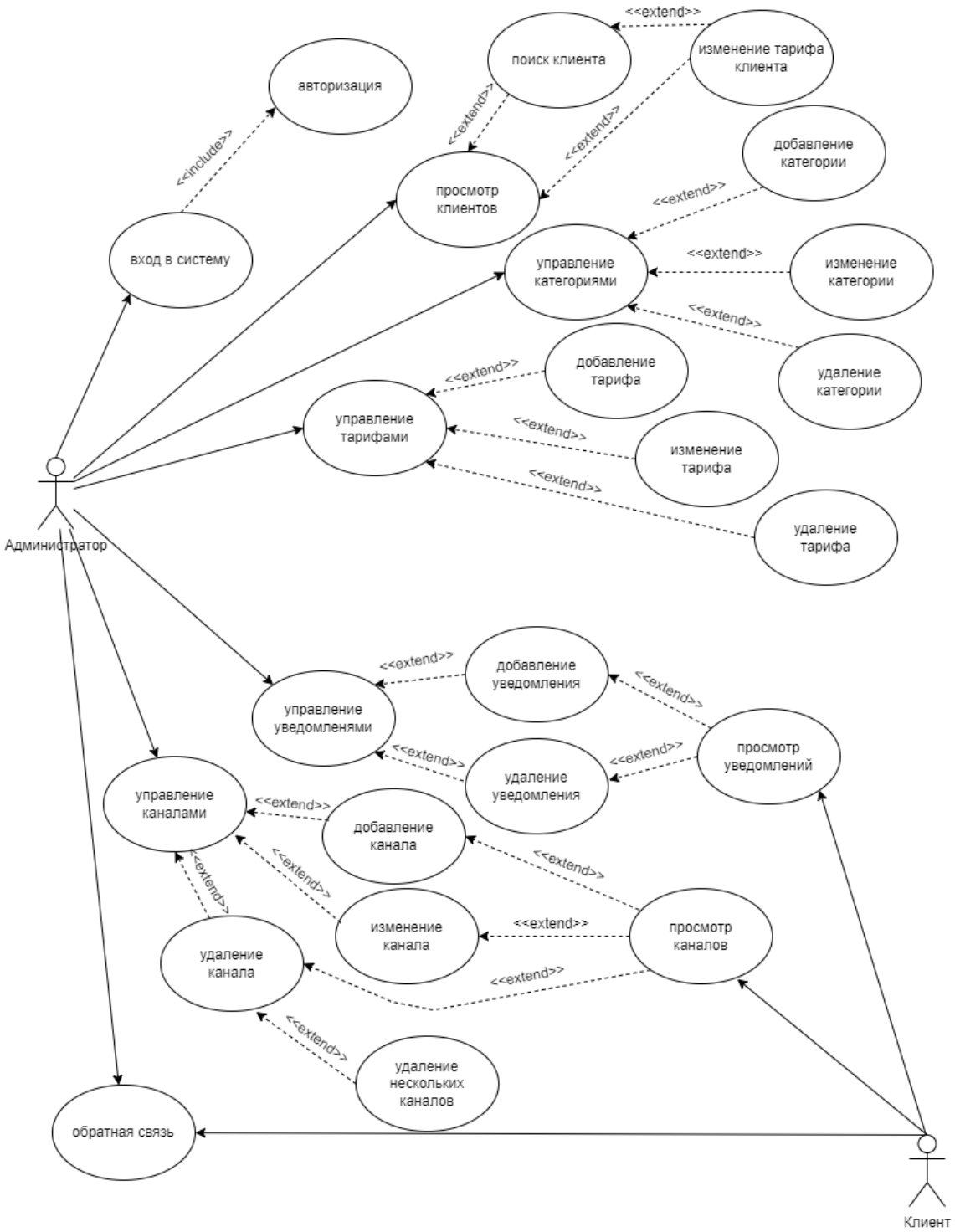


Рис. 2. Диаграмма вариантов использования панели управления IPTV

Для описания полного функционала системы разработана диаграмма прецедентов, представленная на рис. 2. В ней описаны возможности использования панели управления для администратора, а также определяется, какое место в системе занимается клиент. Среди возможностей имеется полный функционал контроля над системой: просмотр, создание, изменение, удаление каналов, категорий, тарифов, уведомлений, а также просмотр информации о клиентах и управление их тарифами. Другими словами, с помощью панели управления можно вносить изменения во все ключевые модули системы. Это эффективно в случае необходимости оперативного внесения изменений, например, создание нового тарифа для акции от руководства или отключение канала, на котором наблюдаются проблемы с трансляцией.Детальный процесс изменения тарифа клиента представлен в виде диаграммы последовательности на рис. 3.

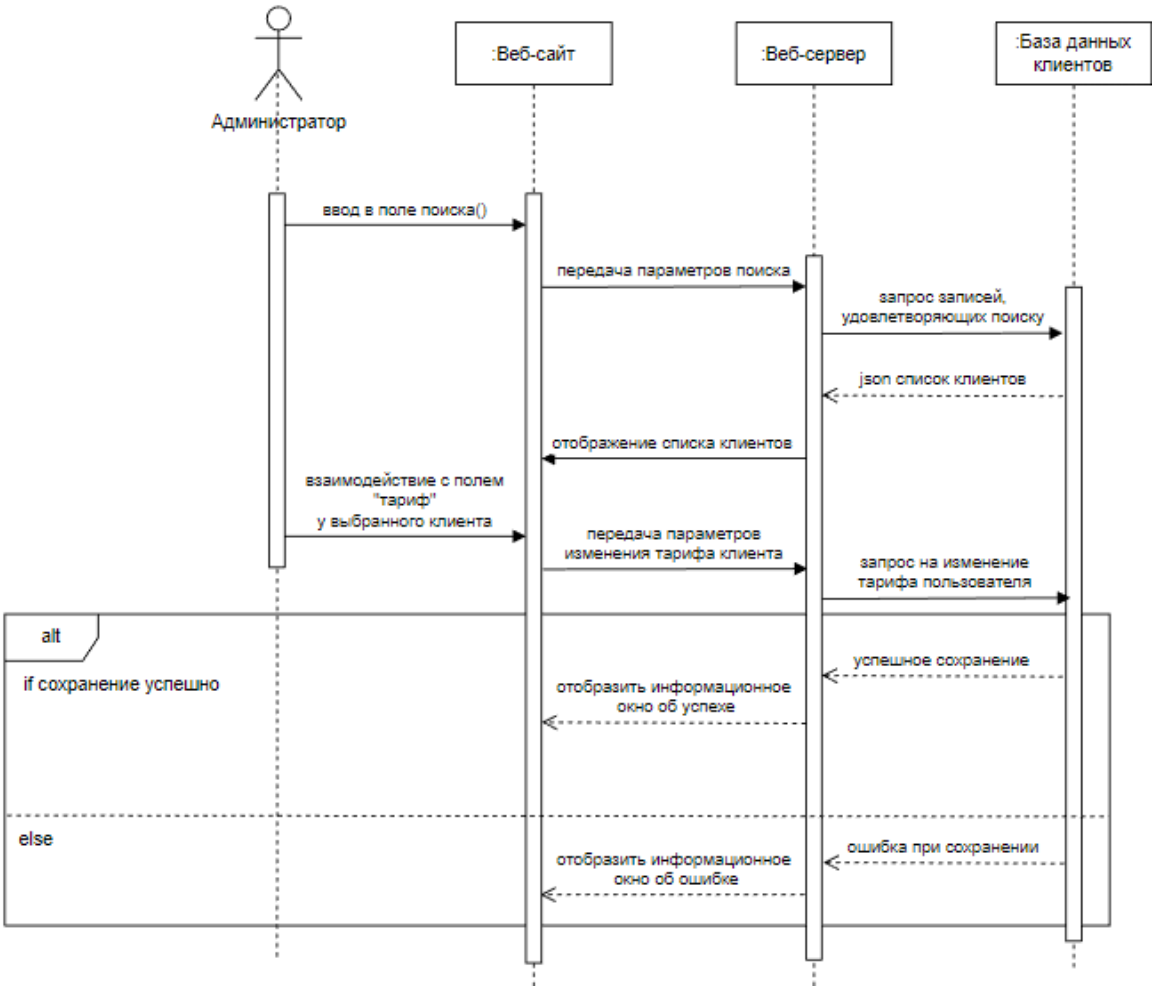


Рис. 3. Диаграмма последовательности прецедента «Изменение тарифа клиента»

Макет страницы со списком тарифов представлен на рис. 4. Данная страница позволяет просмотреть информацию, добавить новый тариф или изменить существующие.

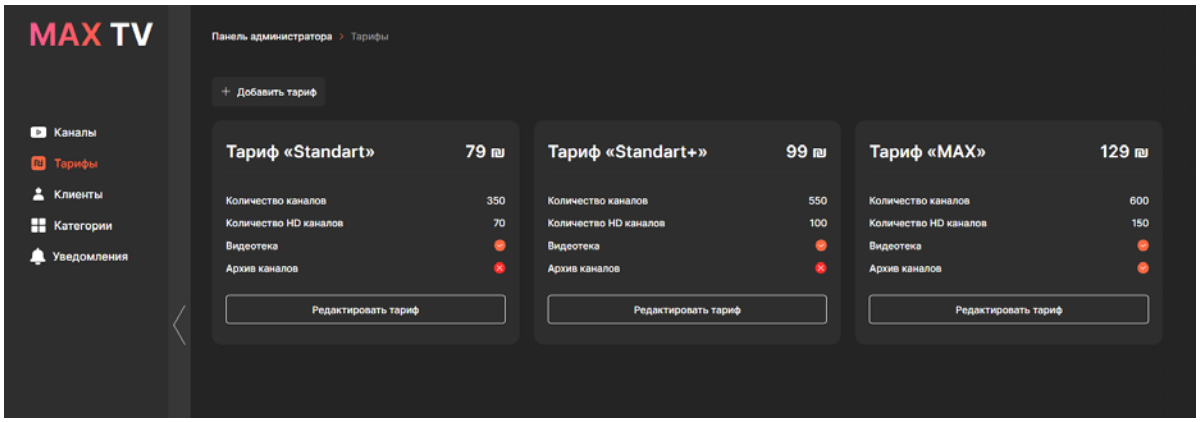


Рис. 4. Макет страницы со списком тарифов

В результате проделанной работы спроектирована информационная система панели управления IPTV, показано, как она внедрена в бизнес-процессы компании оператора, какое место панель управления занимает в архитектуре комплекса IPTV, продемонстрированы варианты использования системы и ее функциональные особенности.

Данная панель позволяет оптимизировать процесс управления контентом и услугами, что может привести к улучшению качества обслуживания пользователей и снижению затрат содержания комплекса.

Список используемых источников

1. Дитер Ш., Головачев Ю. Кирхгесснер Ю. IPTV-эра в телевидении // T-Comm – Телекоммуникации и Транспорт. 2008. № 6. С. 17–20.

2. Понедельникова О. А. Основные компоненты сети IPTV // Вологдинские чтения. 2008. С. 73–74.

Статья представлена доктором технических наук, профессором Птицыной Л. К.

УДК 004.855.5

Ю. Е. Жарова, Д. В. Самолетова, А. П. Федоров
(студенты гр. ИКПИ-13, СПбГУТ)

МАШИННОЕ ОБУЧЕНИЕ. СРАВНЕНИЕ ДВУХ ОСНОВНЫХ ПОДХОДОВ К ОБУЧЕНИЮ НЕЙРОННЫХ СЕТЕЙ

На сегодняшний день системы с искусственным интеллектом применяются во многих аспектах человеческой жизни. В статье рассматриваются методы машинного обучения с учителем и без него. Цель исследования - обеспечить более глубокое понимание того, как функционируют системы искусственного интеллекта, и потенциальных последствий их использования в различных контекстах. С помощью модели

нейронной сети был проведен сравнительный анализ данных методов и выявлены достоинства и недостатки каждого из них.

машинное обучение, искусственный интеллект, нейронная сеть, обучение.

Машинное обучение (*Machine Learning*) – это метод анализа данных, который автоматизирует построение аналитической модели. Это отрасль искусственного интеллекта, основанная на идее, что машины должны уметь учиться и адаптироваться через опыт. Оно тесно связано с вычислительной статистикой, которая делает прогнозы на основе статистических данных, собранных компьютером. Можно сказать, что это способ обучать компьютеры без программирования и явных инструкций, используя только шаблоны и логические выводы.

Существуют три составляющие любого процесса машинного обучения:

1. Данные. Это примеры решений, статистика, расчёты и другая информация, которая помогает обучать нейросети. Большие наборы данных называют датасетами.

2. Признаки. Это те вещи, на которые модель должна обратить внимание во время обучения. Чем конкретнее признаки и меньше их количество, тем быстрее проходит обучение. У сложных моделей могут использоваться миллионы параметров.

3. Алгоритмы. Это способы решения задач. У задачи может быть несколько вариантов решения, а машина должна выбрать лучший из них.

Основные виды машинного обучения:

1. Классическое обучение.
2. Обучение с подкреплением.
3. Ансамбли.
4. Нейросети и глубокое обучение.

Алгоритмы машинного обучения классифицируются на обучение с учителем и без учителя.

Для обучения с учителем требуется, чтобы специалист по данным обладал навыками машинного обучения, чтобы обеспечить как ввод, так и желаемый результат, в дополнение к предоставлению обратной связи о точности предсказаний во время обучения алгоритму. Исследователи данных определяют, какие переменные или характеристики модель должна анализировать и использовать для разработки прогнозов. После завершения обучения алгоритм будет применять полученные данные к новым данным. Примерами методов обучения с учителем являются классификация и регрессия.

Методы обучения без учителя не требуют готовых ответов и используются для поиска зависимостей между объектами. К таким методам относятся: кластеризация снижение размерности и нейронные сети.

Нейронные сети используются для более сложных задач обработки, чем контролируемые системы обучения, включая распознавание изображений, преобразование речи в текст и создание естественного языка. Эти

нейронные сети работают, комбинируя миллионы примеров обучающих данных и автоматически выявляя тонкие корреляции между многими переменными. После обучения алгоритм может использовать свой банк ассоциаций для интерпретации новых данных. Эти алгоритмы стали возможными только в эпоху больших данных, поскольку они требуют огромных объемов обучающих данных.

Практическая часть

Цель данной работы – исследование двух подходов обучения нейронных сетей и выявления их сильных и слабых сторон, а также их отличительных особенностей.

Для сравнения двух подходов к обучению нейронной сети, была создана модель нейронной сети, которая обучалась на датасете MNIST. База данных образцов рукописного написания цифр MNIST содержит 60 000 образцов наборов данных для обучения и тестовый набор из 10 000 образцов. Цифры нормализованы по размеру и расположены в центре изображения фиксированного размера. Для лучшего представления этой базы данных приведена ее небольшая часть на рис. 1.

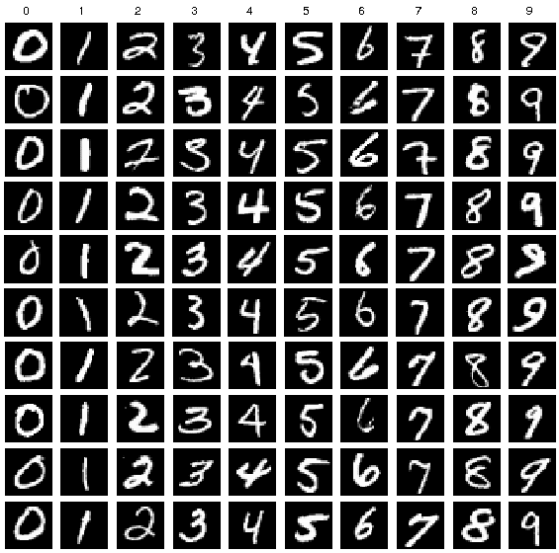


Рис. 1. База данных MNIST

В первом случае модель была обучена без учителя. Обучение без учителя проводилось при помощи автоэнкодера. На вход подавались изображения цифр, а на выходе сеть пыталась восстановить исходное изображение. Это происходило за счет самостоятельной обработки базы данных в течении 30 эпох. На графике (рис. 2) представлена зависимость продолжительности обучения от точности вычислений.

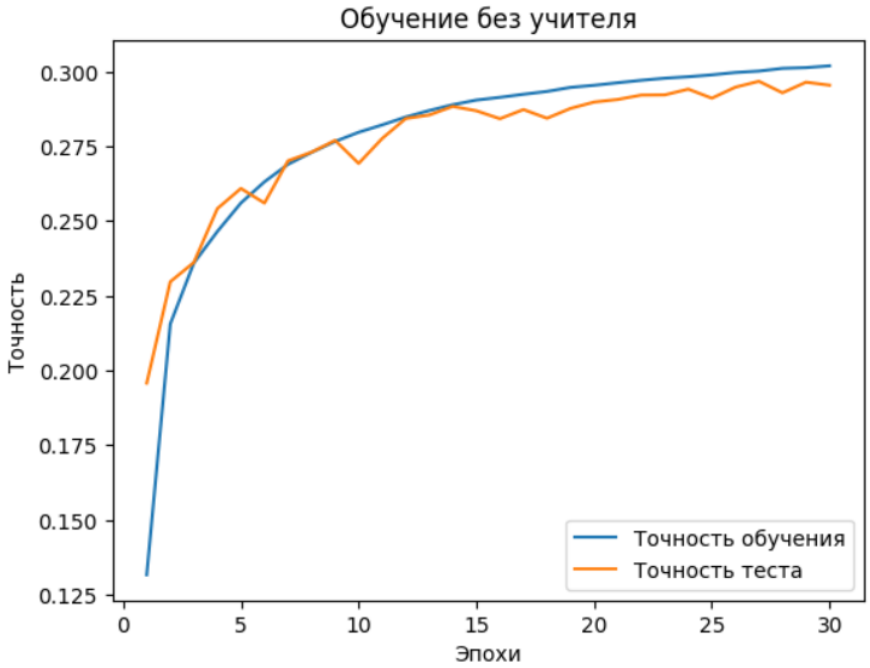


Рис. 2. Обучение без учителя

Во втором случае нейронная сеть была обучена с учителем. Для этого исходная база данных была разделена на тестовую и обучающую выборки. Обучение происходило в течение 30 эпох. Результаты представлены на графике (рис. 3).

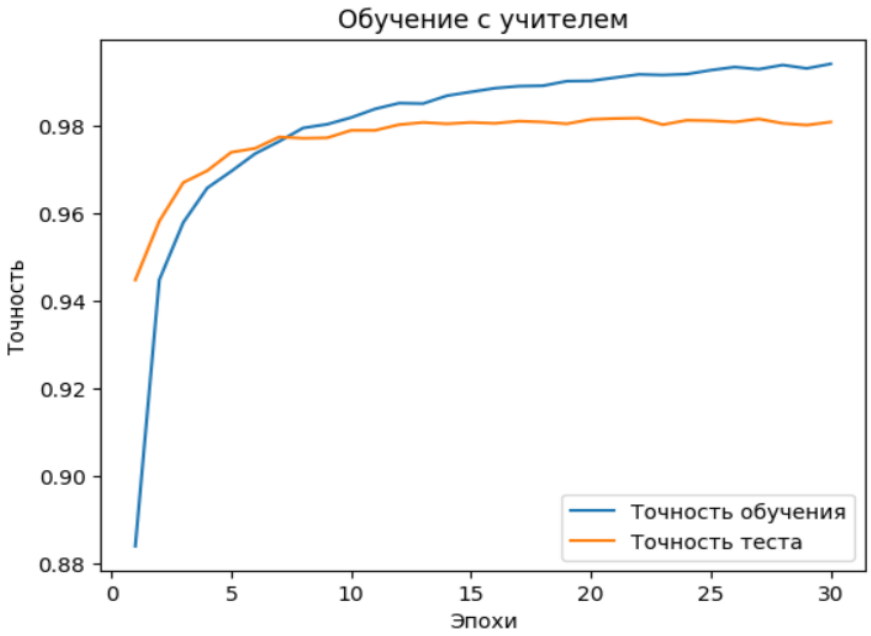


Рис. 3. Обучение с учителем

Сравнив данные графики, можно сделать вывод, что метод обучения с учителем показал более высокую точность как на этапе обучения, так и на этапе тестирования. Такой хороший результат связан с наличием явной классификации данных.

Метод обучения без учителя, в свою очередь, может быть полезен в случаях, когда нет явной классификации, и его применение может позволить выявить скрытые закономерности в данных. Однако в данном случае точность обучения без учителя оказалась ниже, чем при методе с учителем.

Таким образом, выбор метода обучения нейронных сетей зависит от целей и требований к точности модели, а также от особенностей данных, на которых происходит обучение.

Список используемых источников

1. Уоссермен Ф. Нейрокомпьютерная техника. Теория и практика. 1992. 184 с.
2. Гафаров Ф. М., Галимянов А. Ф. Искусственные нейронные сети и их приложения : учебное пособие. Казань : Изд-во Казан. ун-та, 2018 г. 121 с.
3. Обучение нейросети с учителем, без учителя, с подкреплением // neurohive.io. URL: https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/ (дата обращения 06.06.2022).
4. База данных MNIST с рукописными цифрами // https://learn.microsoft.com/ru-ru/azure/open-datasets/dataset-mnist?tabs=azureml-opendatasets (дата обращения 06.06.2022).

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.8

М. А. Изюмова (студентка гр. ИСТ-021, СПбГУТ)
А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)

ПРОЕКТИРОВАНИЕ ВЕБ-РАСШИРЕНИЯ ПО ФОРМИРОВАНИЮ ПЕРСОНАЛЬНОЙ ПОДБОРКИ КНИГ

Актуализирован вопрос проектирования и развития веб-расширения по подбору литературы, соответствующей личным предпочтениям и интересам. Предлагается использование алгоритма рекомендаций с использованием искусственного интеллекта для оптимизации процесса составления персональной подборки книг для пользователя. Сформирована и продемонстрирована модульная схема веб-расширения по формированию персональной подборки книг. Построена диаграмма прецедентов, отражающая основные функциональные группы информационной системы. Представлен алгоритм формирования персональной подборки книг. Построена диаграмма классов информационной системы. Описаны перспективы развития веб-расширения.

алгоритм рекомендаций, информационная система, персонализированная подборка книг, прототип веб-расширения, UML-диаграмма.

В связи с развитием технологий в области искусственного интеллекта, в частности с улучшением алгоритмов искусственного интеллекта, многие повседневные процессы приобретают более персонализированный характер, стараясь таким образом оптимально и быстро решить стоящие перед ними задачи. Текущая тенденция оказала влияние в том числе на процесс чтения.

Создание удобного программного продукта для ежедневного использования может значительно облегчить и улучшить опыт чтения книг как для опытных читателей, так и для новичков, которые только формируют свой читательский вкус.

Актуальность проектирования и разработки веб-расширения по формированию персональной подборки книг определяется необходимостью использования персонального подхода к составлению подборок книг, которые соответствуют личным предпочтениям и интересам отдельного читателя. В настоящее время рекомендательные системы делятся на две большие группы: первая группа – системы, основанные на содержании, в случае литературной рекомендательной системы читателю, который последовательно выбирал книги определенного жанра, следующее произведение будет предложено такого же жанра [1]; вторая группа рекомендательных систем основана на коллаборативной фильтрации, главный принцип подобных систем заключается в том, что похожие объекты интересны похожим пользователям, алгоритмы строятся на основе мер близости между пользователями и/или предметами (книгами) [2]. Успех внедрения технологии коллаборативной фильтрации при составлении персональных рекомендации можно увидеть на примере следующих зарубежных и отечественных приложений и платформ: Google, Amazon, Netflix, Spotify, YouTube, HH.ru, TikTok. Помимо этого, реализация прототипа веб-расширения несет в себе просветительский характер, поскольку может привлечь новых читателей, так как обеспечивает персонализированную подборку книг на основе предпочтений конкретного пользователя. Данная технология помогает читателям находить новые произведения, тем самым расширяя кругозор в литературе.

В рамках выбранной предметной области существует ограниченное количество конкурентоспособных веб-расширений с возможностью персонализированной подборки книг, однако можно выделить веб-расширения с аналогичным функционалом.

Наиболее приближенным по функционалу является веб-расширение «Goodreads Right Click», которое позволяет осуществлять поиск по заголовку, автору или коду ISBN с выдачей результата в поисковой системе книжной платформы Goodreads. Недостатки: явное дублирование функций «Goodreads Now» и «Goodreads Later», при использовании данных опций

не выявлено различий, они выполняют одну и ту же задачу. Данный недостаток может быть связан с недоработкой пользовательской инструкции, где пользователя не уведомляют о возможной проблеме и путях ее устранения.

Другими конкурентными сервисами являются веб-расширение «SwiftRead», которое позволяет читать книги только в определенном формате, и веб-расширение «Library Extension», ориентированное на аудиторию других стран.

Веб-расширение по формированию персональной подборки книг имеет следующие ограничения: привязка к определенному книжному сайту, возможность использования только при наличии личного кабинета пользователя на конечном книжном сайте и заполненных данных в разделах «Прочитать» в количестве не менее 3 книг. Данные ограничения связаны с тем, что веб-расширение подключается к книжному сайту через специальный интернет-протокол и выгружает персональные данные пользователя из личного кабинета, который зарегистрирован на конечном книжном сайте. Таким образом, информационная система реализует сбор данных, выдачу персональной подборки, отправку данных на конечный книжный сайт при взаимодействии с интерфейсом системы, а также содержит облачную базу данных для хранения результатов алгоритма рекомендаций с привязкой к определенному профилю.

На рис. 1 представлена модульная схема веб-расширения по формированию персональной подборки книг. В схеме заложена возможность формирования персональной подборки книг для каждого пользователя с возможностью дообучения алгоритма. При получении подборки пользователь имеет доступ к ее модификации посредством взаимодействия с интерфейсом веб-расширения путем нажатия активных кнопок «Прочитать» или «Не рекомендовать».

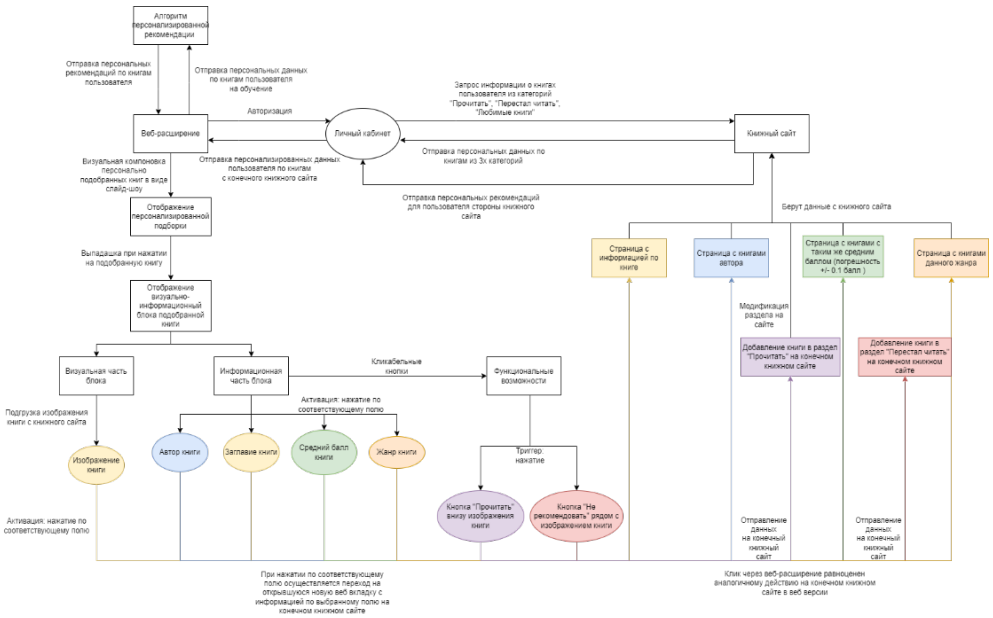


Рис. 1. Модульная схема веб-расширения по формированию подборки книг

Следует подробно рассмотреть аспект взаимодействия пользователя с интерфейсом веб-расширения. Варианты использования веб-расширения пользователем представлены на рис. 2.

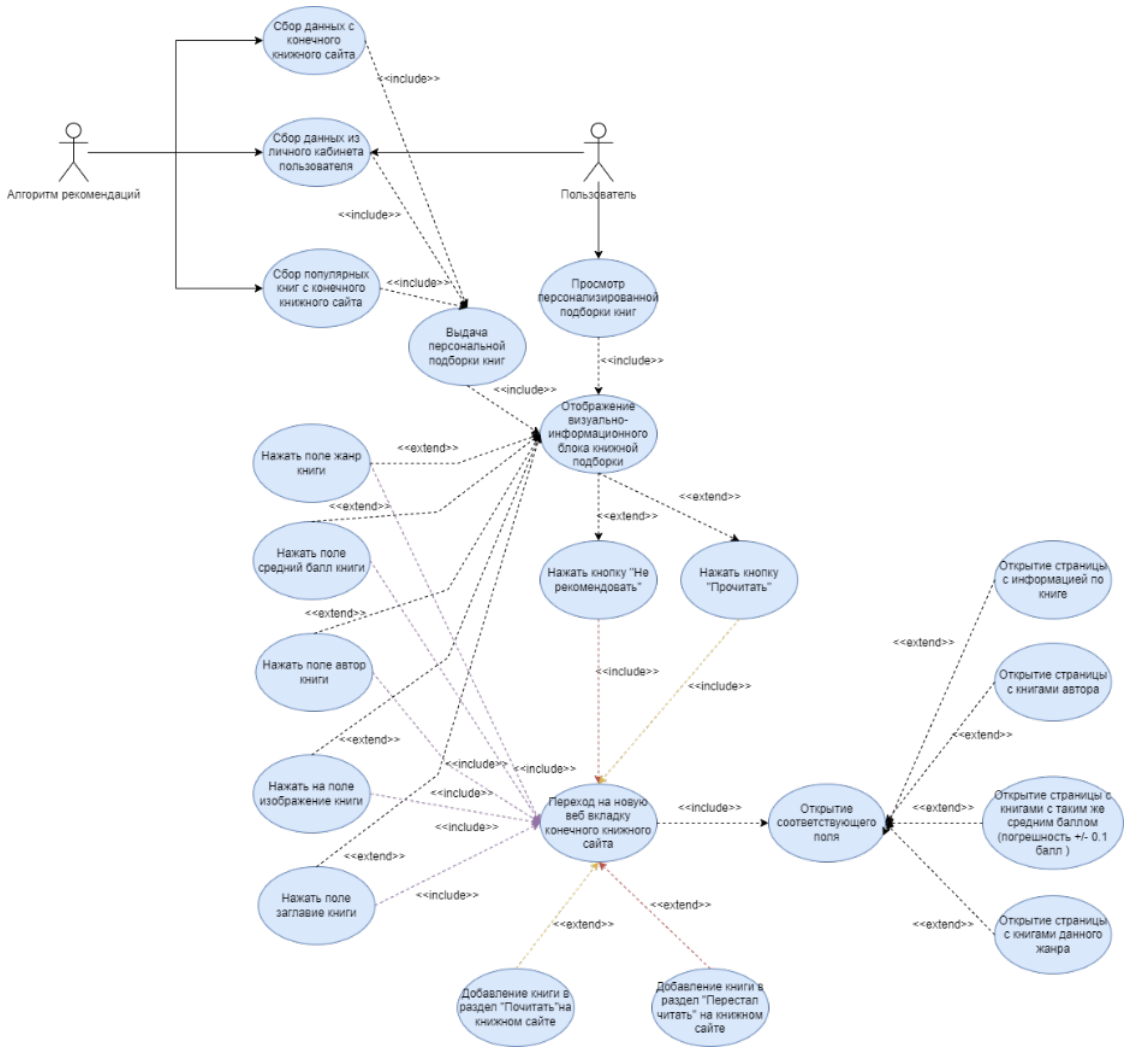


Рис. 2. Диаграмма прецедентов веб-расширения по формированию подборки книг

При нажатии кнопки «Прочитать» веб-расширение отправляет данные книги, под которой была активирована данная кнопка, на сторону алгоритма рекомендаций и книжного сайта. Алгоритм рекомендаций при получении данных разбивает их на компоненты: «автор», «жанр», «средний балл» и обучается на полученных данных. После обучения алгоритм рекомендаций запоминает текущие предпочтения пользователя и при дальнейшем составлении персональной подборки учитывает предыдущий опыт обучения. Книжный сайт после активации кнопки «Прочитать» получает данные от веб-расширения и открывает новую веб-вкладку с информацией по текущей книге, перенаправляя на данную страницу пользователя, а также добавляет произведение в раздел «Прочитать» в личном кабинете пользователя на конечном книжном сайте. Алгоритм формирования персональной подборки книг представлен в виде диаграммы деятельности на рис. 3.

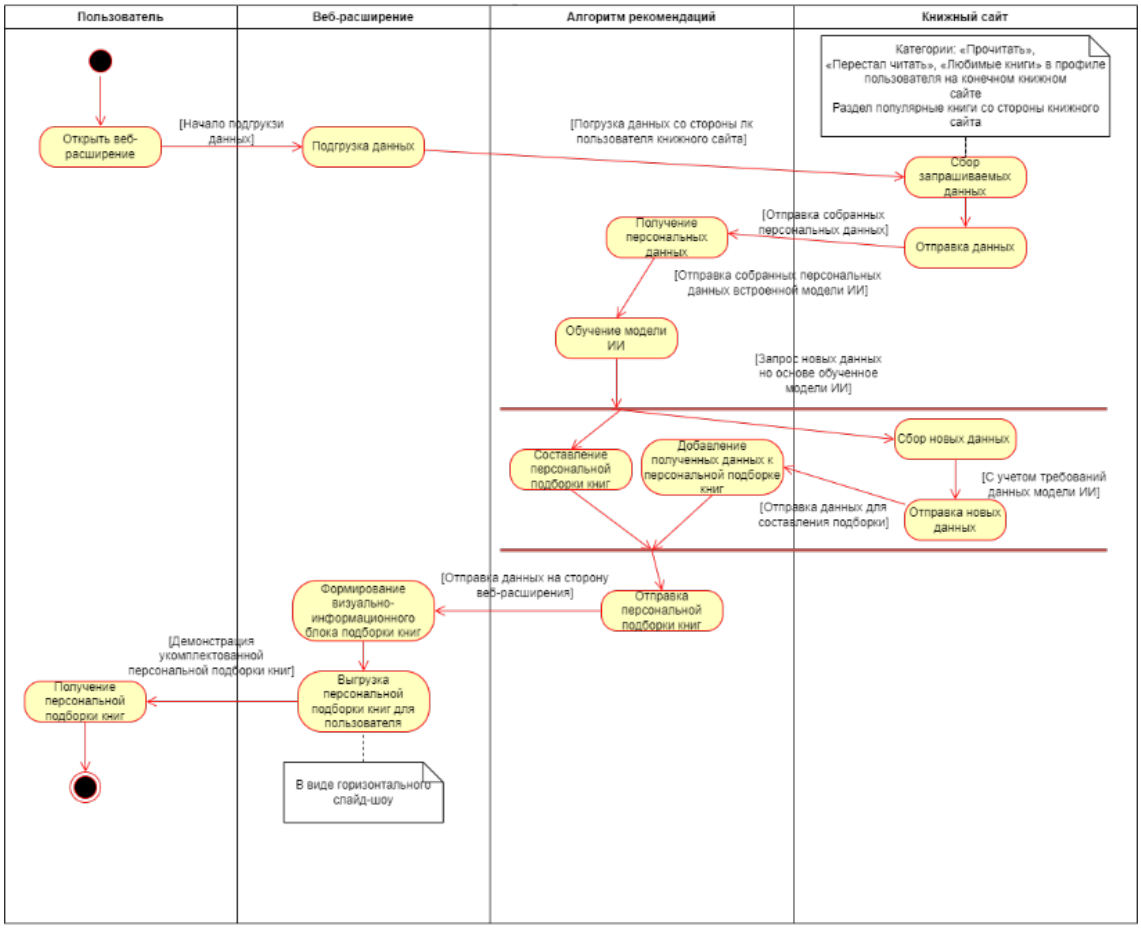


Рис. 3. Диаграмма деятельности процесса формирования подборки книг

Диаграмма классов для рассматриваемой информационной системы представлена на рис. 4.

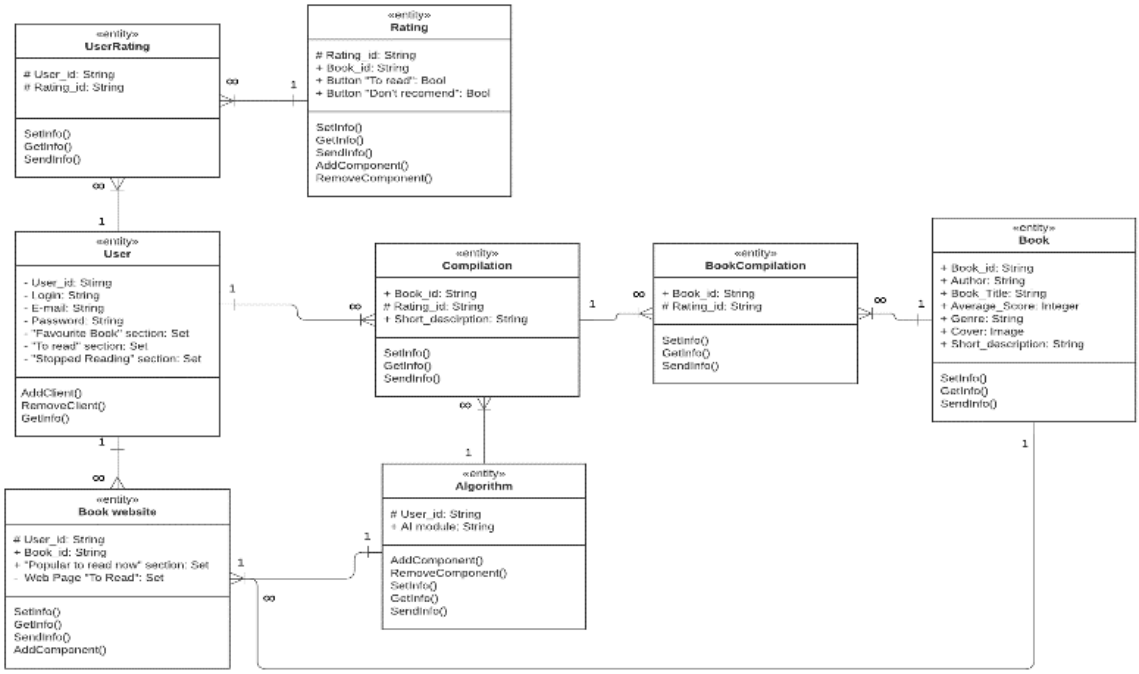


Рис. 4. Диаграмма классов веб-расширения по формированию подборки книг

Концептуальная архитектура информационной системы определяется композицией следующих систем-сущностей, их модулями и составными частями:

– система личного кабинета;
– система книжного сайта;
– модуль алгоритма рекомендаций;
– модуль рейтинг;
– модуль персональной подборки книг.

Перспективы развития данной информационной системы заключаются в нескольких аспектах: расширение системы для интеграции с другими книжными сайтами, библиотеками или сервисами для расширения доступных данных; расширение и улучшение алгоритмов рекомендаций с включением гибридных подходов для точного определения предпочтений пользователей.

В результате проделанной работы формализованы алгоритмы, характерные для различных аспектов работы проектируемой информационной системы, учтены особенности взаимодействия двух систем – веб-расширения и книжного сайта – посредством предоставления модульной схемы и диаграммы классов. Предложен концепт информационной системы, спроектирована его архитектура и учтены функциональные особенности.

Данная информационная система характеризуется персональным подходом для предоставления книг пользователям и вовлечения новых читателей за счет удобного и понятного интерфейса и функциональных возможностей, отсутствующих у конкурентноспособных информационных систем.

Список используемых источников

1. Бондарчук Д. В., Тимофеева Г. А. Применение машинного обучения для формирования персональных рекомендаций в сфере трудоустройства // Экономика и менеджмент систем управления. 2015. Т. 18. №. 4.2. С. 215–221.

2. Шакиров С. М. Литература в формате нарратива (анализ публикаций сервиса персональных рекомендаций Яндекс. Дзен) // Знак: проблемное поле медиаобразования. 2019. №. 1 (31). С. 129–140.

Статья представлена доктором технических наук, профессором Птицыной Л. К.

УДК 004.428.2

И. В. Ильин, Е. И. Перебейнос (студенты гр. ИКПИ-14, СПбГУТ)

ИССЛЕДОВАНИЕ СОВРЕМЕННОЙ АРХИТЕКТУРЫ КРОССПЛАТФОРМЕННЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА REACT NATIVE

В данной статье проводится исследование современной архитектуры кроссплатформенных мобильных приложений на базе фреймворка React Native. Основная цель работы заключается в разработке и анализе эффективных архитектурных подходов, которые позволяют создавать мобильные приложения, способные работать на разных платформах.

кроссплатформенные мобильные приложения, react native, архитектура.

Современные мобильные приложения становятся неотъемлемой частью нашей повседневной жизни. Они помогают нам в работе, развлечении, общении и многих других сферах. Однако, разработка мобильных приложений для разных платформ может быть сложной и трудоемкой задачей. В этом контексте кроссплатформенность становится все более важной и востребованной, поскольку позволяет разработчикам создавать приложения, которые работают на разных платформах с использованием общего кода.

React Native – это популярный фреймворк разработки кроссплатформенных мобильных приложений, который основан на языке JavaScript. Он позволяет разработчикам создавать приложения, используя один и тот же код, который затем компилируется в нативный код для каждой платформы. React Native предоставляет множество возможностей для создания интерактивных и эффективных мобильных приложений.

Целью данного исследования является разработка оригинальной архитектуры кроссплатформенного мобильного приложения на React Native, учитывающей особенности фреймворка и обеспечивающей высокую производительность и удобство использования. Также проводится сравнительный анализ производительности разработанной архитектуры с другими существующими подходами.

Особенности архитектуры кроссплатформенных мобильных приложений на React Native

React Native предоставляет разработчикам гибкую архитектуру для создания кроссплатформенных мобильных приложений. Важными особенностями архитектуры на React Native является:

Компонентный подход. React Native основан на компонентной архитектуре, которая позволяет разбивать приложение на множество независимых компонентов. Это упрощает разработку, тестирование и поддержку кода (рис. 1).

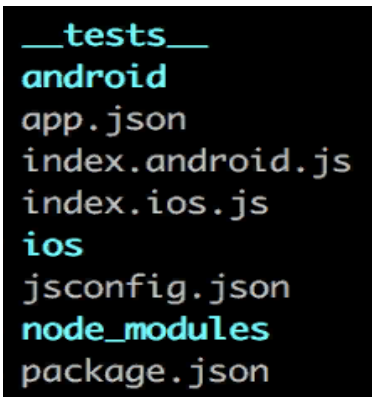


Рис. 1. Сгенерированная структура только что созданного проекта

Концепция работы с DOM (*Document Object Model*) отличается от традиционного веб-разработки. Вместо манипуляции с виртуальным DOM, React Native использует нативные компоненты, которые преобразуются в нативный код на каждой платформе. Они напрямую взаимодействуют с нативным API, обеспечивая высокую производительность и нативный пользовательский интерфейс. React Native использует «мост» (*bridge*) для передачи данных между JavaScript-кодом и нативной частью. Это позволяет создавать мобильные приложения с высокой производительностью, сохраняя при этом преимущества и простоту использования React.

Гибкость React Native позволяет разработчикам использовать существующий код JavaScript и библиотеки для создания кроссплатформенных приложений. Это упрощает переносимость и повторное использование кода.

Разработка оригинальной архитектуры кроссплатформенного мобильного приложения на React Native:

В ходе исследования была разработана оригинальная архитектура кроссплатформенного мобильного приложения на React Native. Архитектура состоит из следующих компонентов:

- Компонент навигации обеспечивает навигацию между различными экранами и модулями приложения (рис. 2).

```
1  const HomeTabs = TabNavigator(
2    {
3      Notification: {
4        screen: NotificationTabContainer,
5        path: 'notification',
6        navigationOptions: {
7          title: 'уведомление'
8        }
9      },
10     Myself: {
11       screen: MyselfTabContainer,
12       path: 'myself',
13       navigationOptions: {
14         title: 'моя'
15       }
16     }
17   },
18   {
19     tabBarOptions: {
20       activeTintColor: Platform.OS === 'ios' ? '#e91e63' : '#fff',
21     },
22     swipeEnabled: true
23   }
24 );
```

Рис. 2. Объекты конфигурации, связанные с навигацией

• Компонент управления состоянием хранит и управляет состоянием приложения, позволяет синхронизировать данные между различными компонентами (рис. 3).

```
1  class App extends Component {
2    render () {
3      return (
4        <Provider store={store}>
5          <AppContainer />
6        </Provider>
7      )
8    }
9  }
```

Рис. 3. Provider – компонент оборачивающий наше приложение и отвечающий за изменение состояние компонентов

• Компонент отображения данных отвечает за отображение данных и взаимодействие с пользователем (рис. 4).

```
const CustomDrawerContentComponent = (props) => (
  <View style={styles.container}>
    <DrawerItems {...props} />
  </View>
);
```

Рис. 4. View – основной компонент для отображения данных на устройстве

• Компоненты взаимодействия с API: обеспечивают взаимодействие с внешними API и обработку полученных данных (рис. 5).

```
endpoints: (builder : ... ) => ({
  getAllUsers: builder.query<IUser[], void>( definition: {
    query: () => "/",
  }),

  registerUser: builder.mutation<IUserResponse, IUserRequest>( definition: {
      query: (data : QueryArg ) => ({
        url: "/api/auth/register",
        method: "POST",
        body: data,
      }),
    }),

  login: builder.mutation<IUserResponse, IUserRequest>( definition: {
    query: (data : QueryArg ) => ({
      url: "/api/auth/login",
      method: "POST",
      body: data,
    }),
  }),
```

Рис. 5. Основной компонент для взаимодействия с внешними API

Заключение

В данной научной статье было проведено исследование современной архитектуры кроссплатформенных мобильных приложений на React Native. Была разработана оригинальная архитектура, учитывающая особенности фреймворка и обеспечивающая высокую производительность и удобство использования. Сравнительный анализ показал преимущества разработанной архитектуры по сравнению с другими подходами. Полученные результаты могут быть полезными для разработчиков, стремящихся создать эффективные и масштабируемые кроссплатформенные мобильные приложения на React Native.

Список используемых источников

1. Smith, J. Cross-platform Mobile App Development with React Native // Proceedings of the International Conference on Mobile Software Engineering and Systems. 2018.

2. Johnson, A. Architecting React Native Applications for Performance // Mobile Application Development Journal, vol. 9, no. 2, 2019, pp. 45–62.

3. Anderson, M. Best Practices for Cross-platform Mobile App Architecture on React Native // Journal of Mobile App Development, vol. 13, no. 4, 2020, pp. 87–102.

4. Patel, S. Performance Optimization Techniques for React Native Apps // Proceedings of the International Conference on Mobile Computing and Networking. 2021.

5. Clark, R. Building Scalable Cross-platform Apps with React Native // Mobile Development Magazine, vol. 15, no. 3, 2022, pp. 73–89.

Статья представлена научным руководителем,
старшим преподавателем кафедры ПИВТ СПбГУТ Помогаловой А. В.

УДК 004.72

И. В. Кильдяев (студент гр. ИСТ-921, СПБГУТ)
Д. М. Мельникова (студентка гр. ИСТ-923, СПБГУТ)

ПРИНЦИП ПОСТРОЕНИЯ САМООРГАНИЗУЮЩЕЙСЯ MESH-СЕТИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ YGGDRASIL

В современное время, создание сети, которая будет отказоустойчивой является главным направлением для бизнеса, так как решение этой проблемы позволит сэкономить огромные деньги, которые теряются при сбоях в сети. Основным способом решения проблемы является использования архитектуры самоорганизующейся mesh-сети. Данные сети имеют структуру mesh-сети и при этом не требуется больших усилий для их настройки. Одним из возможных решений, для создания такой сети является применения сетевой технологии – Yggdrasil. В ходе исследования выявлены возможности

и перспективы применения технологии Yggdrasil для создания самоорганизующейся mesh-сети.

Mesh-сети, Yggdrasil, самоорганизующиеся сети, Wi-Fi, сетевой интерфейс, виртуализация сетевых интерфейсов.

В последние годы сетевые технологии продолжают развиваться с невероятной скоростью, открывая новые горизонты для коммуникации и обмена информацией. Одной из самых захватывающих и инновационных концепций, которая привлекает все большее внимание, является построение самоорганизующейся mesh-сети.

Проблемой данных сетей является сложность выбора технологии из-за большого их количества и малой технической документации по данному направлению.

Самоорганизующаяся mesh-сеть [1, 2] – это децентрализованная сетевая структура, в которой каждое устройство (узел) играет роль источника и приемника данных. В отличие от традиционных централизованных сетей, где все данные проходят через центральный узел или сервер, в mesh-сети каждый узел может передавать данные непосредственно другим узлам, а не через единую точку.

Такая архитектура сети обеспечивает некоторые важные преимущества. Во-первых, она повышает надежность и устойчивость сети. Если один из узлов выходит из строя или недоступен, другие узлы могут обойти этот узел и продолжить передачу данных. Таким образом, самоорганизующаяся mesh-сеть обладает высокой отказоустойчивостью и способна функционировать даже в условиях, когда отдельные узлы не работают или недоступны.

Во-вторых, mesh-сети обладают гибкостью и масштабируемостью. Новые узлы могут быть легко добавлены в сеть без необходимости проведения сложной инфраструктурной работы. Это делает их особенно полезными в ситуациях, когда требуется быстрое развертывание сети или, когда нет доступа к традиционной сетевой инфраструктуре.

Самоорганизующиеся mesh-сети нашли применение в различных областях, включая мобильные сети, умный дом, городская инфраструктура, дистанционное обучение и многое другое. Они предлагают новые возможности для коммуникации и обмена данными, особенно в ситуациях, где традиционные сети могут быть ограничены или недоступны.

Технология Yggdrasil [3], названная в честь мифологического древа из скандинавской мифологии, воплощает идеалы меш-сетей, предоставляя простой и эффективный способ их построения. Она основывается на протоколе IPv6, который обеспечивает уникальность адресов и поддержку большого количества устройств. При этом Yggdrasil автоматически маршрутизирует данные между узлами, образуя гибкую и расширяемую сеть.

Самоорганизация в контексте технологии Yggdrasil проявляется в способности сети автоматически формировать и поддерживать соединения

между узлами без централизованного управления. Она позволяет узлам взаимодействовать и создавать маршруты обмена данными в децентрализованной среде.

Основным механизмом самоорганизации в Yggdrasil является применение протокола IPv6 и алгоритмов маршрутизации, которые позволяют узлам автоматически обнаруживать другие узлы в сети и устанавливать с ними соединения. Каждый узел в Yggdrasil имеет уникальный IPv6-адрес, который может использоваться для определения его местоположения в сети и определения оптимальных путей для передачи данных.

Когда новый узел присоединяется к сети Yggdrasil, он начинает обмениваться информацией с другими узлами, узнавая о их существовании и текущем состоянии сети. Это позволяет узлам строить динамические маршруты и перенаправлять данные по наилучшим путям в зависимости от текущей топологии сети. Если какой-либо узел выходит из строя или недоступен, другие узлы автоматически перестраивают маршруты, обходя этот узел и обеспечивая непрерывность передачи данных.

Самоорганизация Yggdrasil также включает в себя механизмы самовосстановления и самозащиты. Если какой-либо узел временно становится недоступным или подвергается сбою, сеть Yggdrasil автоматически перераспределяет трафик и перенаправляет его через доступные пути. Это позволяет сети сохранять работоспособность даже в условиях неполадок или атак.

Для создания самоорганизующейся mesh-сети с применением сетевой технологии Yggdrasil необходимо:

1) установить сетевую технологию Yggdrasil на выбранное устройство, для создания mesh-узла;

2) включить Yggdrasil с флагом -autoconf для автоматического поиска других mesh-узлов.

Установка сетевой технологии Yggdrasil – не сложная задача, так как все инструкции описаны на официальном сайте [4]. Например, для установки Yggdrasil на операционную систему Linux Mint необходимо:

1. Обновить официальные репозитории с помощью команды sudo apt install update.

2. Установить технологию yggdrasil с помощью команды sudo apt install yggdrasil.

3. Запустить службу yggdrasil для работы в фоновом режиме при запуске операционной системы.

Выполнение выше представленных инструкций можно увидеть на рис. 1.

Создание самоорганизующейся mesh-сети с помощью технологии Yggdrasil начинается с включения узла с помощью команды sudo yggdrasil –autoconf.

```
annakolegova@annakolegova-virtual-machine: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
annakolegova@annakolegova-virtual-machine:~$ sudo apt install yggdrasil
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Следующие НОВЫЕ пакеты будут установлены:
  yggdrasil
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 2 347 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 9 169 kB.
Пол:1 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 yggdrasil amd64 0.4.3-2ubuntu0.22.04.2 [2 347 kB]
Получено 2 347 kB за 5с (435 kB/s)
Выбор ранее не выбранного пакета yggdrasil.
(Чтение базы данных … на данный момент установлено 609174 файла и каталога.)
Подготовка к распаковке …/yggdrasil_0.4.3-2ubuntu0.22.04.2_amd64.deb …
Распаковывается yggdrasil (0.4.3-2ubuntu0.22.04.2) …
Настраивается пакет yggdrasil (0.4.3-2ubuntu0.22.04.2) …
Warning: The home dir /var/lib/yggdrasil you specified can't be accessed: No such file or directory
Adding system user `yggdrasil' (UID 127) ...
Adding new group `yggdrasil' (GID 137) ...
Adding new user `yggdrasil' (UID 127) with group `yggdrasil' ...
Not creating home directory `/var/lib/yggdrasil'.
annakolegova@annakolegova-virtual-machine:~$ sudo systemctl enable yggdrasil && sudo systemctl start yggdrasil
Created symlink /etc/systemd/system/multi-user.target.wants/yggdrasil.service → /lib/systemd/system/yggdrasil.service.
```

Рис. 1.Установка сетевой технологии Yggdrasil

После запуска данной команды, происходит запуск технологии Yggdrasil и выводятся данные:

1) название сборки;

2) версия сборки;

3) IP-адрес, на котором запускается слушатель для организации mesh-сети;

4) сокет, который используется для администрирования mesh-сети;

5) название виртуального сетевого интерфейса, который используется при создании mesh-сети;

6) адрес виртуального сетевого интерфейса;

7) максимальная единица передачи, используемая для сетевых пакетов mesh-сети;

8) публичный адрес mesh-узла;

9) IPv6 адрес, совпадающий с адресом виртуального сетевого интерфейса;

10) адрес подсети, для создания своей локальной сети на IPv6 адресации.

На рис. 2 представлен запуск сетевой технологии Yggdrasil на операционной системе Astra Linux.

```
sudo yggdrasil — Терминал Fly
Файл  Правка  Настройка  Справка
astra@astra:~$ sudo yggdrasil -autoconf
2023/06/12 02:33:32 Build name: yggdrasil-mobile
2023/06/12 02:33:32 Build version: 0.4.7-13-g8e6343d
2023/06/12 02:33:32 TLS listener started on [fe80::ed1e:bae9:7ada:9ad3%eth0]:44707
2023/06/12 02:33:32 UNIX admin socket listening on /var/run/yggdrasil.sock
2023/06/12 02:33:32 Interface name: tun0
2023/06/12 02:33:32 Interface IPv6: 200:15b2:6497:47d2:9438:3eb0:58ac:196a/7
2023/06/12 02:33:32 Interface MTU: 53049
2023/06/12 02:33:32 Your public key is f526cdb45c16b5e3e0a7d3a9f34ab71188b631c5326feeb4c62bb70135f2a384
2023/06/12 02:33:32 Your IPv6 address is 200:15b2:6497:47d2:9438:3eb0:58ac:196a
2023/06/12 02:33:32 Your IPv6 subnet is 300:15b2:6497:47d2::/64
2023/06/12 02:33:32 Connected inbound TLS: 201:61ed:e459:56b7:6699:f675:d262:f691@fe80::60ba:5b4c:a128:2e6e, source [fe80::ed1e:bae9:7ada:9ad3%eth0]:44707
```

Рис. 2. Запуск Yggdrasil для создания самоорганизующейся mesh-сети

В целом, подводя итоги исследования, можно сказать, что самоорганизация Yggdrasil обеспечивает гибкость, отказоустойчивость и масштабиру-

емость сети, при это не усложняя настройку mesh-узлов. Узлы могут динамически присоединяться и покидать сеть, а сеть сама регулирует и оптимизирует маршрутизацию данных. Это делает технологию Yggdrasil привлекательной для создания самоорганизующихся mesh-сетей в различных областях применения.

Список используемых источников

1. Introduction to Mesh. URL: https://en.wikipedia.org/wiki/Mesh_networking
2. Самоорганизующиеся mesh-сети для частного использования. URL: https://dzen.ru/a/ZCEST9LUWy8UmKjw
3. Официальный сайт технологии Yggdrasil Network. URL: https://yggdrasil-network.github.io/
4. Installing on Debian, Ubuntu, elementaryOS, Linux Mint and similar. URL: https://yggdrasil-network.github.io/installation-linux-deb.html

Статья представлена научным руководителем,
старшим преподавателем кафедры БИС СПБГУТ Гвоздковым И. В.

УДК 355.405.2

Ю. С. Молодцова, М. Е. Муратов (студенты гр. ИСТ-111, СПбГУТ)

СРАВНИТЕЛЬНЫ АНАЛИЗ РАЗЛИЧНЫХ МЕТОДОВ ШИФРОВАНИЯ ДЛЯ БЕЗОПАСНОЙ ПЕРЕДАЧИ ДАННЫХ

В современном цифровом мире, несомненно, почти все используют Интернет для различных целей. Чаще всего данные, передаваемые через Интернет, содержат конфиденциальную или частную информацию, которую люди хотят сохранить в тайне. Для сокрытия информации существует множество алгоритмов шифрования. Однако до сих пор ни одно из исследований не сравнивало различные типы алгоритмов более детально. В данной работе будет проведено сравнение широко используемых алгоритмов шифрования. Сравнение будет проведено на основе времени, затрачиваемого на шифрование и расшифровку данных, эффективности с точки зрения безопасности, использования памяти, энергопотребления, джиттера и задержки. Данный отчет также сосредоточен на поиске исследований о доступных методах шифрования данных и их широко используемых алгоритмах, а также на сравнении каждого алгоритма на основе общих параметров.

шифрование, безопасность систем, передача данных.

Введение

Быстрый технологический прогресс и используемое сегодня программное обеспечение позволили злоумышленникам перехватывать данные во время их передачи. В результате для любой организации или частного

лица чрезвычайно важно защитить свою чувствительную и ценную информацию. Безопасность часто означает защиту данных от несанкционированного доступа с помощью лучшей линии обороны. Шифрование – один из способов защиты информации, позволяющий сохранить данные в неизменном виде и защитить их во время передачи от отправителя к получателю. Шифрование определяется как процесс сокрытия информации от злоумышленников или неавторизованных лиц. И наоборот, расшифровка – это процесс преобразования информации обратно в ее нормальный формат и неизменное состояние [1].

Исследование

Исследование шифрования и дешифрования при передаче данных с использованием различных техник и алгоритмов проводилось на основе существующих данных из различных достоверных источников. Эти данные с использованием общепринятых алгоритмов были собраны и проанализированы.

Исследование было разделено на 3 направления:

1. Сравнение широко используемых алгоритмов симметричного шифрования Blowfish, AES, DES и 3DES.
2. Сравнение асимметричных алгоритмов RSA ECC и Elgamal.
3. Сравнение алгоритмов хэширования, а именно MD5, SHA1 и SHA256.

1 направление сравнение алгоритмов симметричного шифрования

Сравнение алгоритмов симметричного шифрования проводилось по следующим параметрам: времена шифрования, времена расшифровки, эффективности (уровень безопасности), энергопотребления, использования памяти, джиттера.

Из рис. 1 видно, что время, затраченное на шифрование тремя различными симметричными алгоритмами, показывает, что Blowfish использовал наименьшее количество времени для входных данных размером 1 МБ, 3 МБ, 7 МБ и 10 МБ соответственно. С другой стороны, алгоритм 3DES занял больше всего времени, за ним следуют AES-256 и DES.

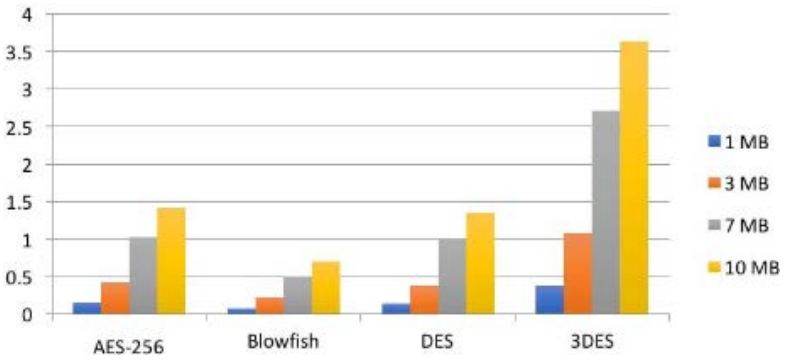


Рис. 1. Среднее время шифрования

Причина, по которой алгоритм 3DES занял больше всего времени, кроется в технике тройного шифрования. Этот алгоритм затрачивает 0,4 с, 1,1 с,

2,7 с и 3,6 с на зашифровку входных данных размером 1МБ, 3 МБ, 7 МБ и 10 МБ соответственно.

На рис. 2 показано сравнение 4 симметричных алгоритмов по среднему времени, затрачиваемому на расшифровку данных разного размера (МБ). Видно, что Blowfish затрачивает меньше всего времени на расшифровку данных с размером входных данных 1 МБ, 3 МБ, 7 МБ и 10 МБ. С другой стороны, алгоритму 3DES требуется наибольшее количество времени для расшифровки данных со средним временем расшифровки 0,3 с, 1,1 с, 2,7 с и 3,7 с при размерах входных данных 1 МБ, 3 МБ, 7 МБ и 10 МБ.

На рис. 3 показано сравнение использования памяти между тремя симметричными алгоритмами: DES, AES и Blowfish. Видно, что Blowfish использовал наименьший объем памяти (25,2 КБ), в то время как DES использовал наибольший объем памяти (43,2 КБ).

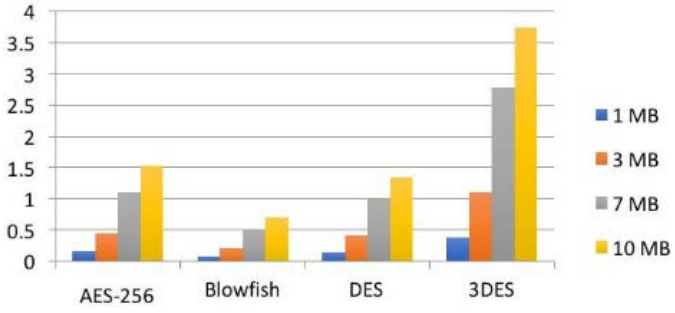


Рис. 2. Среднее время расшифровки для симметричных алгоритмов

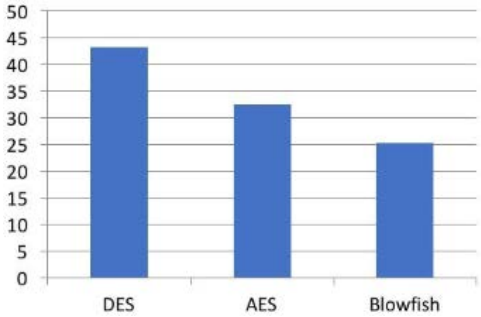


Рис. 3. Сравнение использования памяти между DES, AES и Blowfis

На рис. 4 показано энергопотребление при шифровании и дешифровании для DES, 3DES, AES и Blowfish. Видно, что 3DES потребляет больше всего энергии при шифровании и дешифровании и примерно в два раза больше, чем алгоритмы DES. Blowfish использовал наименьшее количество энергии для шифрования и расшифровки, а AES потреблял примерно в два раза больше энергии, чем Blowfish.

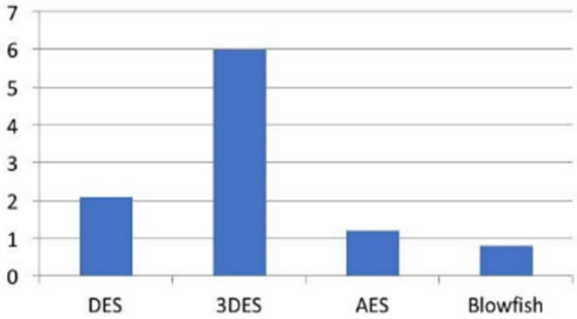


Рис. 4. Сравнение энергопотребления между DES, 3DES, AES и Blowfish [2]

На рис. 5 показана степень джиттера для трех различных полос пропускания с использованием 3DES, DES, AES и Blowfish. Видно, что степень джиттера улучшалась при увеличении пропускной способности. В результате, при использовании полосы пропускания 38 кбит/с возникало большое количество джиттера, от 0,007 до 0,012 мс. Более того, при использовании полосы пропускания 64 кбит/с для AES-256 и DES, они генерировали одинаковое количество джиттера. 3DES имеет наименьший джиттер для всех трех различных полос пропускания. В конечном итоге, при использовании

полосы пропускания 100 мбит/с 3DES имеет наименьшее количество джиттера, почти в два раза меньше, чем Blowfish, AES256 и DES.

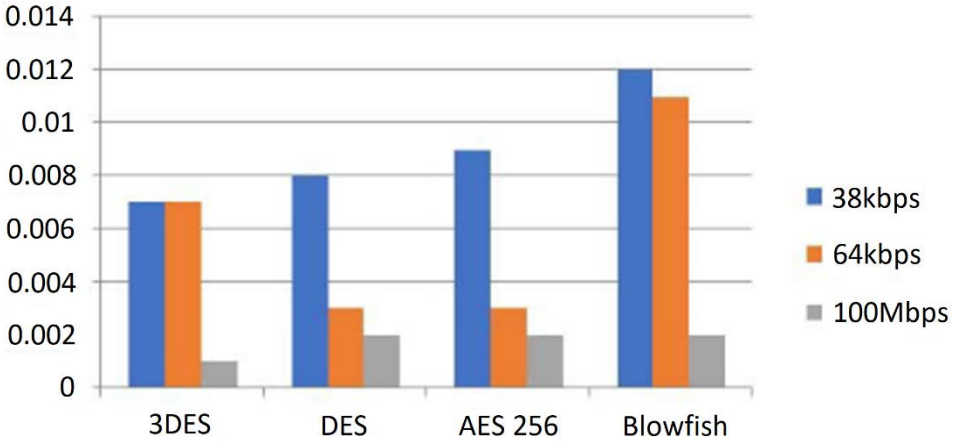


Рис. 5. Степень джиттера 3-х различных полос пропускания, используемых в 3DES, DES, AES и Blowfish

2 направление сравнение алгоритмов ассиметричного шифрования

Сравнение алгоритмов ассиметричного шифрования проводилось по следующим параметрам: производительность генерации подписи, эффективность проверки подписи, время шифрования, время расшифровки, уровень безопасности, использование памяти, энергопотребление.

В таблице 1 сравнивается время, затраченное на генерацию подписи ECC и ключа RSA при различных размерах входного ключа. Видно, что для генерации подписи ECC при различных размерах длины ключа требуется меньше времени, чем RSA.

ТАБЛИЦА 1. Производительность генерации подписи асимметричными алгоритмами [3]

Длина ключа (бит)		Время (с)	
RSA	ECC	RSA	ECC
1024	163	0,11	0,15
2240	233	0,15	0,34
3072	283	0,21	0,59
7680	409	1,53	1,18
15360	571	9,2	3,07

В таблице 2 показано время, затраченное на проверку подписи ECC и RSA при различных размерах входного ключа. Видно, что RSA использует меньшее время для проверки подписи при различной длине ключа по сравнению с ECC.

ТАБЛИЦА 2. Эффективность проверки подписи асимметричными алгоритмами (RSA и ECC) [3]

Длина ключа (бит)		Время (с)	
RSA	ECC	RSA	ECC
1024	163	0,01	0,23
2240	233	0,01	0,51
3072	283	0,01	0,86
7680	409	0,01	1,8
15360	571	0,03	4,53

В таблице 3 сравниваются 3 асимметричных алгоритма на основе параметров времени шифрования, времени дешифрования, времени генерации ключа, длины ключа и размера входных данных. Видно, что Elgamal потребовалось меньше всего времени на генерацию ключа, шифрование и расшифровку. ECC работает быстрее в генерации ключей, времени шифрования и дешифрования по сравнению с RSA.

ТАБЛИЦА 3. Сравнение между асимметричными алгоритмами (ECC, ELGAMAL и RSA) на основе различных метрик [3]

Размер вводимых данных	Алгоритмы	Длина ключа (бит)	Время генерации ключа (мс)	Время шифрования (мс)	Время расшифровки (мс)
100 байт	ECC	521	1779	390	358
100 байт	Elgamal	521	1107	297	218
100 байт	RSA	3072	56799	531	655

Это видно из рис. 6, где на графике сравнивается длина ключа каждого алгоритма и уровень безопасности, измеряемый в MIPS-годах. Исходя из длины ключа, ECC более эффективен, чем RSA, благодаря меньшему размеру ключа для достижения того же уровня безопасности.

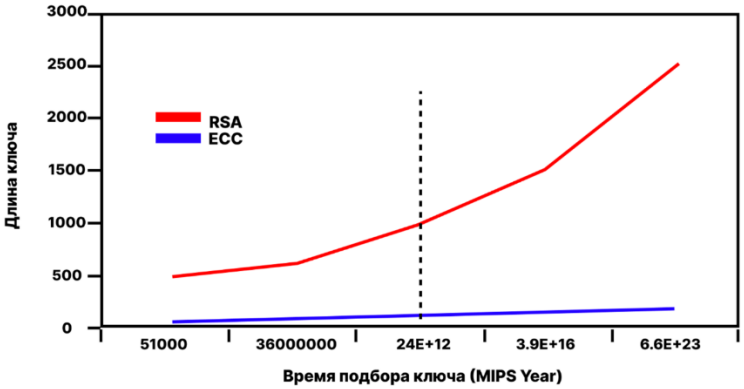


Рис. 6. Сравнительный уровень безопасности RSA и ECC

Использование небольшого размера ключа может привести к уменьшению времени вычислений, использования памяти и, таким образом, имеет лучшую эффективность, чем RSA.

ТАБЛИЦА 4. Требования к памяти для сравнения RSA и ECC при использовании ключей различной длины [4]

Алгоритм	Длина ключа (бит)	Использованная память (байт)
ECC	160	125
RSA	1024	313
ECC	210	140
RSA	2048	621
ECC	106	108
RSA	512	157

В таблице 5 показано сравнение симметричных алгоритмов ECC и RSA по энергопотреблению. Из таблицы видно, что ECC потребляет меньше энергии, чем RSA при различной длине ключа.

ТАБЛИЦА 5. Энергопотребление между ECC и RSA [4]

Алгоритм	Длина ключа (бит)	Энергопотребление (мДж)
ECC	163	13.02
RSA	1024	5.07
ECC	233	25.86
RSA	2048	28.51
ECC	283	50.13
RSA	4096	186.05

3 направление сравнение алгоритмов хэширования

На рис. 7 показано сравнение между SHA256 и MD5 по времени шифрования для различных размеров входных данных. Из графика видно, что разница между временем шифрования при размере входных данных 600 кб незначительна. Однако разница становится больше, когда размер входных данных увеличивается.

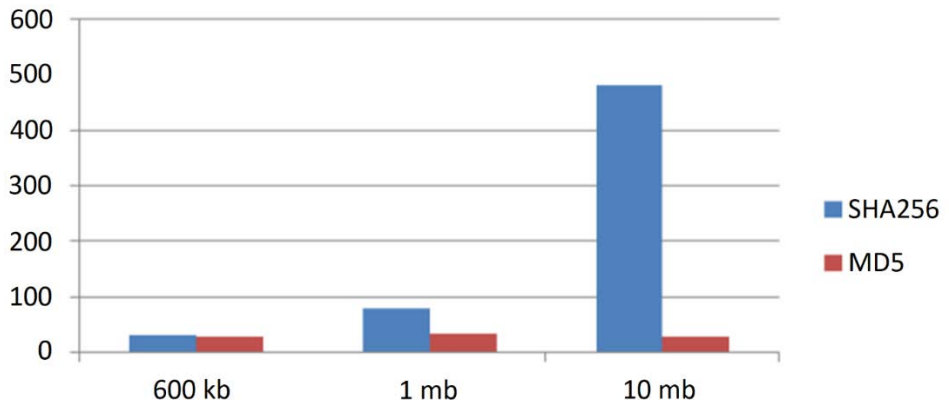


Рис. 7. Время, затрачиваемое на шифрование входных данных разного размера с помощью алгоритмов хэширования

Когда размер входных данных увеличился до 1 Мб, SHA256 потребовалось больше времени, чем MD5. Время шифрования для SHA256 было примерно в 5 раз больше, чем для MD5, когда размер входных данных составлял 10 Мб.

На рис. 8 показано энергопотребление MD5, SHA1 и SHA256. Видно, что MD5 потребляет наименьшее количество энергии, затем SHA256, потом SHA1.

На рис. 9 сравнивается задержка между MD5, SHA1 и SHA256. Видно, что MD5 генерирует наибольшую задержку, за ним следует SHA-2565, затем SHA-1.

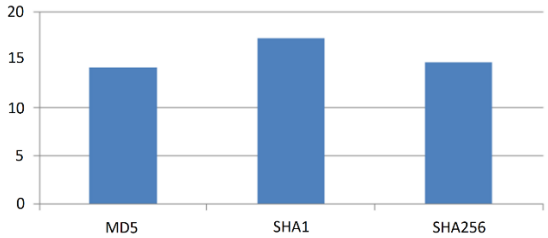


Рис. 8. Сравнение энергопотребления MD5, SHA1 и SHA256

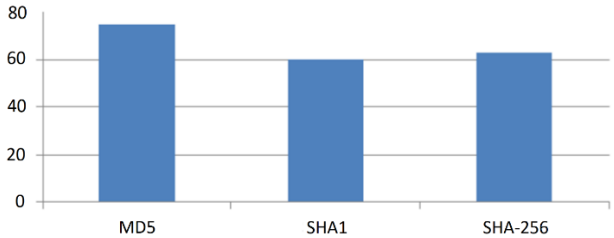


Рис. 9. Сравнение задержек между MD5, SHA1 и SHA256

Заключение

Сравнительный анализ различных методов шифрования при передаче данных показал, что Blowfish является лучшим по всем параметрам (время шифрования, время расшифровки, энергопотребление, использование памяти, задержка, джиттер и уровень безопасности) для симметричных алгоритмов по сравнению с AES, DES и 3DES. AES является вторым лучшим симметричным алгоритмом после Blowfish. С другой стороны, 3DES – самый неэффективный из всех алгоритмов.

Для асимметричных алгоритмов RSA, ECC и Elgamal, ECC лучше по большинству параметров по сравнению с RSA, за исключением времени, необходимого для проверки ключа подписи. Преимущество ECC перед RSA заключается в том, что ECC использует меньшую длину ключа для шифрования и дешифрования и при этом имеет тот же уровень безопасности, что и RSA. Асимметричные алгоритмы, такие как ECC и RSA, являются эффективными алгоритмами для шифрования данных. Однако они не так эффективны, как симметричные алгоритмы (AES и *Blowfish*), поскольку эти алгоритмы потребляют много ресурсов.

Что касается алгоритмов хэширования, MD5 генерирует наибольшую задержку, за ним следует SHA256, затем SHA1. В конечном итоге, SHA256 является более безопасным, чем SHA1 и MD5.

Список используемых источников

1. Г. Сингх и Суприя. Аналитическое сравнение методов симметричного шифрования и кодирования для облачной среды // Международный журнал компьютерных приложений 67(19).
2. Аль-Дулайми, Космас Дж., Мохаммед А. Самоорганизация и экологичные приложения конгнитивных радиосетей // IGI Global.
3. Использование информационных технологий в домохозяйствах. Австралия // Австралийское бюро статистики. http://www.abs.gov.au/ausstats/abs@.nsf/mf/8146.0 (дата обращения 20.04.2023).
4. Могос Г., Раду Г. Гибридный протокол защищенного уровня сокетов // Обзор Академии ВВС, № 2, стр. 91.

Статья представлена научным руководителем,
старшим преподавателем Эль Сабаяр Шевченко Н.

УДК 004.04:37

А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)
К. А. Нерман (студентка гр. ИСТ-012, СПбГУТ)

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ТРЕКЕРА ЗАДАЧ С СИСТЕМОЙ ВОЗНАГРАЖДЕНИЯ

Обоснована актуальность создания информационной системы трекера задач с системой вознаграждения. Описано влияние использования системы вознаграждений на мотивацию и воспитание детей. Определены основные недостатки существующих трекеров задач с системой вознаграждения. Представлена схема взаимодействия модулей информационной системы. Построена диаграмма вариантов использования информационной системы трекера задач с системой вознаграждения, демонстрирующая основной функционал системы. Определены перспективы развития информационной системы трекера задач с системой вознаграждения.

информационная система, проектирование, мотивация детей, финансовая грамотность, воспитание детей, тайм-менеджмент, планирование времени.

В настоящее время возрастает необходимость в использовании современных информационных технологий для оптимизации различных сфер жизнедеятельности, включая воспитание и обучение детей.

Одной из возможностей повышения мотивации детей к выполнению домашних заданий и учебных проектов может стать создание информационной системы, основанной на трекере задач с системой вознаграждения. В такой системе родители могут выдавать задания своим детям, а дети вы-

полнять их и получать за это определенное количество баллов, при использовании которых можно «покупать» себе подарки из списка вознаграждений, составленного родителями.

Актуальность проектирования информационной системы трекера задач с системой вознаграждения заключается в том, что многие родители не всегда могут контролировать режим дня своих детей, что может привести к негативным последствиям для их здоровья и развития. Информационная система может способствовать созданию более комфортных условий для ребенка, а также сокращению уровня стресса у родителей, что может помочь повысить мотивацию детей к обучению.

В контексте информационной системы трекера задач с системой вознаграждения предметной областью является сфера деятельности, связанная с организацией выполнения задач детьми и их вознаграждением за успешное выполнение. Она охватывает такие аспекты, как педагогика, психология и финансы.

Целью информационной системы является помощь родителям в эффективном управлении днем ребенка и в прививании ему полезных привычек, а также в мотивировании ребенка к более качественному времяпрепровождению при помощи системы вознаграждений.

Задачами информационной системы являются улучшение взаимоотношений детей с родителями, помощь в установлении более тесной связи между детьми и родителями и улучшение общения в семье. Взаимодействие в рамках системы может стать дополнительной возможностью для обсуждения проблем и решения конфликтов.

В настоящий момент существует множество информационных систем трекера задач с системой вознаграждения. Одной из них является приложение Forest, которое основано на идее роста виртуального дерева. Когда пользователь начинает работать над задачей, он сажает виртуальное семечко дерева, которое начинает расти. Если пользователь прерывает работу, чтобы проверить социальные сети или другие отвлекающие приложения, дерево умирает.

Еще одним аналогом является Habitica – информационная система, которая помогает пользователям развивать хорошие привычки и достигать своих целей. Она представляет собой комбинацию ролевой игры и трекера задач, где пользователь создает своего персонажа и зарабатывает награды за выполнение задач и развитие привычек. Наградами могут быть виртуальные предметы, увеличение уровня персонажа и многое другое.

Следующая информационная система – это приложение Todoist для управления задачами, которое позволяет пользователям устанавливать приоритеты для каждой задачи, добавлять комментарии и присваивать теги. Todoist также имеет систему наград, которая позволяет пользователям получать достижения за выполнение задач и накапливать очки, которые можно менять на бонусные функции приложения.

Последняя из рассматриваемых информационных систем – это приложение Beeminder, которое помогает пользователям достигать своих целей, обязывая их соблюдать определенную ежедневную дисциплину. Если пользователь не выполняет свои обязательства, то теряет финансовые средства. За успешное выполнение задач пользователи получают внутреннюю валюту для дальнейшего использования.

Основными недостатками перечисленных информационных систем являются невозможность отслеживания, действительно ли пользователь сфокусирован на выполняемом задании и каким именно заданием он занят, а также ограниченность функционала, поскольку нет возможности создания личной системы вознаграждений и получения заданий от других пользователей.

Всех пользователей проектируемой системы можно разделить на две группы: «Родитель» и «Ребёнок», которые имеют следующий функционал.

1. Регистрация пользователей: родители и дети должны иметь отдельные аккаунты с двумя разными ролями «Родитель» и «Ребенок» и возможностью входа в приложение.

2. Родитель может формировать список вознаграждений для ребенка с их стоимостью в баллах.

3. Родитель может создавать задания для ребенка, указывая название задания, срок выполнения, описание и количество баллов, которые ребенок получит за выполнение.

4. Ребенок должен иметь возможность просматривать задания, которые ему назначены и их описание.

5. Ребенок должен иметь возможность отмечать задания как выполненные и получать за это баллы. Родитель должен иметь возможность подтверждения выполнения ребенком задания, что необходимо для избежания обмана системы ребенком. Только после подтверждения баллы будут начислены.

6. Система баллов: приложение должно отображать количество накопленных ребенком баллов на экране и у родителя, и у ребенка.

7. Ребенок должен иметь возможность обменять свои баллы на подарки, которые родитель предоставляет в качестве вознаграждения за выполнение заданий. В случае выбора ребенком вознаграждения для «покупки» должно списаться указанное количество баллов.

8. Родитель должен иметь возможность просматривать историю заданий и баллов, полученных ребенком.

На основе функциональных требований выделены следующие модули информационной системы: «Личный кабинет родителя», «Личный кабинет ребенка», «Календарь», «Система вознаграждения», «Баллы», «Статистика», «Уведомления». Схема взаимодействия этих модулей представлена на рис. 1.

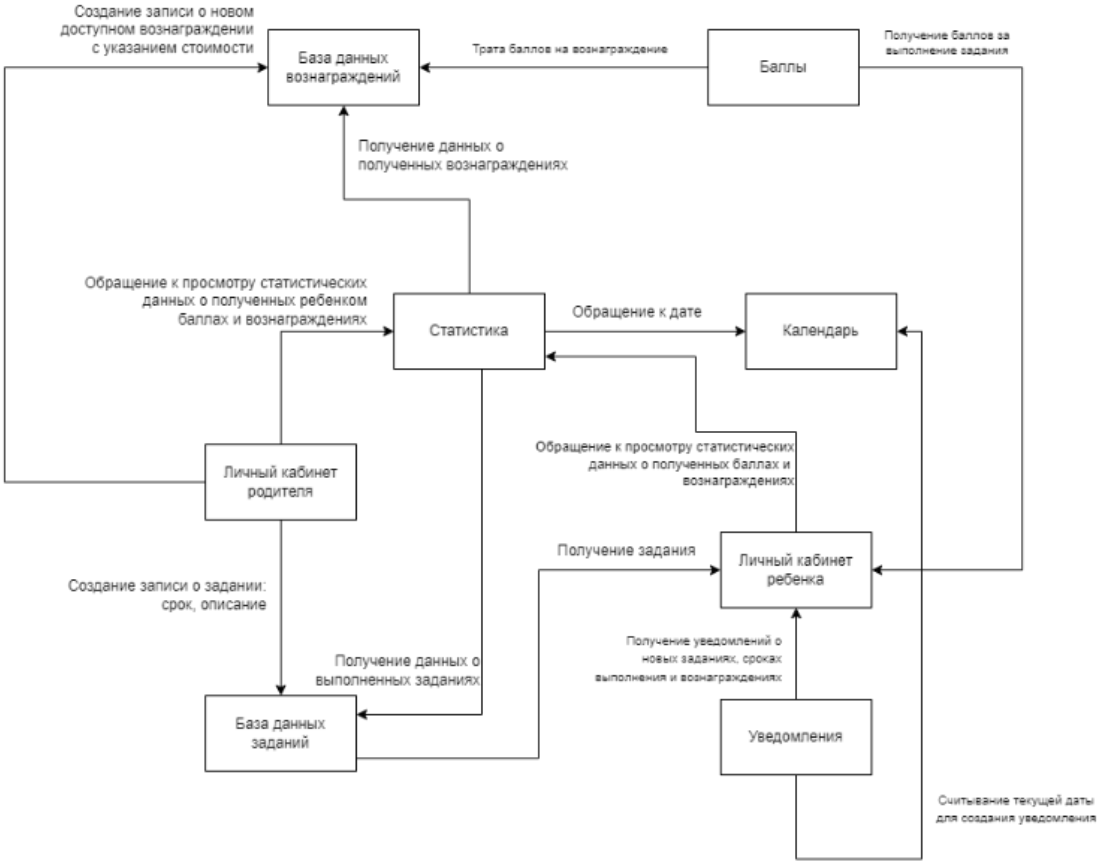


Рис. 1. Схема взаимодействия модулей ИС трекера задач с системой вознаграждения

Алгоритм работы системы включает следующие шаги.

1. Родитель и ребенок входят в систему в аккаунты с различными ролями.

2. Родитель при помощи идентификационного номера ребенка привязывает его аккаунт к своему.

3. Родитель создает задачу для ребенка в приложении-трекере задач, указывая ее описание, дату и время выполнения, а также количество баллов, которые можно заработать за ее выполнение.

4. Ребенок просматривает список задач в приложении и выбирает ту, которую он хочет выполнить.

5. Ребенок выполняет задачу в соответствии с ее описанием и сроком выполнения, ставит отметку о выполнении.

6. При успешном выполнении задачи ребенок получает назначенное количество баллов, которые засчитываются в его аккаунте в приложении.

7. Ребенок может использовать накопленные баллы для обмена на подарки, которые предлагаются родителями в системе вознаграждения.

8. Родитель может отслеживать прогресс своего ребенка в выполнении задач и контролировать его активность в приложении.

9. Приложение может отправлять уведомления и напоминания о сроках выполнения задач.

На основе функциональных требований и алгоритма работы системы составлена диаграмма прецедентов [1], представленная на рис. 2.

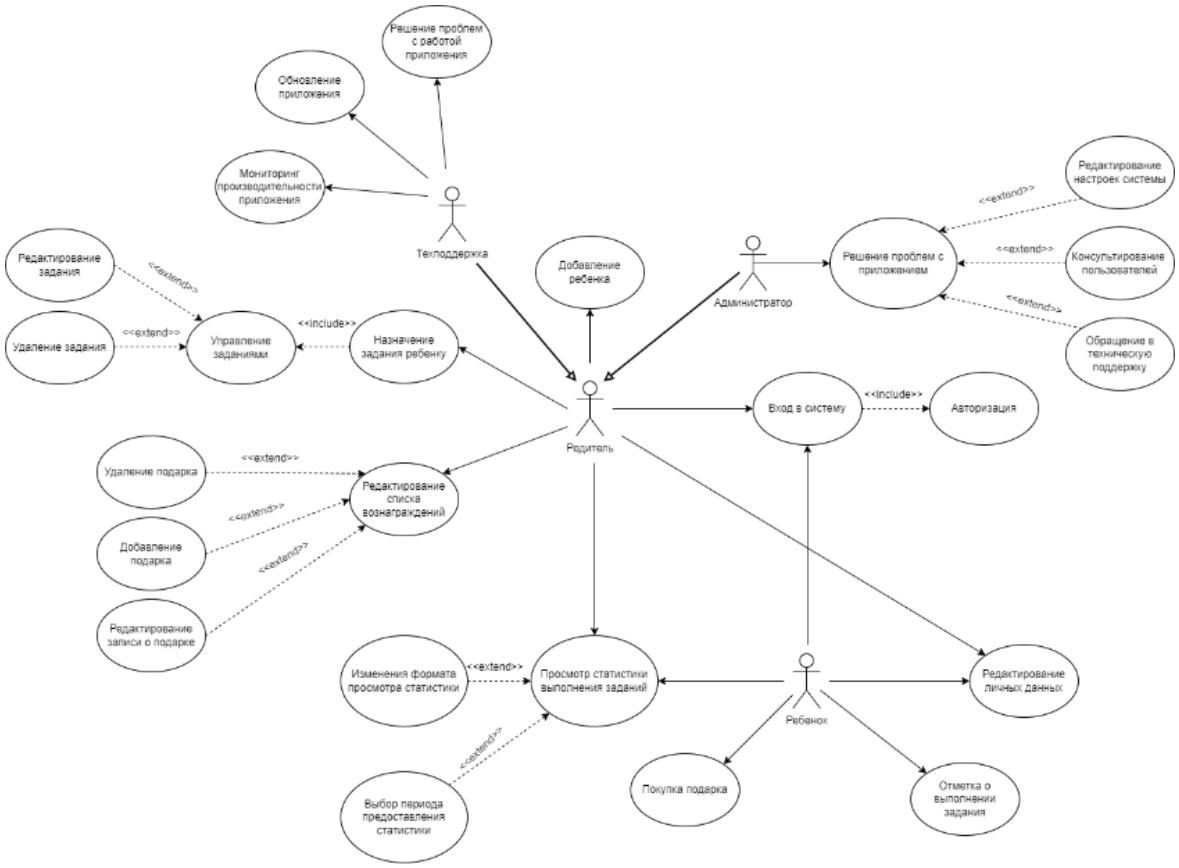


Рис. 2. Диаграмма прецедентов ИС трекера задач с системой вознаграждения

Кроме того, составлена диаграмма пакетов информационной системы, представленная на рис. 3 (см. ниже). Диаграмма пакетов описывает организацию классов и других элементов системы в пакеты и помогает визуализировать структуру системы на уровне компонентов и подсистем, что может быть полезно для понимания общей архитектуры системы и ее модульности.

Перспективы развития информационных систем включают в себя создание удобных, интуитивно понятных и доступных интерфейсов для пользователей, а также более совершенных алгоритмов обработки информации и анализа данных. Одной из ключевых тенденций является развитие и использование искусственного интеллекта, машинного обучения и больших данных для повышения эффективности работы информационных систем и улучшения качества жизни людей.

В области образования использование информационных систем, включающих трекер задач с системой вознаграждения, может стать широко распространенной практикой. Подобный механизм может повысить мотивацию детей к учебе и помочь родителям в управлении дневным режимом ребенка.

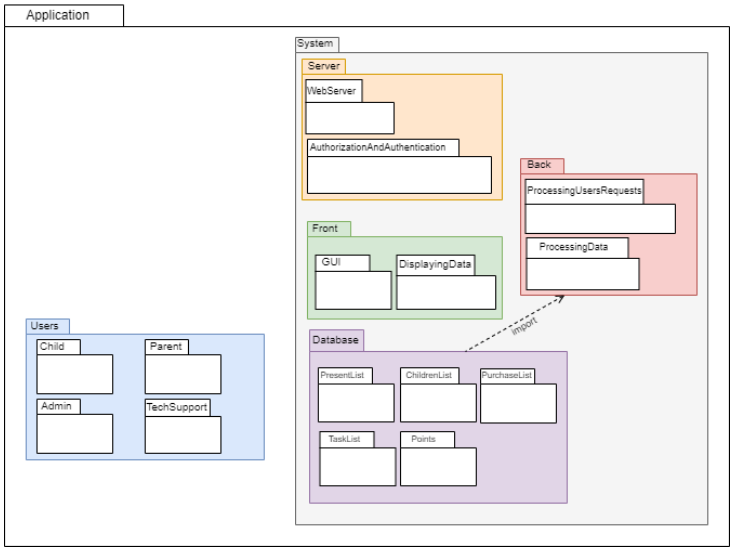


Рис. 3. Диаграмма пакетов ИС трекера задач с системой вознаграждения

Список используемых источников

1. Котлова М. В., Давыдова Е. В. Методы и средства проектирования информационных систем и технологий. СПб. : СПбГУТ, 2015. 62 с.

Статья представлена доктором технических наук, профессором Птицыной Л. К.

УДК 681.518.3

А. В. Рыбалко (студент гр. ИСТ-942, СПбГУТ)

РАЗРАБОТКА АППАРАТНО-ПРОГРАММНОГО УСТРОЙСТВА КОНТРОЛЯ СОСТОЯНИЯ СОТРУДНИКА АВАРИЙНО-СПАСАТЕЛЬНОЙ СЛУЖБЫ

В статье рассматривается система оперативного контроля за состоянием сотрудника аварийно-спасательной службы, которая оценивает основные биофизические параметры в реальном времени и получает данные для прогнозирования опасности возможных ситуаций. Представлена структурная схема системы, включающая в себя автономный модуль (блок измерения) и стационарный блок мониторинга. Предложен алгоритм работы системы. Представлен расчет тракта обработки сигналов от датчика пульса. Разработаны модули питания устройства, а также логические схемы блоков управления и передачи информации. Синтезированы фрагменты программ для автономного модуля и блока мониторинга в среде Arduino и LabVIEW соответственно. Приведены результаты моделирования системы и предложен вариант ее конструкции.

система контроля, датчик, алгоритм управления, математическая модель, структурная схема.

Целью контроля состояния физического состояния человека является сбор, обработка и анализ информации, выраженной в показателях тестов о всесторонней подготовленности, а также обеспечение непрерывного контроля за психофизиологическим состоянием спасателей.

На сегодняшний момент существуют следующие возможности организации такого типа систем. Первый – это умные жилеты, надеваемые сотрудником аварийно-спасательной службы, обычно изготавливается из плотного материала, защищающего его от телесных повреждений, огня и прочих физических воздействий [1]. В такой жилет могут быть «вшиты» все необходимые датчики, а также плата для сбора и передачи данных.

Второй вариант предполагает отдельное расположение датчиков по всему телу, руке, груди и ноге.

Первый вариант снижает количество вариантов размещения на теле человека и комбинацию применяемых датчиков, второй, как правило, представляет собой децентрализованную систему в силу того, что каждый датчик представляет собой самостоятельное диагностическое устройство и затрудняет получение интегральной оценки состояния человека, работающего в экстремальных условиях [2].

Таким образом, предлагаемый метод представляет собой получение биофизических показателей сотрудника аварийной службы с помощью датчиков пульса [3], температуры, артериального давления, частоты дыхания и последующую передачу этих данных в микроконтроллер с предварительной их обработкой. Процесс передачи данных о состоянии человека реализуется посредством беспроводной связи. На приемной стороне, получаемые вторым микроконтроллером данные преобразуются с целью последующего их вывода на экран специального контрольного монитора. Это необходимо для того, чтобы у руководителя спасательными работами были все необходимые сведения о текущем состоянии здоровья спасателей, занятых непосредственно в операции.

Структурная схема предлагаемой системы приведена на рис. 1 (см. ниже).

Представленная на рис. 1 структурная схема состоит из следующих компонентов: МП – модуль преобразования сигнала, МК 1, 2 – микроконтроллер 1, 2, МПП – модуль приёма-передачи данных, блок энергообеспечения, экран, датчик пульса, частоты дыхания, температуры.

Алгоритм работы системы описан блок-схемой, представленной на рис. 2 (см. ниже).

Рассмотрим работу модуля преобразования сигнала на примере пульсометра, на основе датчика пульса TCRT1000, используя программный пакет Multisim (рис. 3, см. ниже). Моделируемый датчик состоит из ИК-светодиода и фототранзистора.

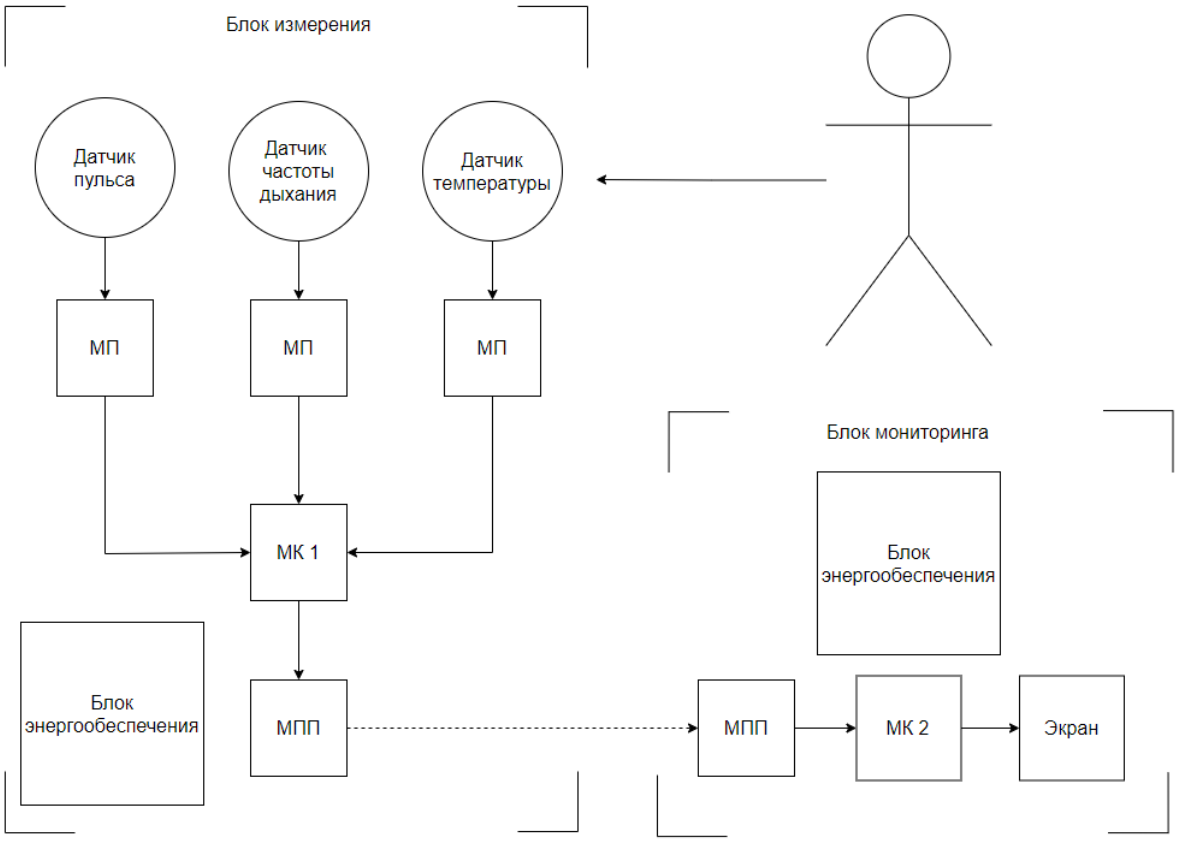


Рис. 1. Структурная схема системы оперативного контроля

Представленная схема позволяет устранить нежелательный сигнал постоянной составляющей и различные помехи, в том числе, наводку сети переменного тока 50 Гц. Базируется расчет данной схемы на основе математического аппарата для функционально - логического уровня. Моделирование и расчёт цифровой части системы основываются на булевой алгебре.

На принципиальной схеме показано, что сначала сигнал с ИК-сенсора проходит через пассивный фильтр верхних частот (C1, R4), чтобы избавиться от постоянной составляющей. Частота среза (f_c) такого фильтра определяется соотношением [1].

$$f_c = \frac{1}{2\pi R_4 C_1} = 0{,}7 \text{ Гц}.$$

Далее сигнал проходит через активный фильтр нижних частот (U2, R6, C2), составленный на основе операционного усилителя. Коэффициент усиления фильтра [3] и частота среза равны следующим значениям [2].

$$f_c = \frac{1}{2\pi R_5 C_2} = 2{,}34 \text{ Гц},$$

$$K = 1 + \frac{R6}{R5} = 101.$$

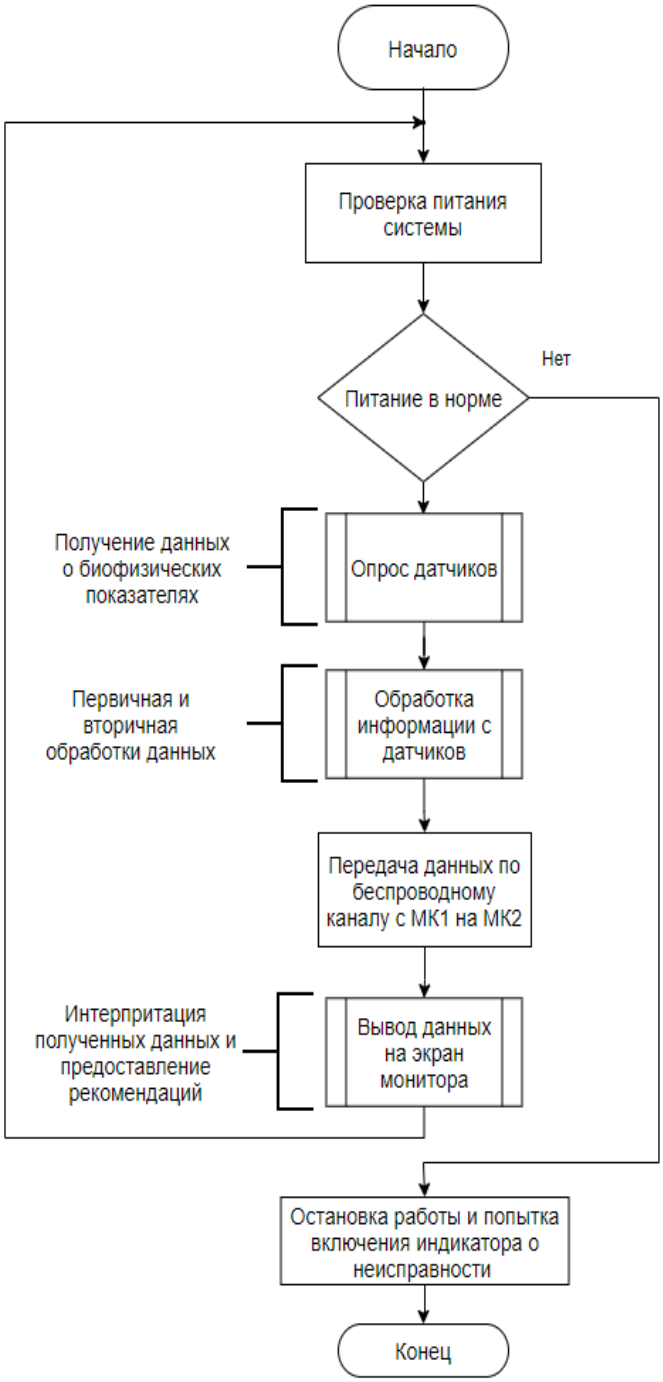


Рис. 2. Алгоритм работы системы оперативного контроля

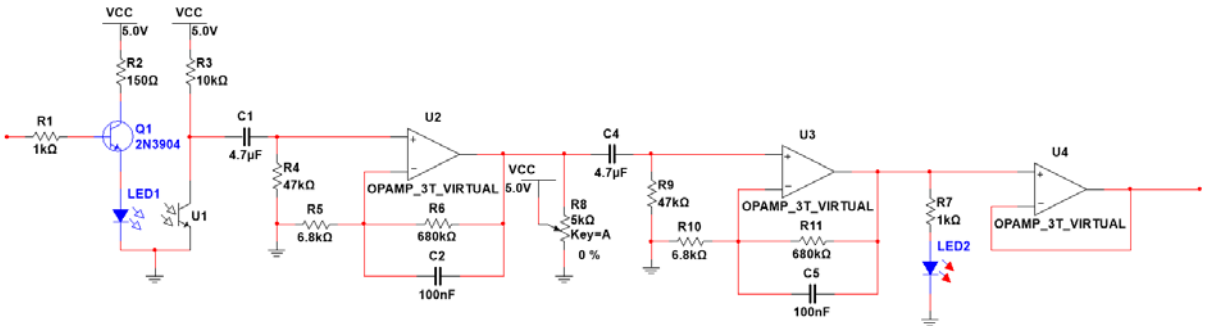


Рис. 3. Принципиальная схема первичной обработки сигнала пульсометра

С учетом второго каскада фильтров (ФВЧ, ФНЧ) максимальный коэффициент усиления модуля составит около 10^4. На выходе блока входной сигнал преобразуется в импульсы с КМОП или ТТЛ уровнями, которые синхронны с сердцебиением. Частота этих импульсов (*f*) связана с частотой сердечных сокращений (BPM) формулой [4].

$$BPM = 60 * f.$$

Диапазон напряжений питания платы сенсора, равный 3–5 В, позволяет использовать ее с семействами микроконтроллеров с напряжением питания от 3,3 до 5 В.

Для реализации системы можно использовать различные современные операционные усилители, программируемые логические и аналоговые интегральные схемы, а также первичные измерительные преобразователи как отечественного, так и зарубежного производства.

Для осуществления передачи данных между микроконтроллерами можно использовать два канала передачи данных. Первый канал будет передавать значение с датчика, а второй – код датчика, с которого поступают данные. Структурная схема передачи данных между МК представлена на рис. 4.

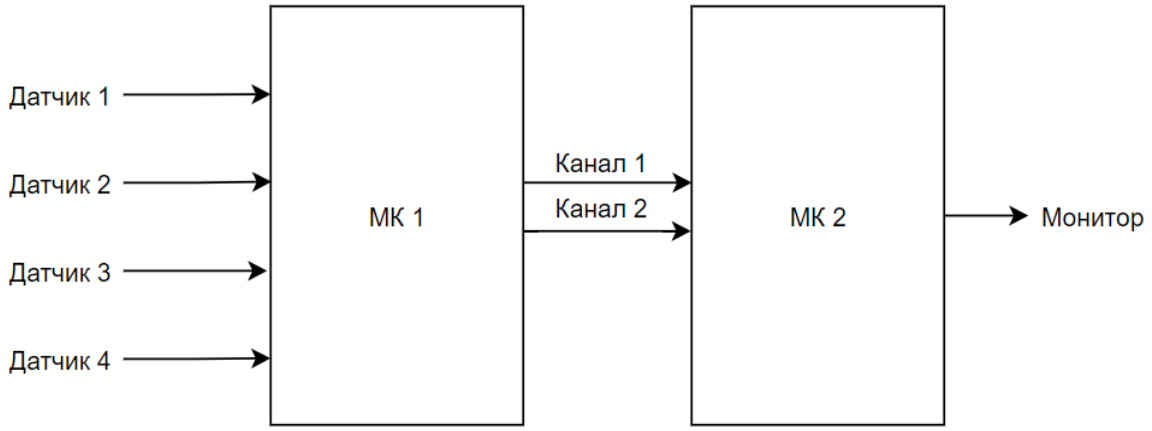


Рис. 4. Структурная схема передачи данных от датчиков на экран монитора

Данное решение заключается в преобразовании множества каналов передачи данных с датчиков в пару.

Достигается это следующим образом: Для обработки сигналов с датчика используется мультиплексор с подключённым двухразрядным Т-триггером для синхронной передачи данных. На т-триггер подаются единичные сигналы периодичностью в несколько секунд. На выходе триггера формируются сигналы от 1 до 4, которые в зависимости от значения и говорят о том, с какого датчика именно приходит сигнал.

В качестве результата моделирования системы рассмотрим виртуальный интерфейс оператора, выводимый на экран микроконтроллера.

На рис. 5 представлен пример задания параметров показателей здоровья выводимых на экран, составленный в среде LabVIEW. Данные параметры передаются на микроконтроллер, при этом значение каждого параметра передаётся раз в несколько секунд, после, согласно программе, данные выводятся на экран в соответствующей графе как представлено на рис. 6. Таким образом, каждой графе соответствуют следующие параметры: графа Heart отображает значение пульса; графа Blood – артериальное давление; графа Temp – температуру тела; графа Resp – частоту дыхания.

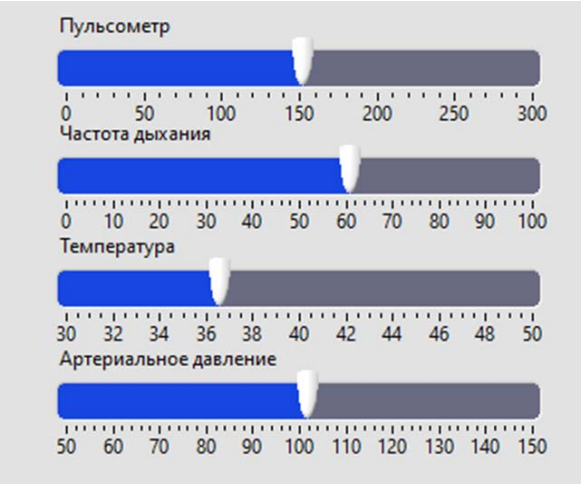


Рис. 5. Моделирование параметров здоровья, составленное в программе Labview

Рис. 6. Пример интерфейса пользователя системой

В статье были рассмотрены: структурная схема, алгоритм работы и результат моделирования в среде Labview системы оперативного контроля состояния сотрудника аварийно-спасательной службы, приведена актуальность разработки подобного класса систем. Также осуществлен выбор математического аппарата для расчета отдельных блоков системы.

Список используемых источников

1. Головин С. Обзор 9 образцов «умной одежды». URL: https://www.ferra.ru/review/health/mHealth-SmartWear.htm (дата обращения 11.02.2023).

2. ATHOS — «УМНЫЙ» КОСТЮМ ДЛЯ ТРЕНИРОВОК [Электронный ресурс]. URL: http://www.furfur.me/furfur/culture/culture/175145-bystree-vyshe-umnee-innovatsionnyy-kostyum-dlya-trenirovok-athos (дата обращения 11.02.2023).

3. Волкова Н. А. Алгоритм диагностики состояния сердечно-сосудистой системы по результатам многократных измерений артериального давления и пульса // Известия высших учебных заведений. Поволжский регион. Технические науки. 2015. № 1 (33). С. 43–49.

Статья представлена заведующей кафедрой ИСАУ СПбГУТ, доктором технических наук, доцентом Верховой Г. В.

УДК 004.9

А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)
А. А. Солдатов (студент гр. ИСТ-012, СПбГУТ)

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОРПОРАТИВНОГО ВИДЕОЧАТА

В статье представлено проектирование информационной системы корпоративного видеочата. Актуальность работы обусловлена неоспоримой пользой от использования видеочатов для упрощения коммуникаций внутри компаний. Анализ существующих систем видеочатов показал, что на данный момент на рынке отсутствует система корпоративного видеочата, отвечающего всем требованиям конечного пользователя. В ходе моделирования определены основные элементы проектируемой информационной системы. Построены диаграммы вариантов использования и классов информационной системы корпоративного видеочата.

видеоконференция, модель системы, онлайн-встреча, удаленная работа, коммуникация сотрудников.

С развитием технологий произошли существенные изменения коммуникаций в обществе: если раньше для того, чтобы поговорить с кем-то, было необходимо организовывать личную встречу, то благодаря техническому прогрессу стало возможным связаться с другим человеком практически из любой точки земного шара. Осуществляется это, в том числе, благодаря видеочатам.

Рабочий процесс в современной компании невозможно представить без использования видеочатов. Однако у существующих на данный момент информационных систем, предоставляющих подобный функционал, есть ряд недостатков: недоработки и сбои в работе системы, отсутствие поддержки продукта, неудобный интерфейс и т. д. В результате анализа существующих информационных систем сформированы общие требования к создаваемой информационной системе видеочата:

– качественные звук и изображение с веб-камеры;
– соблюдение протоколов информационной безопасности;
– постоянная поддержка сервиса, своевременное устранение уязвимостей и ошибок в работе системы;
– интуитивно понятный интерфейс;
– мультиплатформенность;
– возможность ведения длительных конференций (3 часа);
– число участников конференции – более 50;
– возможность обмена файлами;
– возможность демонстрации экрана;

– корпоративный пакет, предоставляемый на множество рабочих станций;

– доступ к облачному хранилищу;

– возможность установки на собственный сервер компании.

Проектируемая информационная система должна иметь следующий функционал:

– создание и завершение конференции;

– передача изображения с веб-камеры пользователя;

– обмен сообщениями;

– передача файлов;

– запись и просмотр конференции;

– демонстрация экрана;

– добавление пользователей в список контактов и удаление из него;

– регистрация пользователей;

– добавление пользователей в конференции.

Модель информационной системы корпоративного видеочата представлена на рис. 1.

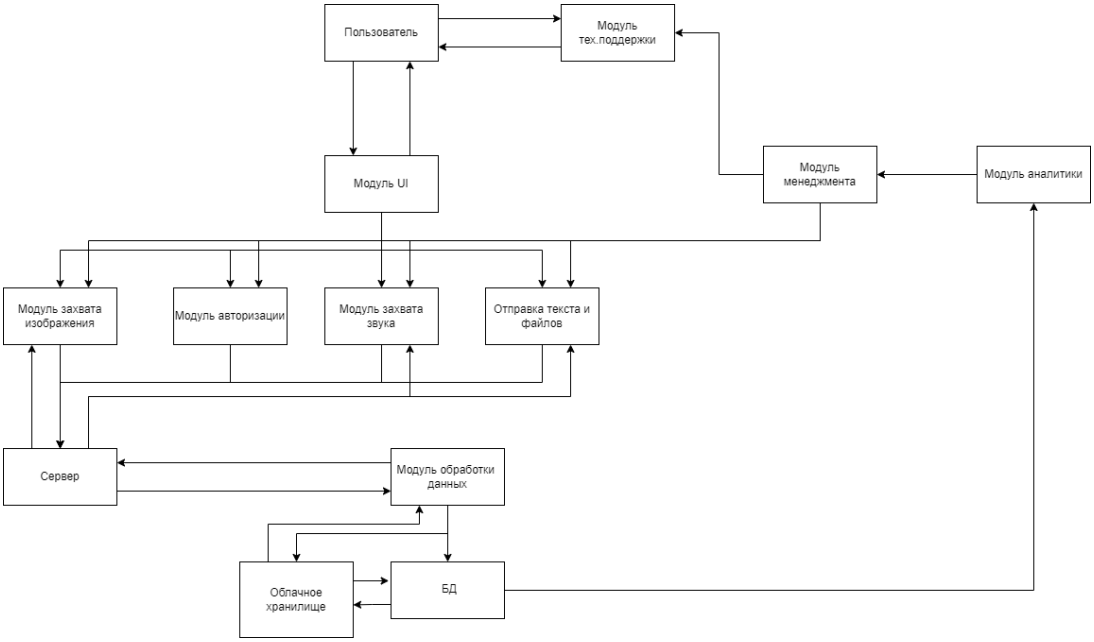


Рис. 1. Модульная схема информационной системы корпоративного видеочата

На представленной модульной схеме описаны модули, выполняющие функции, определенные в функциональных требованиях к информационной системе. Основными из них являются модуль захвата звука, модуль захвата изображения и модуль обработки данных.

Для более подробного описания функций, реализуемых в системе, используется диаграмма вариантов использования (рис. 2). Она отображает основные функциональные аспекты, присутствующие в системе. Вариант

использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом), и описывает типичное взаимодействие между пользователем и системой [1].

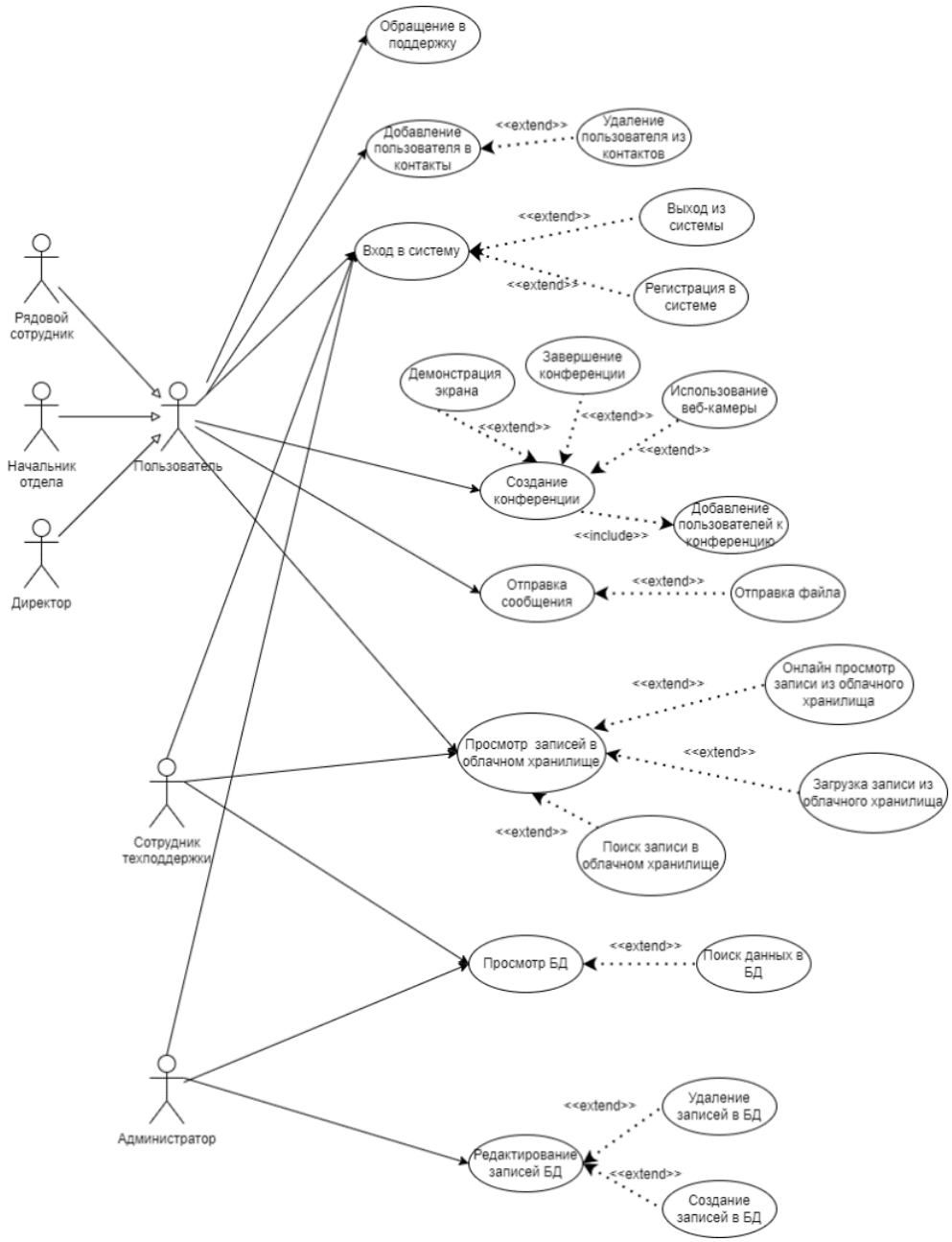


Рис. 2. Диаграмма прецедентов ИС корпоративного видеочата

Диаграммы последовательности используются для моделирования взаимодействия между объектами в системе и определения последовательности действий, которые объекты выполняют для реализации задач. На рис. 3 продемонстрирован процесс создания конференции.

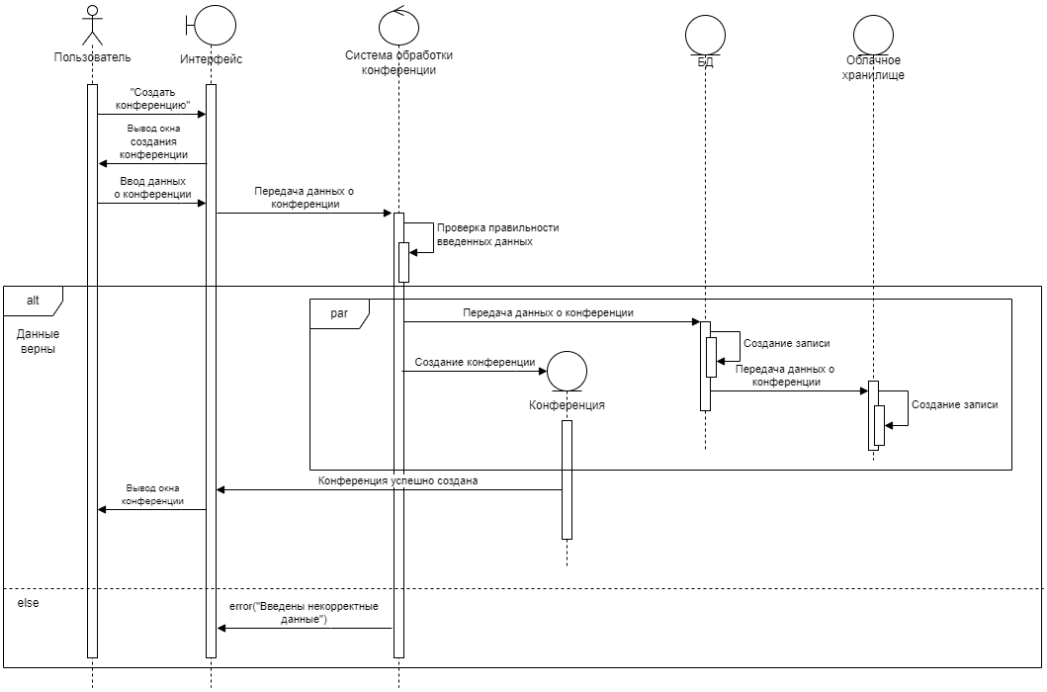


Рис. 3. Диаграмма последовательности процесса создания конференции

Диаграмма классов информационной системы видеочата представлена на рис. 4. Она демонстрирует основные классы системы и их взаимодействие [2].

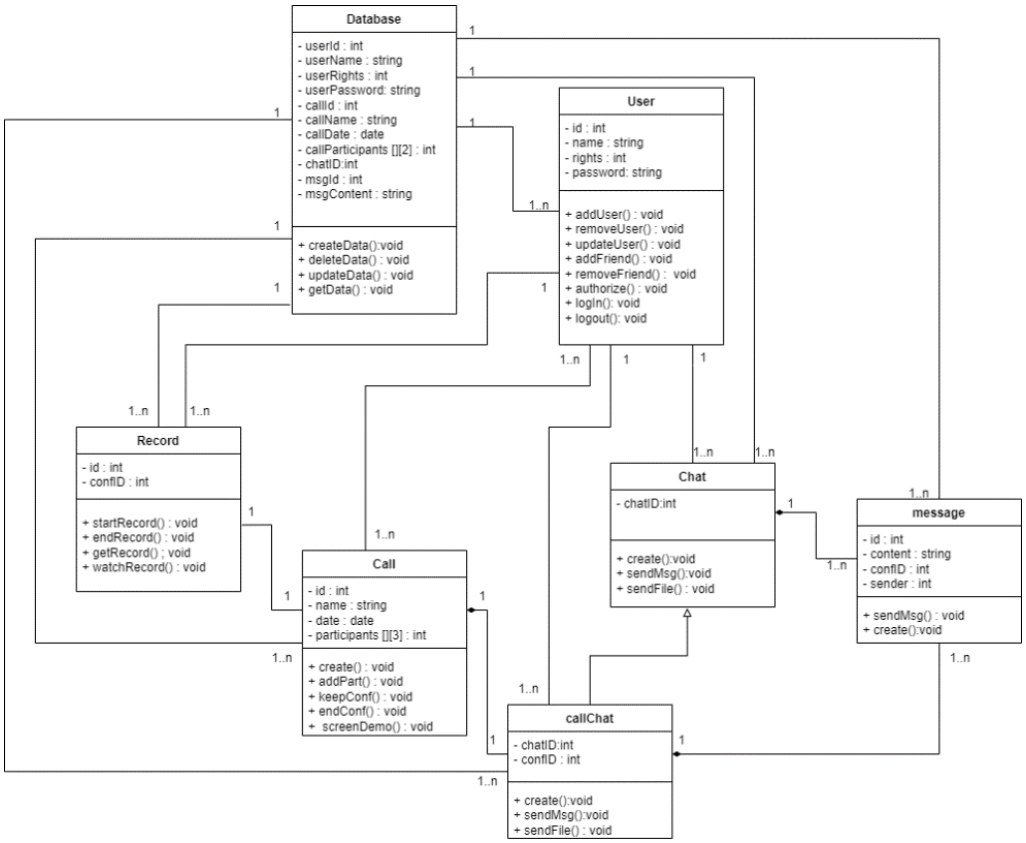


Рис. 4. Диаграмма классов информационной системы корпоративного видеочата

Макет пользовательского интерфейса проектируемой информационной системы представлен на рис. 5.

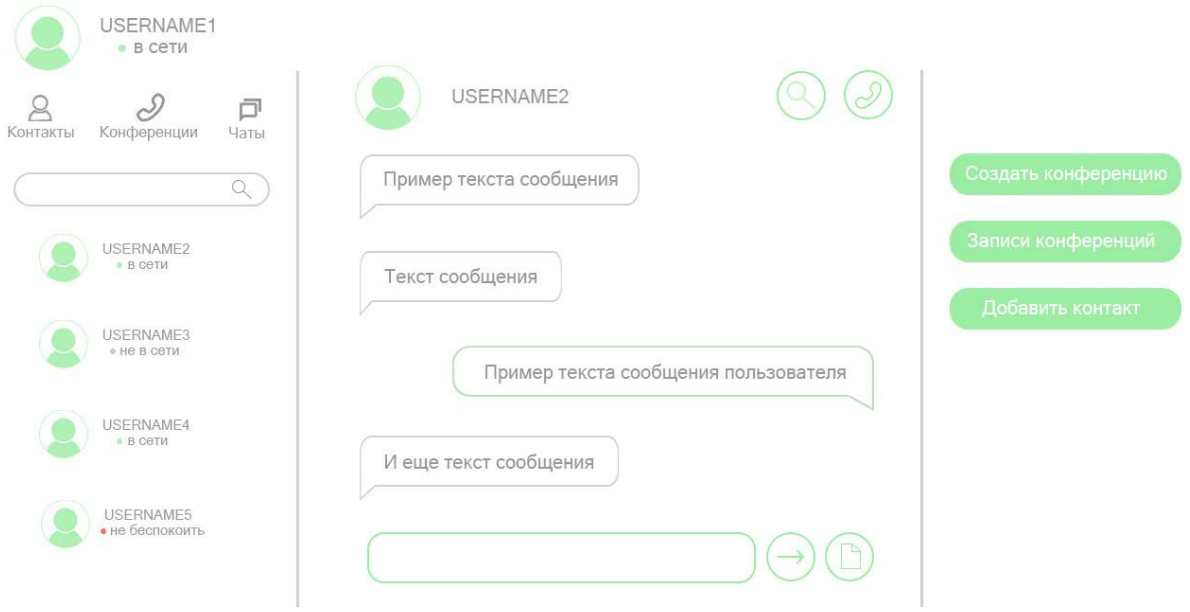


Рис. 5. Макет пользовательского интерфейса ИС корпоративного видеочата

На данном макете представлен интуитивно понятный пользовательский интерфейс. Для упрощения взаимодействия пользователя с системой кнопки, реализующие основной функционал, подписаны и имеют ассоциируемые с функциями пиктограммы.

Система корпоративного видеочата является важным инструментом для современных компаний, позволяющим обеспечить эффективную коммуникацию и совместную работу, а также увеличить производительность и улучшить качество работы команды.

В перспективах развития представленной информационной системы возможно внедрение нейросетей, повышающих качество видео и аудио, а также интегрирование модуля дополненной реальности.

Список используемых источников

1. Котлова М. В., Давыдова Е. В. Методы и средства проектирования информационных систем и технологий. СПб. : СПбГУТ, 2015. 62 с.

2. Буч Г., Рамбо Дж., Якобсон И. Введение в UML от создателей языка. М. : ДМК Пресс, 2010. 496 с.

Статья представлена доктором технических наук, профессором Птицыной Л. К.

УДК 004.58

Е. П. Толоконникова (студентка гр. ИСТ-213, СПбГУТ)

РАЗРАБОТКА МЕТОДА КОНТРОЛЯ ВРЕМЕНИ, ПРОВОДИМОГО В СОЦИАЛЬНЫХ СЕТЯХ

В век переизбытка информации мозг человека подвергается перегрузке вследствие постоянного потребления контента. Мы находимся в гиперинформационной среде, где каждую свободную минуту просматриваем ленту новостей и коротких видео. В данной статье описывается модель социальной информационной системы комплексного действия, конечной целью которой является отказ пользователя от отдыха в социальных сетях.

социальные сети, социальные информационные технологии, приложение, скроллинг.

Мы живём в гиперинформационной среде, где каждую свободную минуту скроллим (от англ. *scrolling* – «просматривание; прокрутка») ленту новостей, сообщений или видео. У каждого из нас свои мотивы. В этой статье я хочу уделить внимание двум: развлечение и отдых.

Человек способен работать в трёх режимах, всё его состояние регулируют три сети: центральная исполнительная сеть, сеть работы по умолчанию или дефолт система мозга и сеть выявления значимости [1]. Центральная исполнительная сеть возбуждается, когда мы потребляем информацию, когда мы сосредотачиваемся на определённой задаче. Сеть работы по умолчанию активна, когда мы думаем «ни о чём», когда нас не увлекают внешние раздражители. В этот момент все произошедшие события, укладываются в картину мира, происходит выработка эмоционального ответа на произошедший опыт и самое главное формирование своего места в этом мире. Эта сеть даёт нам ответы, когда мы их не ждём. Например, за обедом, к нам приходит осознание, как нужно было решать задачу на контрольной, которая была две недели назад, хотя мы не думали над этим вопросом, как нам кажется. Именно в этом состоянии нас посещают гениальные идеи [2].

В здоровом мозге эти две сети не могут быть активны одновременно. Координатором выступает третья – сеть значимости. Она определяет важность информации и подсказывает, какие видео стоит посмотреть, а какие пролистнуть. Наш мозг выбирает то, что проще, поэтому большую популярность обрели платформы с лентой из коротких видео, а девизом большинства контент-мейкеров стали «примитивность и красота».

Люди используют смартфон в качестве пассивного отдыха, но, в результате, не чувствуют себя отдохнувшими и атрофируют своё мышление. Они начинают мыслить стереотипно, шаблонно, не развиваются, имеют проблемы с концентрацией. Также чрезмерное использование социальных

сетей приводит к понижению самооценки и опасениям по поводу социального одобрения [3].

В ходе работы был создан опрос «Время в сети», участие в котором приняли 84 студента, ниже представлены полученные результаты:

- 33 (39,3 %) человека проводят в соцсетях 3–4 часа; 29 (34,5 %) человек – более 5 часов; процент этих людей от общего числа составляет 73,8 %;
- 61 (72,6 %) чел. считает, что проводит много времени в соцсетях;
- 55 (65,5 %) чел. ответили, что скроллят ленту в качестве отдыха;
- 54 (64,3 %) чел. ответили, что скроллят ленту в качестве развлечения;

Стоит сделать уточнение, что в последних двух пунктах 41 человек (48,8 %) выбрали оба варианта.

- 50 (59,5 %) человек не чувствуют себя отдохнувшими, после скроллинга; 3 человека воздержались от ответа;
- 26 (30,9 %) людей не считают свою жизнь интересной и насыщенной; 3 – воздержались от ответа.

Из них 11 (42,3 %) человек проводят 3–4 ч/день в соцсетях; 6 (23,1 %) человек – больше 5 ч/день; общий процент составляет 65,4;

- 31 (36,9 %) человек сравнивает себя с людьми из интернета, свою жизнь с их жизнью.

Из них 12 (38,7 %) человек проводят 3–4 ч/день в соцсетях; 13 (41,9 %) человек – больше 5 ч/день; общий процент составляет 25 человек 80,6 %;

- на вопрос «Бывало ли у вас чувство, что вы сели посмотреть пару видео, очнулись, а уже прошло несколько часов?» 39 (46,2 %) человек ответили «да»; 17 (20,2 %) человек ответили «да, довольно часто»; 1 воздержался; процент этих людей от общего числа составляет 66,4 %;
- 35 (41,6 %) человек хотят избавиться от привычки сидеть в соцсетях; 23 (27,4 %) пытаются контролировать своё время нахождения в социальных сетях, но это плохо получается; 2 человека воздержались;
- 47 (55,9 %) человек хотели бы получать своевременные напоминания о выходе из соцсетей или уведомления со временем, которое они уже провели за листанием ленты; 1 человек воздержался.

Более половины опрошенных предпочли бы напоминания о превышении времени в соцсетях.

Отказ от «залипания» в соцсетях является волевым решением, представляющим сложность в случае «хронического» явления. Поэтому был разработан метод контроля времени в информационной системе для помощи в отказе от чрезмерного использования соцсетей.

Предлагается создание веб-приложения, которое способствует потреблению качественного контента и предотвращению перегрузки мозга в век переизбытка информации.

Функции приложения:

– контроль времени скроллинга при помощи счётчика времени;

– уведомление о выходе из социальных сетей;

– парсинг (парсинг – автоматизированный сбор и структурирование информации с сайтов при помощи программы или сервиса) афиш с культурными событиями;

– уведомление о предстоящих интересных мероприятиях в Санкт-Петербурге;

– парсинг сайта с прогнозом погоды;

– анализ погоды;

– рекомендация дней для прогулки;

– обновление статей на темы «какое влияние на мозг оказывают гаджеты», «чем можно заняться в свободное время», «как можно отдохнуть без гаджетов»;

– оценка местоположения пользователя;

– оценка пройденного пользователем расстояния при помощи шагомера;

– ведение рейтинга среди пользователей и вознаграждение стимулирующими баллами.

Стимулирующие баллы будут начисляться за:

- количество пройденных шагов в день, неделю и месяц;
- приглашение друга;
- количество часов, которые пользователь отсутствовал в социальных сетях;
- за местоположение пользователя: если точка его локации находится в месте, где проходит культурное мероприятие;

Система вознаграждения предусмотрена для мотивирования пользователя к сокращению времени, проводимого за гаджетом.

Предполагаемый эффект, оказываемый приложением, должен быть следующим:

– мозг больше отдыхает;

– пользователь избавляется от клипового мышления;

– появляется больше свободного времени;

– улучшается ментальное и физического здоровье;

– в жизни становится больше физической активности;

– расширяется кругозор;

– появляется много новых впечатлений;

– приходит больше креативных идей;

– уходит сравнивание своей жизни с красивой картинкой в социальных сетях;

– формируется внутренний мир;

Таким образом, предложенный метод позволит достичь цели сокращения времени, проводимого в соцсетях и отказ от проведения досуга и отдыха в гаджетах.

Список используемых источников

1. Белоусова М. Три кита мышления ЦИС, СВЗ, ДСМ [Электронный ресурс] // URL: https://vk.com/@sense.academy-tri-kita-myshleniya-cis-svz-dsm (дата обращения 27.06.2023).
2. Белоусова М. Как работает мозг в разных режимах? [Электронный ресурс] // vk.com. URL: https://vk.com/@sense.academy-defolt-sistema-mozga-bluzhdanie (дата обращения 27.06.2023).
3. Киракасянц А. Доктор Курпатов из Сбербанка: «С телефоном вы глупеете» [Электронный ресурс] // frankmedia.ru. URL: https://frankmedia.ru/10900 (дата обращения 27.06.2023).

Статья представлена научным руководителем,
старшим преподавателем Соловьевой А. В.

УДК 004.27

А. А. Шакурова (студентка гр. ИСТ-132м, СПбГУТ)

РАЗРАБОТКА МОДЕЛИ ПАРСИНГА ДАННЫХ

В данной статье рассматривается вопрос о сокращении временных и денежных затрат на процесс поиска и сбора информации с веб-сайтов. На основе проведенных исследований разрабатывается модель парсинга данных, включающая в себя дополнительный модуль работы с поисковой системой. В результате модифицированная модель парсинга данных может быть полезна для использования в системе мониторинга товаров конкурентов владельцами интернет-магазинов.

модель парсинга данных, поисковая система, интернет-магазин.

Проблема исследования заключается в том, что на поиск и анализ какой-либо информации в сети Интернет тратится большое количество как денежных, так и временных затрат. Особенно данная проблема актуальна для мониторинга конкурентов интернет-магазинов. Так как по данным аналитики Data Insight [1] в 2022 году общий оборот рынка розничной интернет-торговли в России увеличился на 38 % по сравнению с 2021 годом. Также, по их оценкам, за 2022 год количество онлайн-заказов в РФ выросло на 65 %. Положительная динамика ведет к тому, что в дальнейшем число интернет-магазинов, а значит и количество конкурентов в различных нишах будут и возрастать.

Научные исследования в области парсинга данных интернет-магазинов [2–5] доказывают, что парсинг позволяет упростить сбор данных с сайтов конкурентов, несмотря на возможные проблемы. Однако, они не включают в себя работу с поисковыми системами. А также, они нацелены только

на программную часть и не затрагивают проблемы взаимодействия пользователя с программой.

Модель парсинга данных разрабатывается с целью снизить затрачиваемое время на выполнение процессов поиска и сбора информации за счет их автоматизации. Также, разработка модели нацелена на упрощение сложности взаимодействия пользователя с интерфейсом программы-парсера. Таким образом, владельцы интернет-магазинов смогут быстро и легко осуществлять мониторинг своих конкурентов. На основании собранных данных они смогут эффективно управлять ценообразованием в магазине для повышения своих продаж и конкурентоспособности на рынке.

Разработка модели парсинга данных основывается на проведенных ранее исследованиях:

– исследование инструментов поиска и сбора информации;

– исследование пользовательских интерфейсов (ПИ) программ-парсеров.

В результате было определено, что в модель парсинга данных нужно внедрить дополнительный модуль работы с поисковой системой для исключения необходимости в самостоятельном поиске конкурентов. Были отобраны библиотеки языка программирования (ЯП) Python, подходящие для извлечения данных с веб-страниц интернет-магазинов: requests, lxml и selector. Также, по результатам исследований были определены: основной функционал парсеров и критерии оценки ПИ для разработки собственной программы мониторинга.

Модель парсинга данных состоит из трех основных компонентов:

– парсер, который осуществляет поиск и сбор информации с сайтов конкурентов;

– файл-обработчик, который добавляет найденную информацию в базу данных, тем самым обеспечивает быстрый поиск по накопленной информации;

– веб-интерфейс – пользовательский графический интерфейс для работы пользователя.

Наглядно архитектура модели парсинга данных представлена на рис. 1.

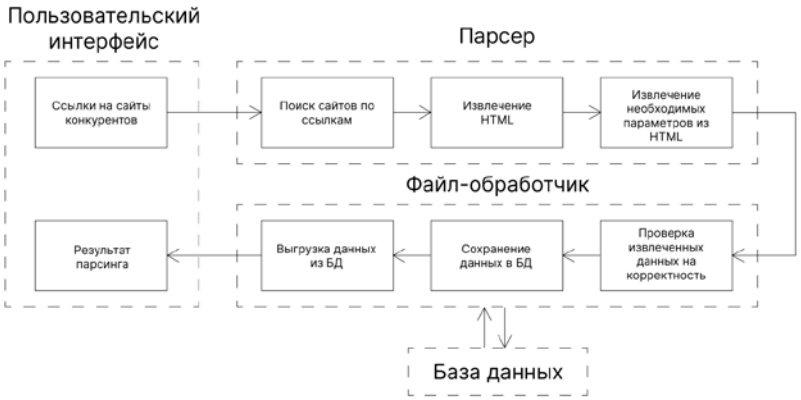


Рис. 1. Архитектура модели парсинга данных

В данном случае парсер включает в себя только один модуль – модуль поиска информации по заданным ссылкам. Модификация модели парсинга данных заключается во внедрении дополнительного модуля, который позволит осуществлять поиск информации по заданным запросам. После внедрения дополнительного модуля, пользователь сможет, отправив запрос, получить список его потенциальных конкурентов, что значительно ускорит процесс их поиска и отбора. Алгоритм работы парсера после модификации представлен на рис. 2.

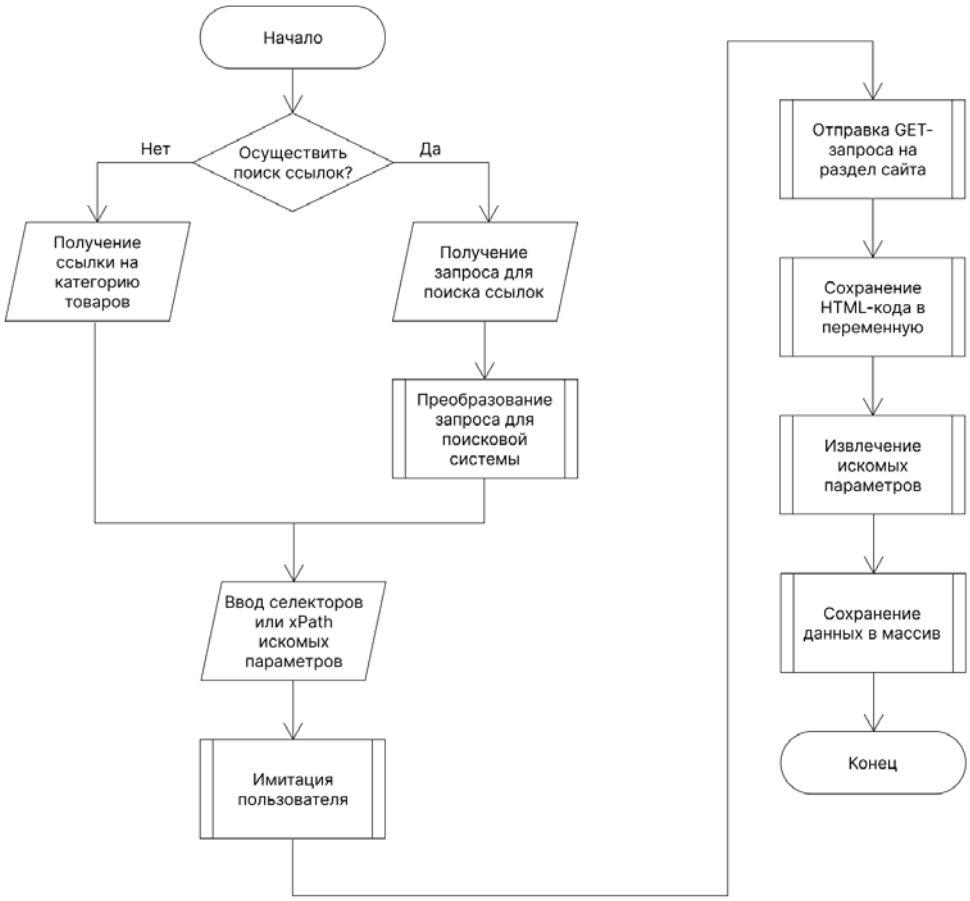


Рис. 2. Алгоритм работы парсера с модулем поиска по заданным запросам

На диаграмме вариантов использования нотации UML отображено взаимодействие пользователя с программой-парсером (рис. 3). В качестве пользователей в данном случае выступают: владелец интернет-магазина и системный администратор. Первый имеет возможность добавить свои товары, список своих конкурентов, запустить парсинг товаров конкурентов и парсинг поиска конкурентов, а также связаться с технической поддержкой. Второй имеет доступ к собственной панели управления, где он может отвечать на вопросы пользователя и принимать обратную связь от него, проводить диагностику системы и устранять ее неполадки или сбои. Также, он имеет доступ к настройке парсера.

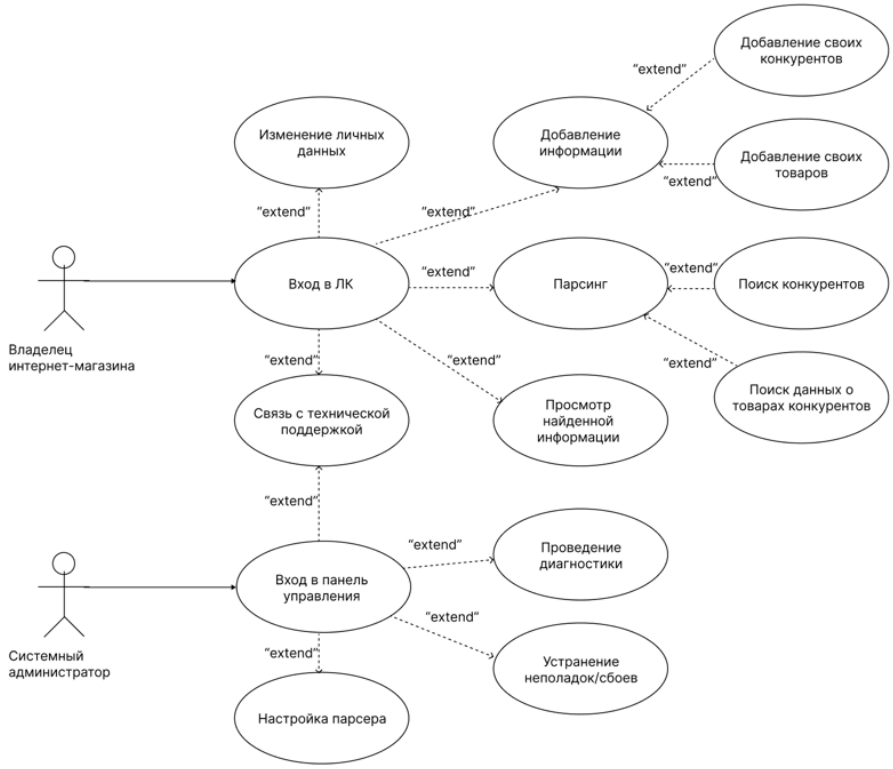


Рис. 3. Диаграмма взаимодействия пользователя с программой-парсером

Модификация модели осуществляется на ЯП Python, так как у него имеются специальные библиотеки для парсинга данных.

На рис. 4а демонстрируется результат парсинга цен с веб-сайта, ссылку на который ввел пользователь, а на рис. 4б – результат парсинга по запросу пользователя.

```
query = input('Введите ссылку на категорию товара: ')
result_query = requests.get(query)
form_text = result_query.text

no_sale_price = "div.product-item__price"
for node in HTMLParser(form_text).css(no_sale_price):
  #print(node.html) #все найденные теги с классом
  print(node.text(deep=True, separator='', strip=False)) #содержимое тега
  #node.text(deep=True, separator='', strip=False)

Введите ссылку на категорию товара: https://hollyshop.ru/catalog/basecare/maski/?page=1

2 630 ₽
2 550 ₽
213 ₽
2 390 ₽
234 ₽
200 ₽
420 ₽
1 550 ₽
340 ₽
290 ₽
460 ₽
```

```
query = input('Введите запрос: ')
result_query = requests.get('http://www.google.com/search?q='+query)
form_text = result_query.text

url = "div.byrV5b"
for node in HTMLParser(form_text).css(url):
  url_competitor += node.text(deep=True, separator='', strip=False)

print('\n'.join(url_competitor))

Введите запрос: пекарни спб
https://british-bakery.ru/
https://peterbakery.ru/
https://gorpekarni.ru/
http://www.pudhleba.ru/
http://fvolchek.ru/
https://bushe.ru/
```

Рис. 4. Результаты поиска данных по заданным ссылкам (а) и по заданным запросам (б)

Для разработки прототипа программы выбран фреймворк Django, так как ранее модель была разработана на основе библиотек ЯП Python, а в качестве базы данных (БД) – легкая СУБД SQLite. Архитектура прототипа представлена паттерном MVT – Model, View, Template (рис. 5). Структура модели БД представлена на рис. 6.

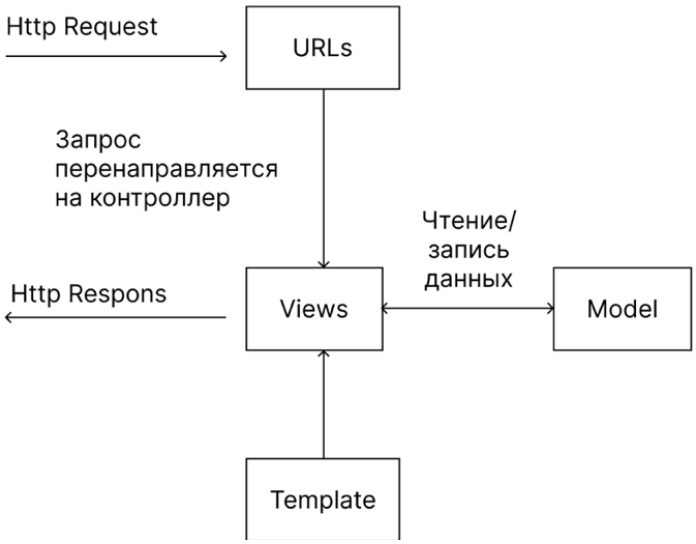


Рис. 5. Архитектура прототипа

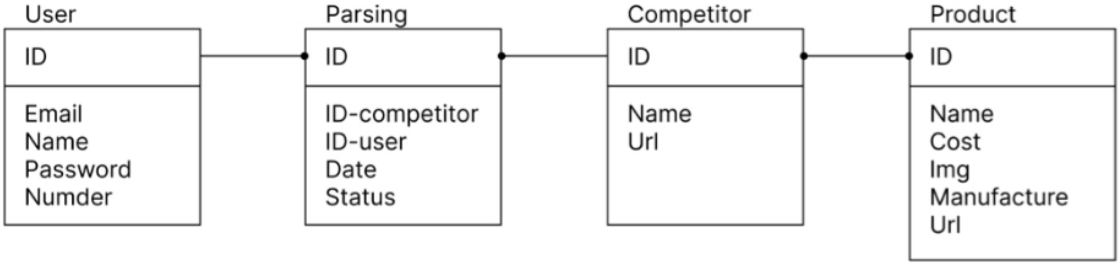


Рис. 6. Структура модели БД

Так как логика представления у компонента Template организована в виде сгенерированной разметки html, определяются шаблоны, которые будут задействованы при реализации прототипа (рис. 7).

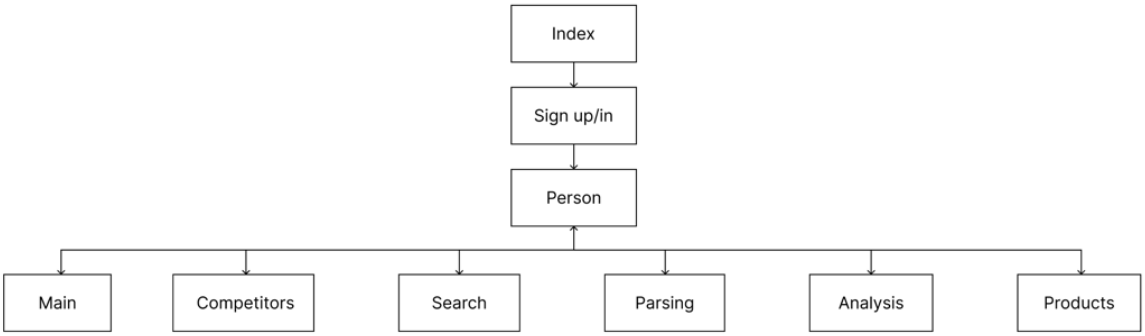


Рис. 7. Структура html-страниц

Макет прототипа программы разработан в графическом редакторе Figma. Пример первых трех страниц представлен на рис. 8.

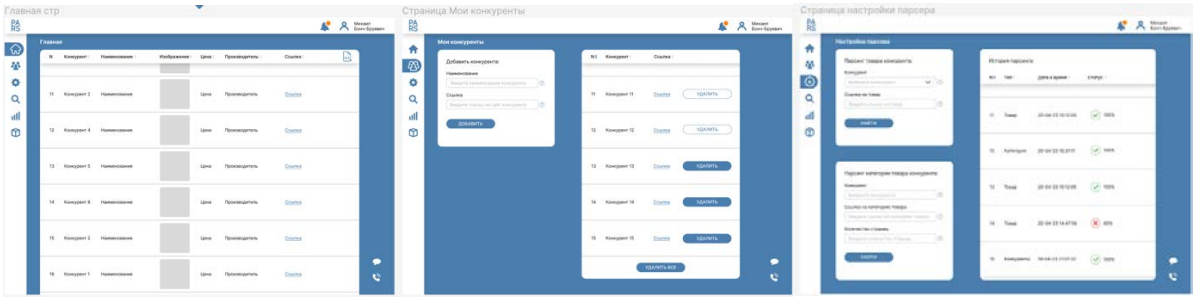


Рис. 8. Макет прототипа программы

Таким образом, разработана модель парсинга данных, позволяющая снизить затрачиваемое время на выполнение процессов поиска и сбора информации о конкурентах за счет их автоматизации и внедрения дополнительного модуля. Также, разработанная модель упрощает процесс взаимодействия пользователя с интерфейсом программы-парсера за счет продуманного пользовательского интерфейса.

Список использованных источников

1. Data Insight Маркетинговое исследование «Интернет-торговля в России 2022» [Электронный ресурс]. URL: https://datainsight.ru/eCommerce_2022 (дата обращения 04.05.2023).

2. Колесов А. Д., Бедердинова О. И. Разработка парсинг-системы бизнес-аналитики // Лучшие студенческие исследования: сборник статей III Межд. научно-исслед. конкурса, Пенза, 10 марта 2021 г. Пенза : Наука и Просвещение (ИП Гуляев Г. Ю.), 2021. С. 57–60.

3. Якимов П. В., Епифанов А. С., Галибин С. В. Разработка веб-сервиса анализа цен интернет-магазинов конкурентов // Информатика и вычислительная техника: сборник научных трудов, Чебоксары, 2021 г. Чебоксары : ЧГУ имени И. Н. Ульянова, 2021. С. 309–313.

4. Сухова А. П., Фирсов А. В. Разработка приложения для мониторинга цен на книги в интернет-магазинах: решение задач при парсинге сложных сайтов // ИНТЕКС-2021 : сборник материалов Всерос. науч. конф. молодых исследователей с междун. участием. Том Ч. 4. Москва, 12–15 апреля 2021 г. М. : ФГБОУ ВО «РГУ им. А. Н. Косыгина (Технологии. Дизайн. Искусство)», 2021. С. 50–53.

5. Евсеев А. С., Латыпов Р. Р., Постолит Е. А., Синельникова-Мурылева Е. С. Технические и методологические проблемы сбора данных о ценах онлайн-ритейлеров // Экономическое развитие России. 2022. Т. 29. № 11. С. 36–45.

Статья представлена научным руководителем, доцентом кафедры ИКД СПбГУТ, кандидатом педагогических наук, доцентом Гуниной Е. В.

УДК 656.022.82

А. И. Ширин (студент гр. ИСТ-212м, СПбГУТ)

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ СИСТЕМЫ СОПРОВОЖДЕНИЯ ТРАНСПОРТНОЙ ЛОГИСТИКИ МАСШТАБА МЕГАПОЛИСА

Показана высокая востребованность информационных логистических систем. Актуализирована разработка логистического приложения для контроля грузоперевозок и доставок. Рассмотрены принципы организации и функциональные возможности систем сопровождения транспортной логистики. Раскрыта концепция декомпозиции задачи обработки и интегрирования информации в системах сопровождения транспортной логистики.

мегаполис, транспорт, логистика, автоматизация, сопровождение, информационная система.

В современном мире предприятие любого масштаба имеет свою централизованную систему. Транспортная логистика – одна из ключевых составляющих такой системы. Задача логистики заключается в сопровождении товара с конвейера до получателя, от производителя до точки сбыта товара. Перед логистической системой ставятся условия: товар должен быть доставлен с минимальными затратами в установленные временные рамки в нужном месте и с надлежащим качеством [1]. При выполнении всех условий задача логистической системы считается достигнутой.

Система сопровождения в транспортной логистике (ССТЛ) – информационная система, которая решает задачи транспортной логистики, объединяя условия работы логистических предприятий и компаний в одно целое [2]. Благодаря развитию информационных технологий работа транспортной логистики стала значительно быстрее и эффективнее, так как специализированные логистические информационные системы позволяют наблюдать за доставкой заказов в реальном времени. Они являются связующим звеном в отношениях между клиентом (заказчиком), менеджером, логистом, исполнителем (перевозчиком) и получателем. Основной причиной модернизации подобных систем служит повышение производительности перевозки грузов.

Каждая ССТЛ уникальна, но принципы организации и функциональные возможности у них схожи.

Главный принцип работы ССТЛ грузовой и коммерческой работы при осуществлении логистического процесса состоит в том, что с момента поступления заказа до момента его выдачи вся необходимая для работы информация хранится в памяти компьютера. Процесс перемещения товаров

фиксируется и корректируется движением сведений о его доставке по структурам данных в привязке к исполнителю, получателю и распределителю. Таким образом, образуется глобальная динамическая информационная система.

Для надёжной работы при разработке ССТЛ требуется соблюсти определенные условия: согласование работы базовых логистических операций, организация управленческого контроля и анализ логистических решений на разных уровнях.

При создании ССТЛ руководствуются следующими принципами:

- доступность: ССТЛ должна быть легкодоступна и понятна для каждого субъекта транспортной операции;
- точность: сведения о логистической операции должны демонстрироваться чётко и без перебоев в режиме реального времени;
- динамичность: ССТЛ должна показывать изменения в заказах;
- своевременность: произошедшее событие в процессе транспортной операции должно быть отражено в ССТЛ в наименьшие сроки;
- гибкость: модель системы сопровождения должна учитывать и предусматривать изменения, которые могут произойти из-за дополнительных требований со стороны исполнителя или заказчика;
- эффективность оформления отчетных данных: интерфейс ССТЛ должен отображать необходимые сведения в удобной и понятной форме для каждого субъекта грузоперевозки;
- поэтапность: возможность постепенного надстраивания иерархической системы для грамотного распределения работы между разными этапами;
- единство: вся необходимая информация хранится на одном сервере, в результате чего не затрачивается время на поиск сведений и установление связи с каждым субъектом транспортной операции;
- принцип единого источника: первичные документы не дублируются и используются многократно для одной или нескольких операций;
- принцип многосторонней выгоды: ССТЛ стремится прогнозировать и строить работу таким образом, чтобы добиться максимально возможного физического финансового и физического комфорта для каждого участника логистической операции;
- соблюдение базовой инфраструктуры ССТЛ.

В настоящее время координирование потоками информации имеет электронных характер, т. е. участники логистической цепочки выбирают безбумажный способ хранения информации. Несмотря на попытки всё больше автоматизировать и упрощать документы, регулировать тарифы грузоперевозки и финансовые расчеты, сохраняется устаревшая технология транспортно-логистического процесса.

Переход к применению информационных систем сопровождения ССТЛ в логистических процессах обеспечит упрощение транспортных операций, облегчение процессов погрузки, доставки и выдачи грузов, сокращение ручного труда, снижение нужды в документировании информации на бумажных носителях, слежение за доставкой груза динамично, точно, своевременно и доступно для каждого участника транспортной операции.

В соответствии с назначением ССТЛ выделяется пять уровней обработки и интегрирования информации (рис. 1).

На нижнем уровне система сопровождения обслуживает требуемые операции. Именно на нём обрабатывается информация, представляющая приём заказа, распределение ресурсов, сведения о грузах, отправку и доставку заказов, отправку заказчику сведений о процессе выполнения логистической операции. Весь цикл операции регулируется с помощью оперативной и своевременной информации. Одновременно происходит много процессов. ССТЛ обрабатывает данные, полученные из разных источников, так что производительность системы сопровождения на данном этапе играет ключевую роль, определяющую производительность, скорость обработки информации.

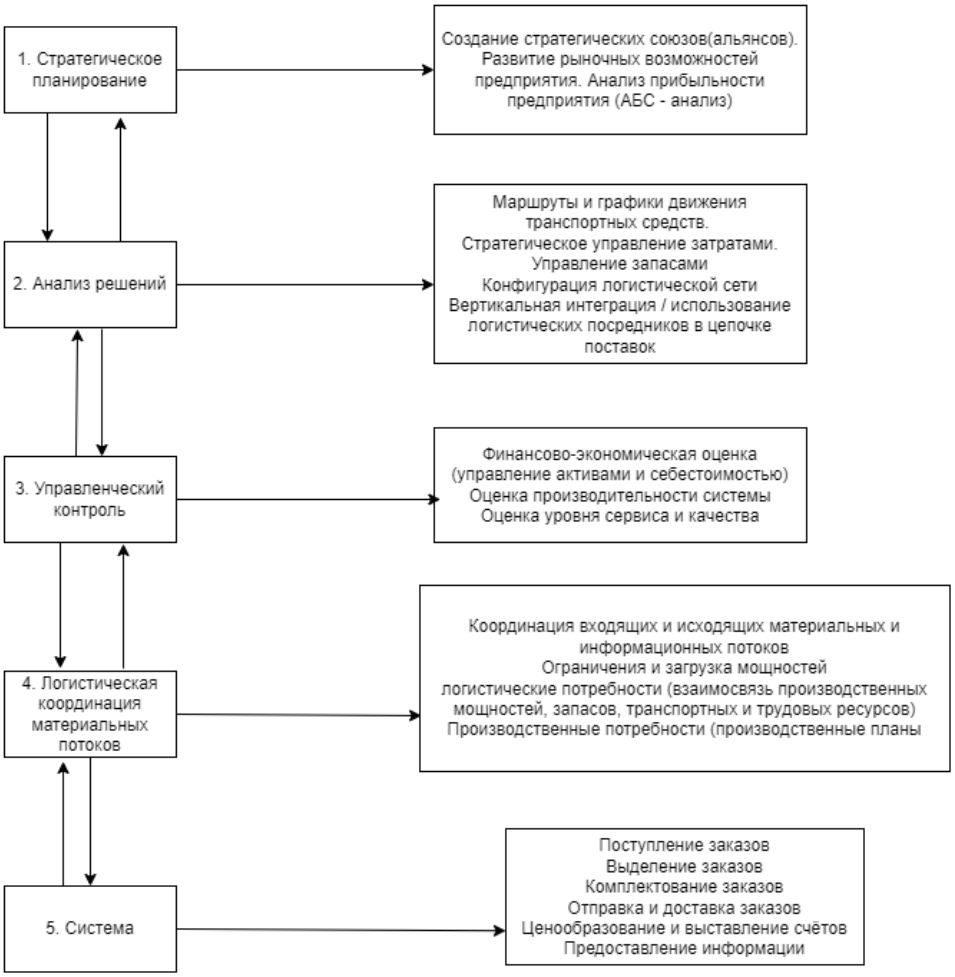


Рис. 1. Уровни системы сопровождения транспортной логистики

На втором уровне входящие и исходящие данные координируются. Система анализирует транспортные возможности, количество ресурсов, физические и финансовые возможности компании, потребности производства.

На третьем уровне осуществляется управленческий контроль. На нём оцениваются результаты работы ССТЛ. Обработанные данные служат базой для анализа экономической эффективности, качества предоставляемой грузоперевозки и пропускной способности системы.

На четвертом уровне происходит уже анализ последствий для транспортной системы. Программа учитывает маршрут движения, затраты на исполнение заказа, управление ресурсами, товарами, рациональность выбранной схемы работы логистической сети, использование посредников в схеме логистического процесса. Четвертый уровень способствует корректировке программы в любой момент времени. Он координирует работу транспортной операции и интегрирует намеченные на предыдущих этапах планы.

Самая важная роль четвертого уровня – это «переключение» режима ускорения или замедления логистической операции и «слежение» за достижением поставленной цели.

Пятый уровень осуществляет информационную поддержку по моделированию и корректировке транспортной операции. Управленческие решения носят долговременный характер, они связаны с развитием возможностей рынка, созданием стратегических альянсов, стратегией управления логистической системы. Для стратегического планирования требуются сведения, которые поступают от четвертого уровня. Эта информация поступает через фильтр «Анализ решений» и отражается в хозяйственных планах после того, как будут сопоставлены альтернативные варианты стратегий доставки товаров.

Для преодоления априорной неопределенности относительно возможных изменений в окружающей среде при сопровождении транспортной логистики может применяться генетический подход [3].

Автоматизация информационно-логистических процессов помогает создавать и поддерживать высокий уровень функционирования системы. Благодаря автоматизации потребители обладают полной и точной информацией на комфортном интерфейсе и в удобное время. Автоматизация процессов, входящих в транспортно-логистические операции, это ключевая особенность ССТЛ.

Список используемых источников

1. Кузменко Ю. Г., Варганова М. И. О внедрении принципов финансовой логистики в деятельность современных российских предприятий // Пучков В. Р., Ермоленко В. В. Управление логистическими процессами компании: инновации в логистике (на примере ООО «Кубань Транс Авто») // Управление инновационной экосистемой региона и коммерциализацией нововведений. Организационное, аналитическое и информационно-документационное сопровождение деятельности инфраструктуры : сборник трудов V Всероссийской (национальной) научно-практическая конференции с международным

участием, Краснодар, 24–29 мая 2021 года / Отв. редактор В. В. Ермоленко. Краснодар : Кубанский государственный университет, 2021. С. 184–192.

2. Гоголин С. С., Фаддеева Е. Ю. Повышение эффективности принятия управленческих решений в логистике на основе применения информационно-аналитических систем // Вестник Академии знаний. 2018. № 29(6). С. 111–117.

3. Коберси И. С., Шкуркин Д. В. Применение генетических алгоритмов для оптимизации транспортных задач // Известия ЮФУ. Технические науки. 2012. № 2(127). С. 172–176.

Статья представлена научным руководителем,
доктором технических наук, профессором Птицыной Л. К.

УДК 681.518.5

М. А. Шохина (студентка гр. ИСТ-942, СПбГУТ)

РАЗРАБОТКА БЛОКА УПРАВЛЕНИЯ НА БАЗЕ ПЛИС ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ОСВЕЩЕНИЯ ПОМЕЩЕНИЙ ПРЕДПРИЯТИЙ

В статье рассматриваются вопросы, связанные с разработкой блока управления системой освещения на базе программируемой логики, предназначенного для контроля системы освещения различных помещений предприятий, в том числе, автоматизированных рабочих мест сотрудников. Обосновывается актуальность разработки подобного класса устройств. Представлены структурные схемы блока управления и всей интеллектуальной системы освещения. Предложены алгоритмы работы блока управления в зависимости от типа помещения. Разработаны модуль питания блока и тракт первичной обработки сигналов от датчиков с непрерывной функцией преобразования. Приведены результаты моделирования блока в среде Multisim и предложен вариант его конструкции.

интеллектуальная система, освещение промышленных помещений, датчик, алгоритм управления, структурная схема.

Эффективным способом снижения расходов электроэнергии промышленными предприятиями является минимизация ее затрат на освещение помещений [1]. С этой целью осуществляется внедрение соответствующих автоматизированных систем, позволяющих оптимизировать расход электроэнергии на предприятии, а также обеспечить оптимальную интенсивность освещения автоматизированных рабочих мест (АРМ) сотрудников. Эти системы могут учитывать величину естественного освещения (окна помещений), график работы специалистов (работа или обед), естественный износ источников освещения (снижение светоотдачи или выход из строя) и присутствие людей.

В настоящее время существует множество подобных интеллектуальных систем, позволяющих: собирать данные с электросчетчиков, автоматически изменять освещенность по данным датчиков движения для контроля присутствия и передвижения людей, управлять освещением с учетом сменности и графика работы, управлять освещением в зависимости от уровня естественного света, интегрировать систему управления освещением с охранной системой [2].

Анализ существующих решений позволил выявить следующие их преимущества и недостатки. К преимуществам следует отнести: наличие ПО для визуализации работы системы и её быстрой диагностики, гибкость настройки. В качестве основных недостатков следует отметить: громоздкость, сложность структуры и настройки, относительная дороговизна эксплуатации для предприятий «средней» и «малой» величины.

Для устранения указанных недостатков предлагается система (рис. 1), подходящая для небольших предприятий и организаций. В основе такой системы находится блок управления, который может быть построен не на дорогих микропроцессорах, а на ПЛИС или микроконтроллере.

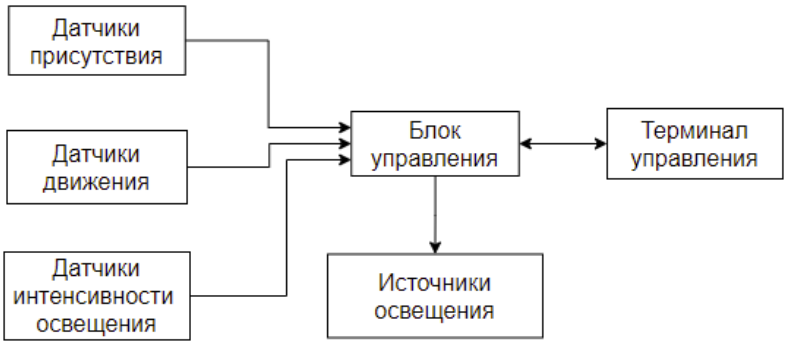


Рис. 1. Структурная схема интеллектуальной системы освещения промышленных помещений

В качестве сенсоров используются датчики присутствия (контактные, релейные) и движения (инфракрасные, ультразвуковые), которые необходимы для фиксирования наличия персонала в помещения и в зависимости от этого, автоматического управления системой освещения. Для определения уровня естественного освещения используются датчики интенсивности освещения (фотодиоды и фоторезисторы). Блок управления – это устройство, которое посылает управляющий сигнал для регулировки освещения, на основе данных с датчиков. Терминал управления позволяет дистанционно управлять системой освещения, настройки расписания её работы, следить за расходом электроэнергии и исправностью компонентов посредством специального программного обеспечения.

Рассмотрим более подробно структуру блока управления (рис. 2). На вход портов ввода/вывода поступает сигнал от датчиков, который, проходя через тракт предварительной обработки сигнала (ТПО), поступает

на входное согласующее устройство (ВхСУ) (компаратор или параллельный АЦП) и далее на модуль управления (МУ) и может выводиться на вспомогательные внешние устройства. Также сигнал управления с МУ поступает на выходное согласующее устройство (ВыхСУ) (преобразователь уровня или ЦАП) и по каналам передачи данных на исполнительные устройства (ИУ) (ключ управления симмистором или ШИМ-регулятор), которые управляют источниками освещения (ИО).

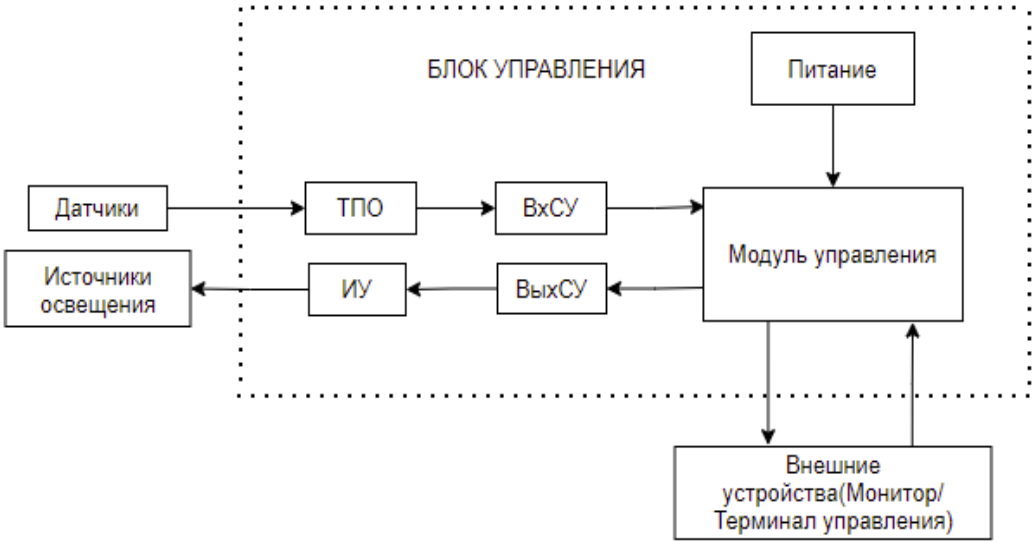


Рис. 2. Структурная схема блока управления системой освещения промышленных помещений

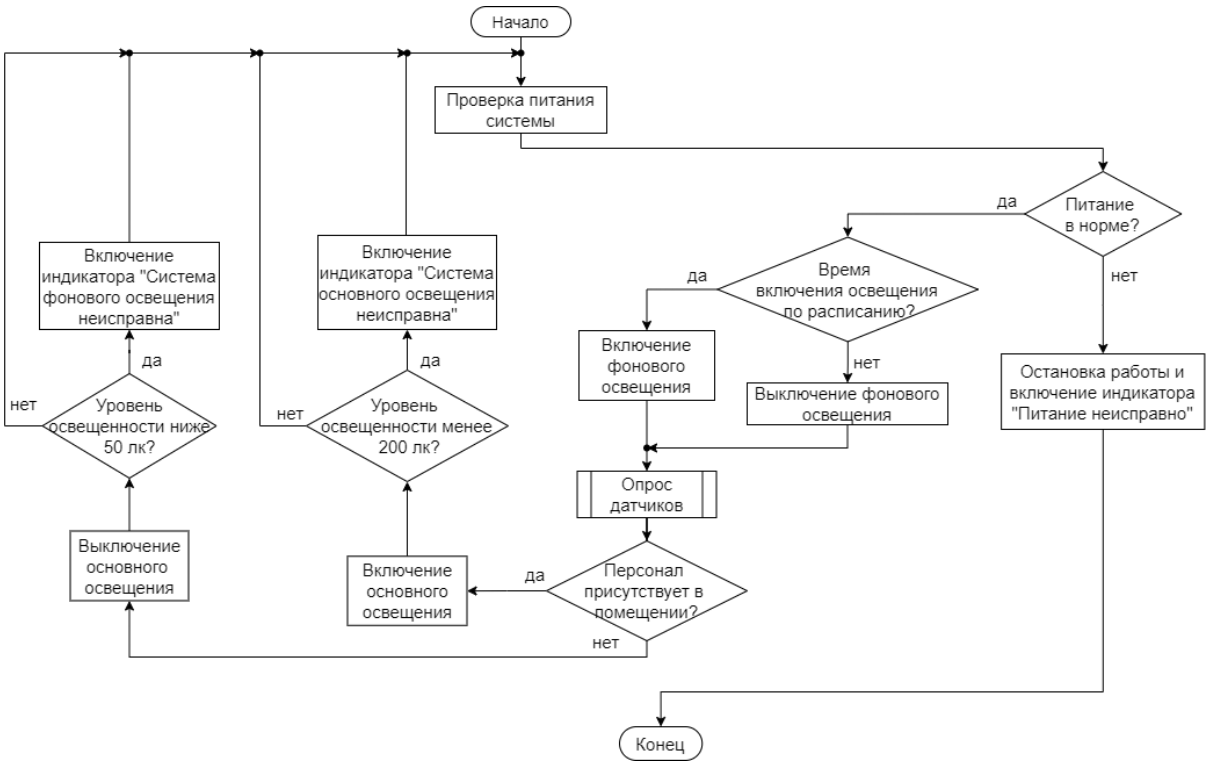


Рис. 3. Алгоритм управления освещением на склад

Алгоритм работы системы управления освещением рассмотрим на примере склада (рис. 3). Сначала осуществляется внутренняя проверка системы. Если в системе отсутствует питание, то включается индикатор «питание неисправно», иначе система включается или продолжает свою работу. Затем, если по расписанию задано время работы системы освещения, включается фоновое освещение и производится последовательный опрос датчиков присутствия и интенсивности освещенности. В случае срабатывания датчика движения, включается основное освещение. Если уровень освещенности при фоновом или основном освещении ниже заданных границ (50 лк и 200 лк соответственно), то подаются соответствующие сигнал о неисправности системы.

Тракт предварительной обработки (ТПО) сигнала предназначен для максимизации соотношения сигнал-шум и нормирования динамического диапазона данного сигнала, поступающего на АЦП, а в дальнейшем на МУ.

Структурная схема ТПО представлена на рис. 4. Она состоит из трансимпедансного усилителя (усиления тока с фотодиода), имитатора помехи (для имитирования работы реального датчика), дифференциального усилителя (для удаления статической помехи), фильтров (для удаления помех) и порогового устройства (для преобразования аналогового сигнала в цифровую информацию).

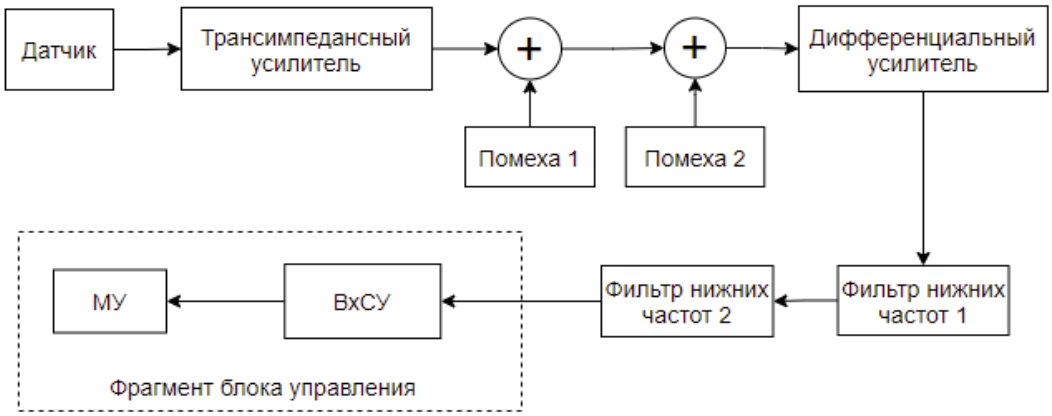


Рис. 4. Структурная схема тракта обработки данных

Для фильтрации сигнала от помех необходимо использовать фильтр нижних частот, в среде Multisim было произведено моделирование процесса фильтрации сигнала (рис. 5).

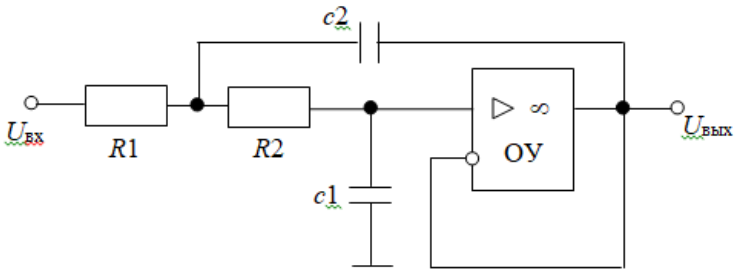


Рис. 5. Схема фильтра нижних частот

Так как частота полезного сигнала не превышает частоты 10 Гц, то с помощью ФНЧ необходимо ослабить все частоты выше этого значения.

Для расчета значения частоты среза используется следующая формула:

$$f_{\text{в}} = \frac{1}{2\pi RC}.$$

Подставив $f_{\text{в}} = 10$ Гц получим:

$$f_{\text{в}} = \frac{1}{2\pi RC} = \frac{1}{2*3{,}14*16*10^3*10^{-6}} \approx 10 \text{ Гц}.$$

На рис 6. приведены АЧХ и ФЧХ двухкаскадного активного ФНЧ 2-го порядка.

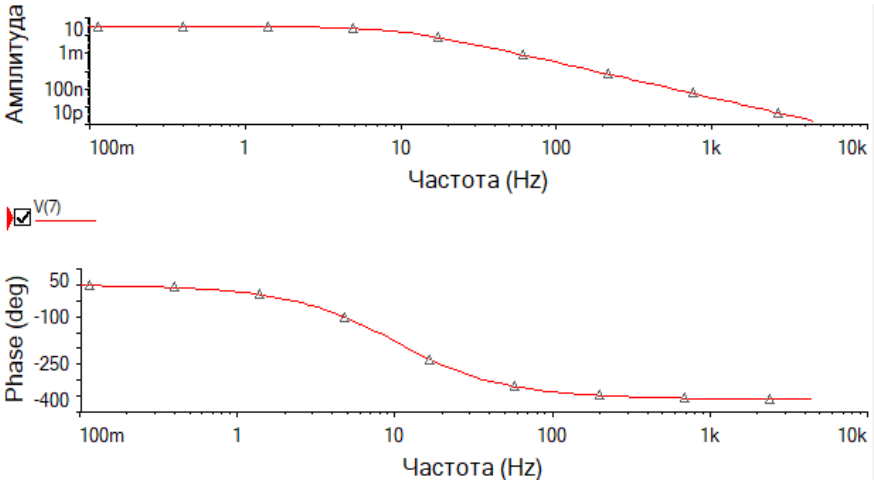


Рис. 6. АЧХ и ФЧХ двухкаскадного активного ФНЧ 2-го порядка

Коэффициенты передачи фильтра при частоте 10 Гц и при частотах помех 100 Гц и 150 кГц (табл. 1).

ТАБЛИЦА 1. Коэффициенты передачи при разных частотах

Частота сигнала	К передачи	Амплитуда, В
5 Гц	≈ 1	0,59
100 Гц	0,0001	0,03
150 кГц	$<10^{-9}$	0,015

Конструкция корпуса блока управления может представлять собой шкаф управления. Пример расположения элементов блока управления (ПЛИС, блок питания, индикаторы) в корпусе изображен на рис. 7.

Конструкция блока должна предусматривать наличие индикаторов для отображения работоспособности систем освещения. Например, в случае штатной работы индикатор «Питание» будет гореть зеленым. В случае проблем с питанием системы или неисправности светильников, соответствующие индикаторы будут гореть красным.

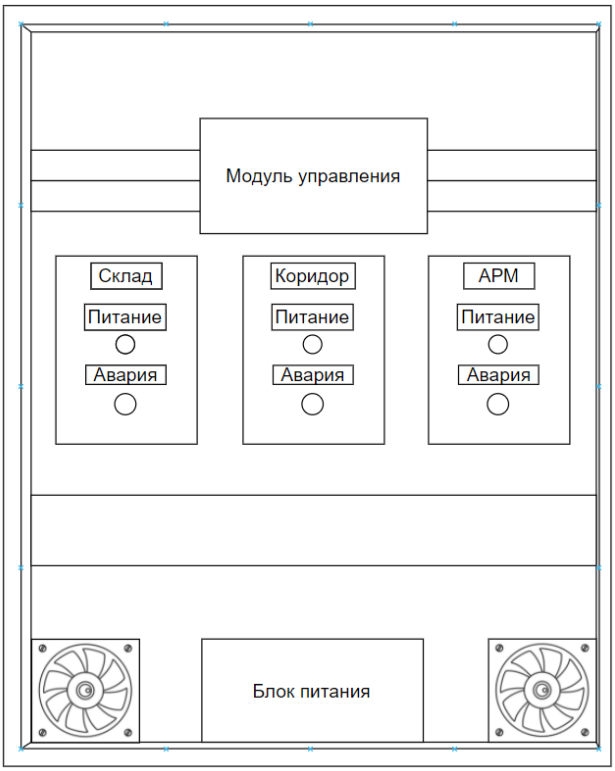


Рис. 7. Расположение элементов блока управления в корпусе

Представленная система может использоваться непосредственно для освещения промышленных помещений, а при определенной адаптации являться элементом «умного» дома или «умной» теплицы.

В статье представлена структурная схема интеллектуальной системы освещения промышленных помещений, а также обосновывается выбор современной элементной базы для ее реализации. Разработан алгоритм работы системы, учитывающий наличие персонала на рабочем месте, а также уровень искусственного освещения.

Список используемых источников

1. ФЗ от 23 ноября 2009 г. N 261-ФЗ «Об энергосбережении и о повышении энергетической эффективности и о внесении изменений в отдельные законодательные акты Российской Федерации».

2. Системы автоматического управления освещением зданий. URL: http://electricalschool.info/main/lighting/409-sistemy-avtomaticheskogo-upravlenija.html (дата обращения 22.03.2023).

Статья представлена заведующей кафедрой ИСАУ СПбГУТ, доктором технических наук, доцентом Верховой Г. В.

УДК 004.891

Ф. Э. Шульгин (студент гр. ИСТ-212м, СПбГУТ)

МЕТОДЫ ОПТИМИЗАЦИИ ГРАФИКА ДВИЖЕНИЯ ТРАНСПОРТА

Оптимизация трафика движения транспортных средств является сложной проблемой, которая требует многодисциплинарного подхода и продолжительных исследований. На данном этапе развития информационных технологий, имеется возможность оптимизировать решение части задач с применением инженерных методов, а также с помощью формального математического подхода.

Описаны методы работы инструментов, таких как симуляционные модели, анализ сетей, оптимизация сигналов светофоров и системы интеллектуальных транспортных систем. Исследованы проблемы, с которыми сталкиваются разработчики данных систем, и возможные пути их решения.

оптимизация движения, симуляционные модели, АСУДД, графы в построении путей.

Для оптимизации графика движения транспорта существуют три основные задачи, которые решаются повсеместно. Во-первых, необходимо определить места, где можно собирать данные о движении транспорта. Во-вторых, нужно разработать методы сбора данных, которые позволят получать информацию о транспортном потоке в режиме реального времени. В-третьих, требуется разработать алгоритмы и методы использования этих данных для оптимизации графика движения транспорта [1].

Для решения первой проблемы, часто используют графы. Графы могут помочь в определении критически важных узлов и ребер, которые необходимо учитывать при сборе данных. Например, для оптимизации графика движения транспорта можно использовать граф дорожной сети, который показывает связи между различными улицами, дорогами и перекрестками. С помощью графа можно выделить критически важные узлы, такие как перекрестки с большим потоком транспорта или участки дорог, которые часто приводят к пробкам.

Другой пример использования графов – это определение наиболее эффективного маршрута для сбора данных о транспорте. Пример такого графа изображен на рис. 1. Граф может помочь в выявлении критических узлов, которые необходимо охватить при сборе данных. Таким образом, графы позволяют более эффективно распределять ресурсы для сбора данных и улучшать процесс оптимизации движения [2].

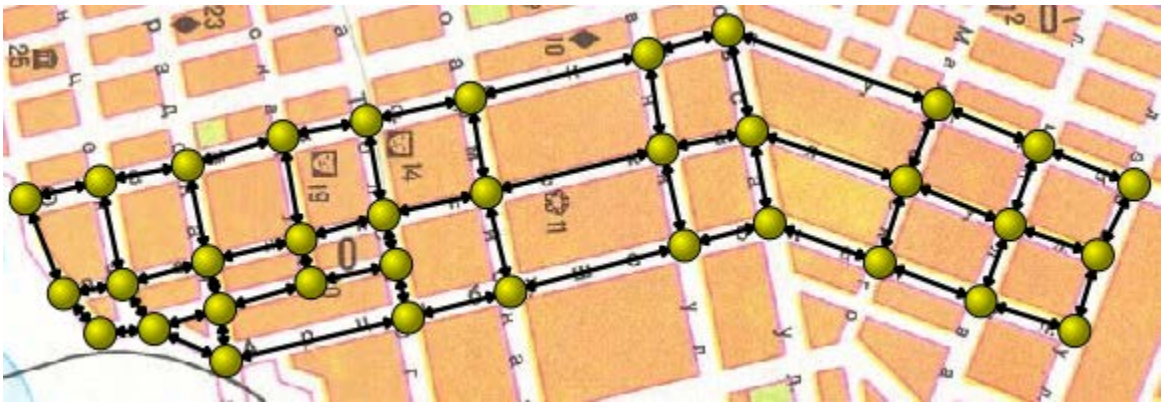


Рис. 1. Наложение графа на дороги карты города

Решение второй проблемы – «как собирать данные», напрямую зависит от решения первой проблемы. Установка и настройка оборудования для сбора данных может требовать значительных затрат, поэтому решение первой проблемы может помочь оптимизировать использование бюджета.

Если данные о движении транспорта собираются не эффективно, то оптимизация движения становится сложной задачей. Правильно установленное оборудование для сбора данных позволяет собирать информацию о движении транспорта более точно и эффективно.

Кроме того, решение первой проблемы может частично оптимизировать использование бюджета. Если процесс сбора данных оптимизирован, то можно сократить затраты на установку и настройку оборудования. Это позволяет использовать бюджет более эффективно и выделить дополнительные ресурсы на другие задачи, связанные с оптимизацией движения транспорта.

Существует несколько способов получения данных о движении транспорта. Один из них – это установка датчиков на дорогах или камер, которые могут использовать компьютерное зрение для сбора информации о движении. Кроме того, можно использовать GPS-навигацию (или ГЛОНАСС), wi-fi и bluetooth сигналы или собирать данные от пользователей мобильных приложений.

Данные о движении транспорта, собранные с помощью датчиков на светофорах или камер, установленных на дорогах, могут быть использованы для анализа движения и оптимизации графика движения. Для сбора данных в режиме реального времени можно использовать специализированные системы, такие как системы транспортного мониторинга. Эти системы могут собирать данные о движении транспорта с помощью GPS-навигации, датчиков на дорогах и других источников.

В настоящее время часто применяются комплексные решения, такие как АСУДД (автоматизированная система управления дорожным движением), которые объединяют в себе различные методы сбора и анализа данных о движении транспорта. АСУДД может использоваться для монито-

ринга и управления светофорами, контроля скорости движения, регулирования потока транспорта и других задач. Она собирает данные о движении транспорта с помощью GPS-навигации, датчиков на дорогах и других источников, а затем анализирует эти данные для принятия решений об оптимизации движения. С помощью таких комплексных решений как АСУДД происходит оптимизация движения транспорта. На рис. 2 показаны основные источники генерации данных для АСУДД, устройства благодаря которым эти данные собираются, а также устройства контроля движения [3].

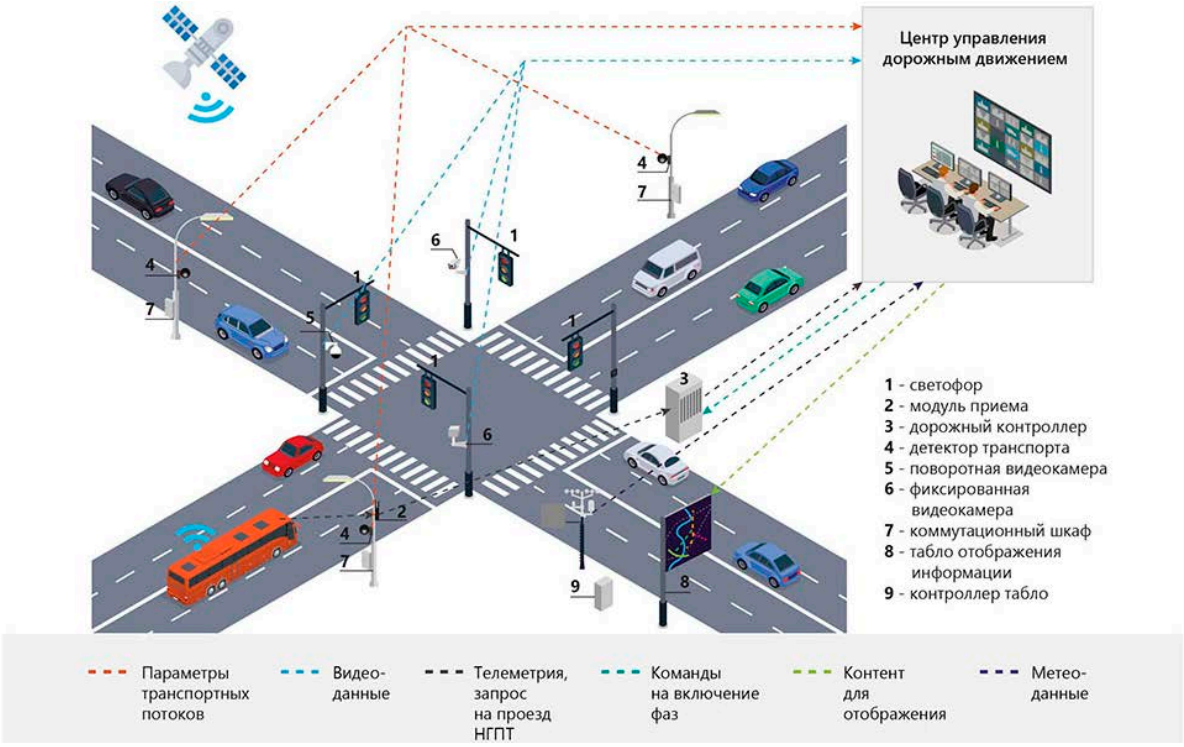


Рис. 2. Агенты и генерируемые данные АСУДД

Для использования данных о движении транспорта для оптимизации графика движения можно применять различные методы, включая математическое моделирование, алгоритмы оптимизации и машинное обучение [4].

Существует несколько методов оптимизации, которые могут быть использованы для решения задач оптимизации графика движения транспорта. Например, можно использовать методы генетической оптимизации, муравьиный алгоритм или нейронные сети [5].

Рассмотрим шаги по оптимизации графика движения транспорта на примере алгоритма генетической оптимизации.

1) *Определение целевой функции*. Целевая функция должна отражать то, что мы хотим оптимизировать в трафике в городе. Например, мы можем стремиться к минимизации общего времени задержек, сокращению средней скорости движения или минимизации эмиссий выбросов.

2) *Определение параметров*. Определяются параметры, которые будут управлять алгоритмом генетической оптимизации. Эти параметры включают размер популяции (N), количество поколений (G), вероятности скрещивания (P_c) и мутации (P_m), а также другие специфические параметры, связанные с трафиком в городе (например, максимальная скорость, расстояние между автомобилями и другие).

3) *Инициализация популяции*. Начальная популяция представляет собой набор случайно созданных индивидов, где каждый индивид представляет собой комбинацию параметров, определяющих трафик в городе.

4) *Оценка приспособленности*. Каждый индивид в популяции оценивается с использованием целевой функции. Для оптимизации трафика в городе можно использовать следующую формулу для расчета приспособленности (*fitness*) индивида:

$$fitness = 1 / (target_function_value + epsilon),$$

где *target_function_value* – значение целевой функции для данного индивида, *epsilon* – малое положительное число, которое предотвращает деление на ноль. Чем меньше значение целевой функции, тем выше приспособленность индивида.

5) *Селекция*. Выбираются наиболее приспособленные индивиды из популяции для создания следующего поколения. Один из популярных методов селекции – пропорциональная селекция. Приспособленность каждого индивида используется для расчета вероятности выбора данного индивида.

6) *Скрещивание*. Выбранные индивиды совершают операцию скрещивания, в результате которой создаются новые потомки. Метод скрещивания может варьироваться в зависимости от специфики задачи и представления индивидов.

7) *Мутация*. С некоторой вероятностью P_m происходит мутация в потомке. Мутация вносит случайное изменение в генетический материал, что позволяет исследовать новые решения.

8) *Создание нового поколения*. После скрещивания и мутации создается новое поколение индивидов, которые заменяют предыдущее поколение. Этот процесс повторяется до достижения заданного количества поколений G.

9) *Возврат к шагу* 4. Новое поколение оценивается с использованием целевой функции, и процесс селекции, скрещивания и мутации повторяется до достижения условия останова (например, определенное количество поколений или достижение определенного значения приспособленности).

Обычно для задачи оптимизации используют одновременно несколько различных подходов, например – симуляционную модель. В случае с оптимизацией графика движения транспорта – это будет модель, которая отражает дорожную сеть, транспортные средства, пассажиров и другие факторы,

влияющие на движение транспорта. Модель может быть агентно-ориентированной, где каждое транспортное средство и пассажир представлены отдельным агентом, или быть основанной на дискретных событиях [6].

Таким образом, происходит оптимизация переданных параметров в созданную функцию. Если модель работает исправно, возможна «полевая» корректировка движения транспорта – установка или замена дорожных знаков, строительство или реконструкция дорожного полотна, повышенный контроль движения.

Список используемых источников

1. Алексеев А. И., Калашников А. В. Моделирование и оптимизация графика работы автомобильного транспорта на основе информационных технологий // Вестник Московского государственного технического университета им. Н. Э. Баумана. Серия: Машиностроение. 2018. № 6. С. 70–84.

2. Вершинина А. А. Методы оптимизации работы грузоперевозок в городских условиях // Вестник Московского университета. Серия 15: Водный транспорт. 2019. № 2 (58). С. 26–35.

3. Гуляев А. А., Чистякова Н. С. Оптимизация графика работы грузового автомобильного транспорта при использовании системы ГЛОНАСС // Материалы XXIV Международной научно-практической конференции «Транспорт России». М., 2017. С. 219–223.

4. Загородний А. С. Разработка методики оптимизации графика работы транспорта на основе данных мониторинга за движением транспортных средств // Информационные технологии в транспорте. 2018. № 2. С. 31–36.

5. Кораблев В. И., Шубин И. В. Оптимизация графика движения транспорта в режиме реального времени // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2017. № 2 (52). С. 46–54.

6. Никульшина И. М., Козлова О. В. Методика оптимизации графика работы грузоперевозок на основе генетического алгоритма // Автомобильный транспорт.

7. Птицына Л. К., Лебедева А. А., Птицын А. В. Расширение знаний о качестве функционирования интеллектуальных информационных агентов // Наука. Информатизация. Технологии. Образование : материалы XI международной научно-практической конференции, Екатеринбург, 26 февраля –2018 года. Екатеринбург : Российский государственный профессионально-педагогический университет, 2018. С. 577–583.

Статья представлена научным руководителем, доцентом кафедры ИУС СПбГУТ, кандидатом технических наук, старшим научным сотрудником Филипповым Ф. В.

УДК 004.9:332.872.2

А. О. Жаранова (ассистент кафедры ИУС, СПбГУТ)
Д. И. Яремко (студент гр. ИСТ-012, СПбГУТ)

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО АРЕНДЕ КВАРТИР

В данной статье представлен проект информационной системы для удобного и эффективного поиска и аренды недвижимости. Проведен анализ существующих систем, определены их достоинства и недостатки в части поиска потенциальных квартиросъёмщиков для арендодателей. Сформирован единый образ комплексной системы, объединяющей весь функционал, необходимый для обеспечения полного цикла аренды жилья. Разработанная система позволит пользователям удобно и быстро находить подходящие объявления, настраивать свои профили, взаимодействовать с другими пользователями через чат, оставлять отзывы и осуществлять онлайн-оплату.

информационная система, арендатор, недвижимость, аренда, проектирование, UML.

В настоящее время владельцы жилья и арендаторы ищут удобные способы управления процессом аренды квартир. Помимо этого, растет спрос на рынке аренды жилья, так как в современном мире многие люди предпочитают аренду вместо покупки [1]. В связи с этим актуализируется вопрос реализации информационной системы для эффективного и удобного поиска и аренды недвижимости.

В большинстве существующих информационных систем арендодатели ограничены возможностями используемого сервиса в подборе арендаторов и вынуждены полагаться на отклики на объявления. Во многих случаях отсутствует удобный функционал для онлайн-оплаты аренды квартиры и залога, что оставляет возможность для мошенничества от одной из сторон сделки. В системах часто применяется модель, в которой агенты взимают высокую комиссию с арендаторов, хотя основные услуги оказываются собственникам недвижимости.

В предлагаемой системе предусмотрено расширение функционала. Модули проектируемой информационной системы включают:

– модуль регистрации и авторизации, позволяющий пользователям зарегистрироваться в системе, создать свой профиль и авторизоваться для доступа к функциональности системы;

– модуль профилей пользователей, позволяющий пользователям создавать и управлять своими профилями, заполнять информацию о себе, добавлять фотографии, указывать контактные данные и другие сведения;

– модуль поиска и фильтрации объявлений, который предоставляет пользователям возможность находить объявления об аренде недвижимости,

используя различные фильтры, такие как цена, тип недвижимости, район, количество комнат и другие параметры;

– модуль отзывов и рейтингов, который позволяет пользователям оставлять отзывы о квартирах, арендодателях и агентах недвижимости и наоборот, а также просматривать рейтинги других пользователей;

– модуль 3D-панорам, который предоставляет пользователям возможность просматривать 3D-панорамы квартир и позволяет более точно представить себе недвижимость, которую они собираются арендовать;

– модуль верификации арендаторов, который позволяет арендодателям проверять личность потенциальных арендаторов;

– модуль онлайн-оплаты аренды и дополнительных услуг, который позволяет пользователям оплачивать аренду и дополнительные услуги онлайн, используя различные способы оплаты, такие как банковские карты, электронные кошельки и т. д.;

– модуль выбора даты и времени для просмотра квартиры и заселения, который позволяет пользователям выбрать удобную для них дату и время для просмотра квартиры и заселения;

– модуль расчета дополнительных расходов, который позволяет пользователям рассчитать дополнительные расходы, связанные с арендой квартиры, такие как коммунальные услуги и интернет;

– модуль чата, который обеспечивает коммуникацию между пользователями и позволяет арендодателям и арендаторам общаться между собой, задавать вопросы, обсуждать детали и договариваться о дате просмотра квартиры;

– модуль проверки кредитной истории, который позволяет арендодателям проверять кредитную историю потенциальных арендаторов, чтобы убедиться в их финансовой состоятельности и платежеспособности;

– модуль интерактивной карты, который позволяет пользователям просматривать доступные для аренды квартиры на карте города, а также видеть информацию о цене, площади, количестве комнат и других характеристиках каждой квартиры;

– модуль умного подбора квартир, который использует алгоритмы машинного обучения, чтобы предложить пользователям наиболее подходящие варианты квартир на основе их предпочтений и запросов;

– модуль онлайн-консультации, который обеспечивает связь пользователей с агентами по недвижимости, которые помогут им в выборе квартиры и ответят на все их вопросы, а в случае успешной сделки арендодатель или агент смогут получать комиссионные;

– модуль оповещений, который позволяет пользователям быть в курсе всех изменений, связанных с арендой и обновлениями, касающимися их объявлений.

Модули проектируемой информационной системы должны взаимодействовать между собой, чтобы обеспечить пользователей полным и удобным функционалом, их взаимодействие представлено на рис. 1.

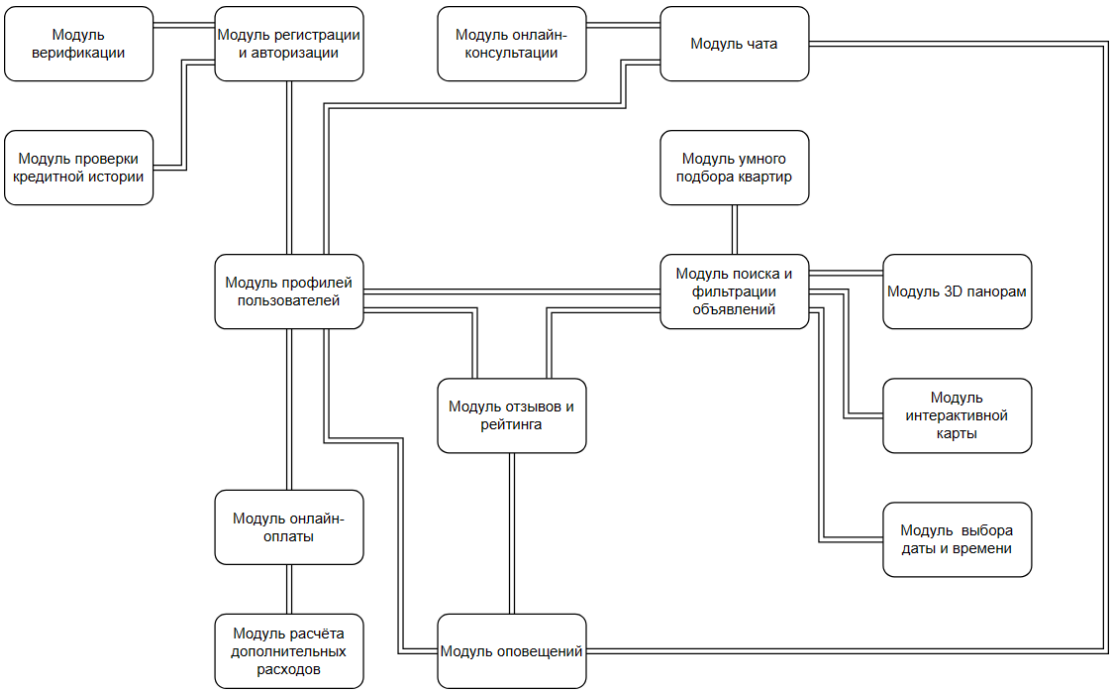


Рис. 1. Модульная схема информационной системы по аренде квартир

Основные функциональные возможности системы представлены на рис. 2 в виде диаграммы прецедентов [2].

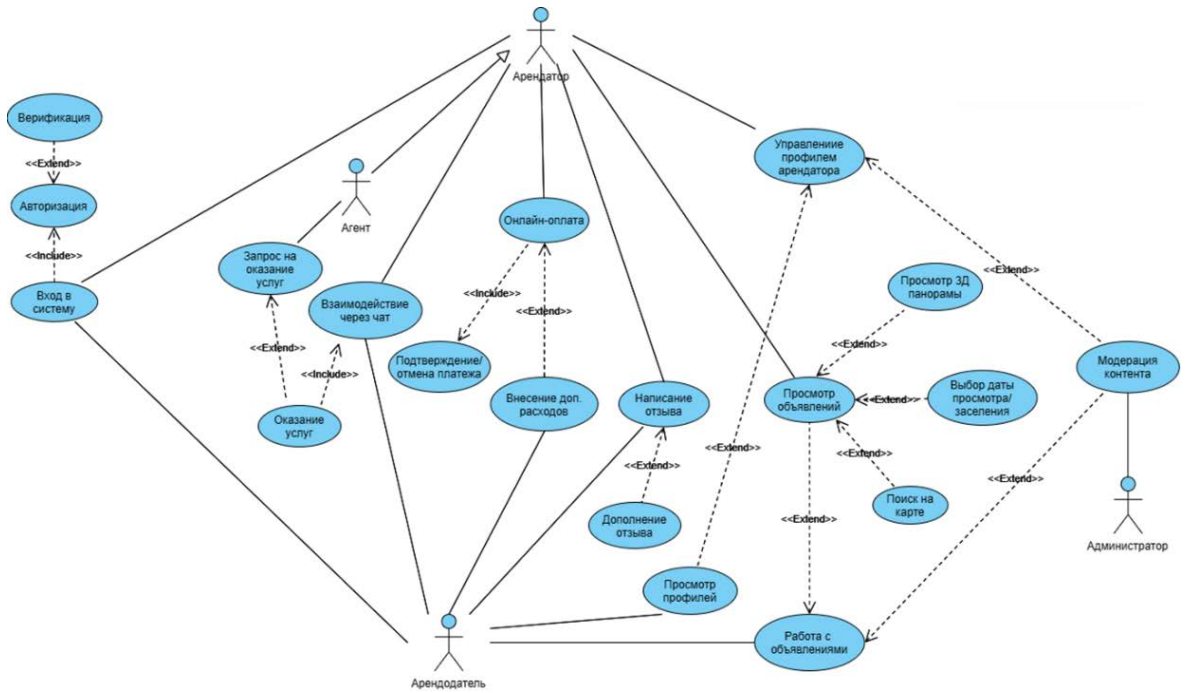


Рис. 2. Общая диаграмма прецедентов информационной системы по аренде квартир

Система предоставляет функционал верификации пользователей, поиска объектов недвижимости на интерактивной карте, написания отзывов с выставлением рейтинга, выбора даты просмотра и заселения в квартиру. Весь контент, загружаемый пользователями системы, модерируется администратором.

На рис. 3 представлена диаграмма последовательности для сценария «Связь с арендатором», которая наглядно демонстрирует действия пользователей в системе.

Сценарий «Связь с арендодателем» имеет схожую структуру со сценарием «Связь с арендатором», за исключением того, что вместо просмотра арендаторов осуществляется поиск квартиры.

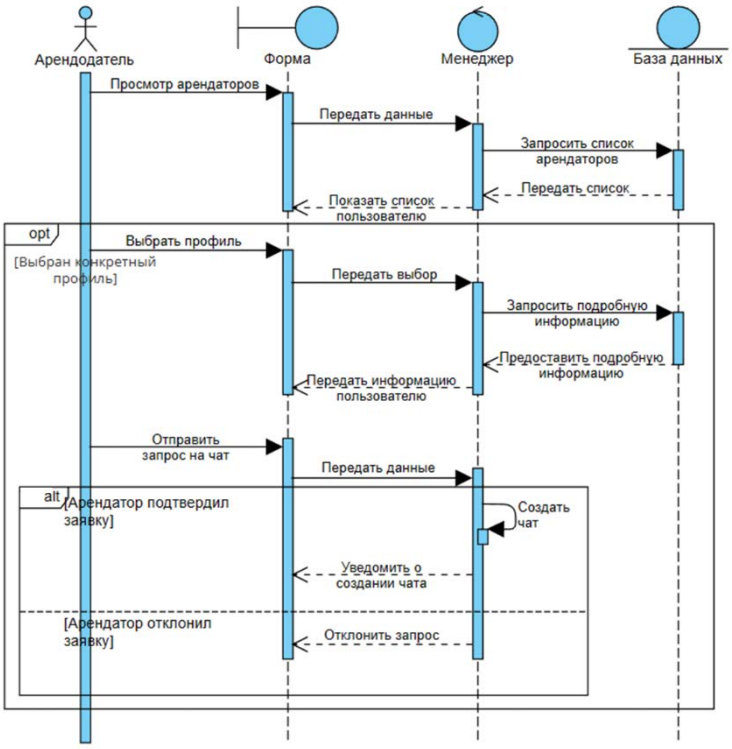


Рис. 3. Диаграмма последовательности сценария «Связь с арендатором»

Для агентов доступен функционал поиска клиентов для онлайн-консультации. Он начинается с отправки агентом запроса на консультацию, после чего ожидается ответ на запрос. Если запрос не подтвержден, агент остается в состоянии «Ожидание ответа на запрос». Если запрос подтвержден, происходит переход в состояние «В процессе консультации», где создается чат, уведомляются пользователи и сделка добавляется в список активных. Затем сценарий продолжается до завершения консультации, где предлагается оценка и написание отзыва, производится расчет комиссионных, а сделка переносится в список завершенных.

Написание отзывов и выставление рейтинга доступно для объявлений о сдаче квартир (рис. 4) и для профилей всех типов пользователей. Для этого

необходимо открыть соответствующее объявление или профиль. Если в системе есть подтверждение завершения сделки, пользователю предоставляется функционал написания отзыва. Все отзывы проходят модерацию перед публикацией.

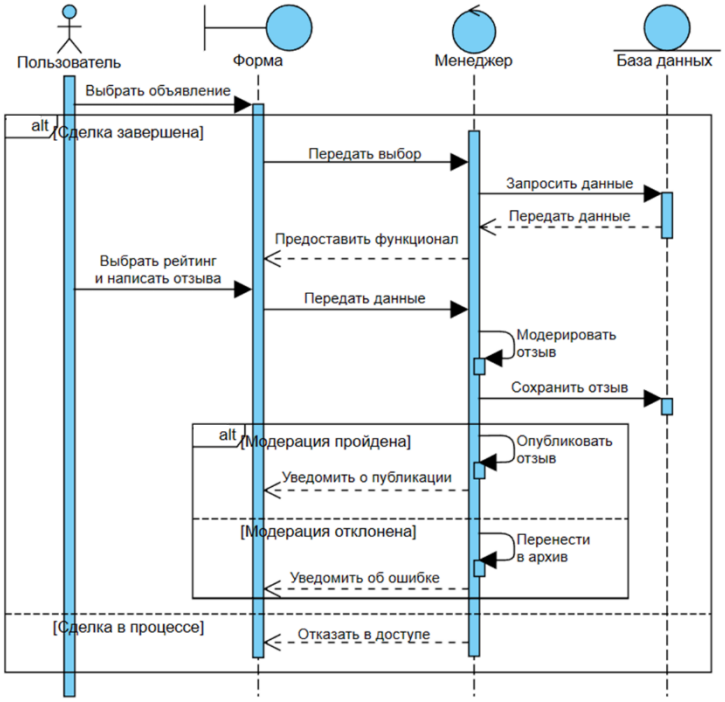


Рис. 4. Диаграмма последовательности написания отзыва

В перспективах развития данного сервиса имеется ряд возможностей для внедрения новых функций и алгоритмов, например, внедрение алгоритмов анализа и рекомендаций. Путем анализа предпочтений и истории поиска пользователей можно предоставлять персонализированные рекомендации по жилью, соответствующие их потребностям. Разработка алгоритмов анализа рекомендуемых цен на квартиры позволит определять выгодность конкретного объявления и предоставлять рекомендации арендодателям. Система может быть расширена функционалом поиска сожителей для совместной аренды.

Список используемых источников

1. Кузнецов Д. А. «Теневая» аренда жилья: проблемы, пути решения // Вестник Академии знаний. 2020. № 41(6). С. 187–191.

2. Котлова М. В. Методы и средства проектирования информационных систем и технологий : учебное пособие. СПб. : СПбГУТ, 2015. 62 с.

Статья представлена доктором технических наук, профессором Птицыной Л. К.