

**Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»**

**56-я НАУЧНАЯ КОНФЕРЕНЦИЯ АСПИРАНТОВ, МАГИСТРАНТОВ
И СТУДЕНТОВ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»**

(Минск, 21–24 апреля 2020 года)

**МАТЕРИАЛЫ КОНФЕРЕНЦИИ
по направлению 2:**

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И УПРАВЛЕНИЕ

Минск БГУИР 2020

УДК 004.9+681.5
ББК 32.97+32.965
П99

Редакционная коллегия:

Л. Ю. Шилин, Д. П. Кукин, А. В. Марков, Д. В. Шункевич,
А. А. Навроцкий, Л. В. Николаева, В. И. Журавлев,
А. Ф. Трофимович, А. Б. Гуринович

П99 **56-я** научная конференция аспирантов, магистрантов и студентов учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»: материалы конференции по направлению 2: Информационные технологии и управление (Минск, 21–24 апреля 2020 года) / редкол. : Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2020. – 208 с. : ил.

ISBN 978-985-543-577-9.

В сборник включены доклады, представленные на 56-й научной конференции аспирантов, магистрантов и студентов высших учебных заведений.

Материалы одобрены организационным комитетом и печатаются в авторской редакции.

Адресуется аспирантам, магистрантам, студентам высших учебных заведений, научным сотрудникам, а также специалистам предприятий в сфере IT-технологий.

УДК 004.9+681.5
ББК 32.97+32.965

ISBN 978-985-543-577-9

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2020

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ

Председатель

Богуш В.А. ректор, д-р физ.-мат. наук, профессор

Заместители председателя:

Давыдов М.В. первый проректор, канд. техн. наук, доцент

Осипов А.Н. проректор по научной работе, канд. техн. наук, доцент

Ответственный секретарь

Гладкая В.С. заведующая сектором студенческой науки, магистр техн. наук

Члены оргкомитета:

Лихачевский Д.В. декан факультета компьютерного проектирования, канд. техн. наук, председатель комиссии по организации и проведению конференции «Электронные системы и технологии»

Шилин Л.Ю. декан факультета информационных технологий и управления, д-р техн. наук, профессор, председатель комиссии по проведению конференции «Информационные технологии и управление»

Короткевич А.В. декан факультета радиотехники и электроники, канд. техн. наук, доцент, председатель комиссии по проведению конференции «Радиотехника и электроника»

Лукашевич М.М. декан факультета компьютерных систем и сетей, канд. техн. наук, доцент, председатель комиссии по организации и проведению конференции «Компьютерные системы и сети»

Дробот С.В. декан факультета инфокоммуникаций, канд. техн. наук, доцент, председатель комиссии по организации и проведению конференции «Инфокоммуникации»

Князева Л.П. декан инженерно-экономического факультета, канд. физ.-мат. наук, доцент, председатель комиссии по проведению конференции «Проблемы экономики и информационных технологий»

Кулешов Ю.Е. начальник военного факультета, канд. воен. наук, доцент, председатель комиссии по проведению конференции «Инновационные технологии в учебном процессе»

Бондарик В.М. декан факультета доуниверситетской подготовки и профессиональной ориентации канд. техн. наук, доцент, председатель комиссии по организации и проведению конференции «Функционирование белорусского и русского языков в условиях информатизации общества»

Маковский М.Л. зам. декана факультета компьютерных технологий Института информационных технологий БГУИР, председатель комиссии по организации и проведению конференции «Информационные системы и технологии»

Тумилович В.Г. начальник управления подготовки научных кадров высшей квалификации, д-р техн. наук, профессор

Строгова А.С. заведующая аспирантурой отдела аспирантуры и докторантуры канд. техн. наук, доцент

Бойправ О.В. доцент кафедры ЗИ, председатель Совета молодых ученых БГУИР канд. техн. наук, доцент

Чубенко Е.Б. заместитель председателя Совета молодых ученых БГУИР вед. научн. сотрудник НИЛ 4.3, канд. техн. наук

Хорошко Л.С. доцент кафедры МНЭ, канд. физ.-мат. наук

Парафенюк Д.А. магистрант гр. 847301

Алексеева С.А. начальник управления воспитательной работы с молодежью

Комиссия конференции «Информационные технологии и управление»

- Шилин Л.Ю. декан факультета информационных технологий и управления,
д-р техн. наук, профессор - председатель комиссии по
проведению конференции «Информационные технологии и управление»
- Трофимович А.Ф. заместитель декана факультета информационных
технологий и управления, член редакционной коллегии сборника
материалов конференции «Информационные технологии и управление»
- Гуринович А.Б. заместитель декана факультета информационных
технологий и управления, канд. физ.-мат. наук, доцент, ответственный
за подготовку, выпуск программы и электронного сборника материалов
конференции «Информационные технологии и управление»
- Голенков В.В. д-р техн. наук, профессор
- Кукин Д.П. канд. техн. наук, доцент, зав. кафедрой ВМиП
- Марков А.В. канд. техн. наук, доцент, зав. кафедрой СУ
- Шункевич Д.В. канд. техн. наук, доцент, зав. кафедрой ИИТ
- Лукьянец С.В. канд. техн. наук, профессор кафедры СУ
- Навроцкий А.А. канд. физ.- мат. наук, доцент, зав. кафедрой ИТАС
- Николаева Л.В. канд. ист. наук, доцент, зав. кафедрой ГД
- Журавлев В.И. канд. техн. наук, доцент, зав. кафедрой ТОЭ
- Хаджинов М.К. канд. техн. наук, доцент кафедры СУ

Секция "Интеллектуальные информационные технологии"

Председатель: д-р техн. наук, проф. Голенков В.В.
Члены жюри: канд. техн. наук, доцент, зав. каф. ИИТ Шункевич Д.В.
канд. физ.-мат. наук, профессор Гулякина Н.А.
канд. техн. наук Давыденко И.Т.
Секретарь ст. преподаватель Гракова Н.В.

ДИНАМИЧЕСКАЯ ГЕНЕРАЦИЯ ГРАФИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЭЛЕМЕНТОВ ДИАГРАММ С ИСПОЛЬЗОВАНИЕМ СЕМАНТИЧЕСКИХ СЕТЕЙ

Рассматривается способ динамической генерации графического представления элементов диаграммы в компоненте визуализации, разработанного в рамках универсального подхода к визуализации различных знаний.

ВВЕДЕНИЕ

Пользовательский интерфейс компьютерной системы является одним из важнейших средств донесения информации конечному потребителю. Различные информационные конструкции требуют различных способов и подходов к их отображению. Таким образом, всегда остается проблема расширяемости средств визуализации знаний.

I. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

Создание универсальных средств визуализации представляет собой очень сложную задачу. Частное решение, позволяющее использовать методы визуализации, отображение которых имеют графоподобную структуру и может быть визуализировано с помощью графа реализовано в качестве компонента для систем, построенных по технологии OSTIS [1]. OSTIS - это открытая технология для проектирования интеллектуальных компьютерных систем на основе семантических сетей с теоретико-множественной интерпретацией.

II. ОСОБЕННОСТИ ШАБЛОНОВ

Разработанный компонент предназначен для работы в web-приложениях, поэтому графическое представление элементов описано с помощью SVG текста [2]. В шаблоне может использоваться конструкция вида “#{nrel_*}”, во время визуализации эта конструкции будет заменена на значение sc-ссылки, которая связана отношением nrel_* с узлом, обозначающим конкретное изображение. Пример шаблона и элемента с его семантической окрестностью представлен на рисунке 1. Визуализация данного шаблона представлена на рисунке 2.

Для шаблонов класса connector_class_image, которые представляют элементы ребер в графе, используется конструкция вида

“#{position_path}”. Переменная position_path просчитывается алгоритмом размещения элементов компонента визуализации баз знаний.

III. ВЫВОДЫ

Для решения проблем отображения различных элементов, был использован подход, позволяющий создавать любые графические представления на основе SVG текста и семантической окрестности элемента.

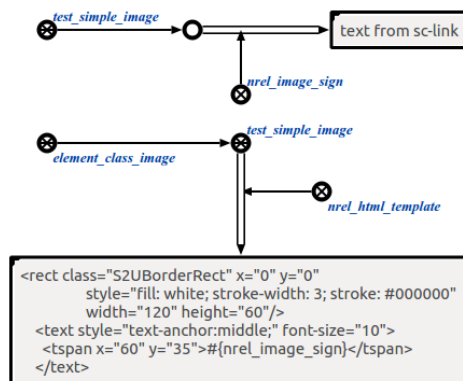


Рис. 1 – Пример шаблона и элемента

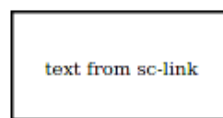


Рис. 2 – Визуализация шаблона

1. Метасистема IMS.OSTIS [Электронный ресурс]. Минск, 2020. – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 01.03.2020.
2. Бобков, А. В. Компонент визуализации графоподобных диаграмм на основе их формального описания в базе знаний / А. В. Бобков // ИТС 2018: материалы международной научной конференции, БГУИР, 2018 - С. 92-93.

Бобков Андрей Валерьевич, магистрант факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, bobkov.andrey.v@gmail.com.

Научный руководитель: Степанова Маргарита Дмитриевна, к.т.н., доцент, доцент кафедры интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, stepanova@bsuir.by

АЛГОРИТМЫ ПОИСКА ПОЛНОСТЬЮ ПРЕДСТАВЛЕННЫХ ФРАГМЕНТОВ ЗНАНИЙ ДЛЯ МОДЕЛИ УНИФИЦИРОВАННОГО СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В работе приводится описание системы системы вычисления канонической формы фрагмента sc-текста, основанной на квазиполиномиальном алгоритме решения изоморфизма графов.

ВВЕДЕНИЕ

Важной задачей при внедрении эффективных программных систем обработки знаний является поиск в базах знаний. Проще представляется подход к поиску на основе перехода к каноническим формам структур знаний. Алгоритмы сравнения канонических форм проще, чем сравнение структур общих форм. Кроме того, время, необходимое для расчета канонической формы, переносится на фазу, не зависящую от самого поиска, например, на фазу формирования базы знаний. Представляется, что такой подход позволяет сократить время отклика системы на запросы пользователей.

I. СИСТЕМА ВЫЧИСЛЕНИЯ КАНОНИЧЕСКОЙ ФОРМЫ ФРАГМЕНТА SC-ТЕКСТА.

Долгое время задача изоморфизма графов алгоритма решения, который в худшем случае работал не лучше полного перебора. В 2015 году Ласло Бабаи представил свой алгоритм решающий данную задачу за квазиполиномиальное время [1]. Данный алгоритм положен в основу разработанной системы вычисления канонической формы фрагмента sc-текста. Такие тексты представляются графами в которых ребра могут соединять как вершины, так и другие ребра. Для упрощения алгоритма sc-текст [2] кодируется классическим помеченным ориентированным графом. Для этого используется подход описанный в статье [3]. Основная идея подхода - поставить в соответствие sc-дуге вершину классического графа, для сохранения направления связи данная вершина соединяется двумя встречными

ребрами с вершиной начала, и одним исходящим с вершиной конца связи. Для представленного таким образом графа вычисляется каноническая форма. Она задаёт такую перестановку вершин, что матрицы смежности двух изоморфных графов становятся идентичными. Это позволяет, зная каноническую форму, определять изоморфность двух графов за $O(N^2)$, где N количество вершин. Кроме того, каноническая форма задаёт однозначное представление матрицы смежности одномерной строкой. Что позволяет проводить поиск в библиотеке размеченных графов за логарифмическое время при использовании древовидных структур данных или ещё быстрее при использовании хеш-таблицы.

ЗАКЛЮЧЕНИЕ

Разработанная система способна позволяет вычислять канонические формы фрагментов sc-текстов. Результат ее работы позволяют решать задачу нахождения дублирования библиотеке таких фрагментов.

Список литературы

1. Babai L. Graph isomorphism in quasipolynomial time // Proceedings of the forty-eighth annual ACM symposium on Theory of Computing. – 2016. – С. 684-697.
2. Представление и обработка знаний в графодинамических ассоциативных машинах // В. В. Голенков [и др.] – Минск. – 2001. – 413 с.
3. Представление семантических сетей и алгоритмы их организации и семантической обработки на вычислительных системах с массовым параллелизмом. / В. П. Ивашенко [и др.] // Open Semantic Technologies for Intelligent Systems (OSTIS-2015) : материалы V междунар. науч.-техн. конф. / БГУИР. – Минск 2015 – С. 133-140.

Бруцкий Дмитрий Сергеевич, магистрант факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, dzmitry.brutski@gmail.com

Научный руководитель: Ивашенко Валерьян Петрович, доцент кафедры интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, ivashenko@bsuir.by

МЕТОДИКА ОЦЕНКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПО ОТДЕЛЬНЫМ НАПРАВЛЕНИЯМ

В работе приводится краткое описание методики оценки пользовательского интерфейса по отдельным направлениям.

ВВЕДЕНИЕ

Актуальность проблемы анализа качества пользовательских интерфейсов вызвана следующими причинами: большинство пользователей обладает средними или низкими навыками работы с информационными системами; большинство программистов-разработчиков не являются специалистами в области разработки пользовательских интерфейсов; в процессе развития средств вычислительной техники происходит постоянное усложнение информационных систем и расширение их функциональности. Следовательно, построение эффективных пользовательских интерфейсов, позволяющих упростить работу пользователей и сделать ее более эффективной имеет первоочередное значение.

I. ОЦЕНКА ОТДЕЛЬНЫХ ЭЛЕМЕНТОВ ЭКРАННОЙ ФОРМЫ

Каждому элементу экранной формы соответствует определенный набор параметров (шрифт, пропорции, цвет, информационная нагрузка). Оценка элемента экранной формы, как самостоятельного объекта, основана на сопоставлении его параметров с предварительно заданным набором эталонных значений и допустимых отклонений. Методика анализа отдельных экранных форм основана на системе правил проверки элементов.

По каждому из направлений анализа (шрифт, пропорции, цвет, информационная нагрузка) формируется группа правил, определяющих каким должны быть эталонные значения параметров элемента экранной формы. Каждое правило содержит описание элементов, к которым оно должно применяться, эталонные значения параметров элемента, а также величину отклонения параметров от эталонных значений.

Предлагаются следующие правила проверки и анализа элементов:

- правила анализа пропорций;
- правила анализа шрифтов;
- правила анализа цветов;

Веренич Максим Сергеевич, магистрант кафедры интеллектуальных информационных технологий БГУИР, maksim.viarenich@gmail.com.

Научный руководитель: Степанова Маргарита Дмитриевна, кандидат технических наук, старший научный сотрудник, доцент кафедры интеллектуальных информационных технологий БГУИР, stepanova@bsuir.by.

- проверка информационной нагруженности.

Анализ пропорций заключается в проверке горизонтальных и вертикальных размеров элемента экранной формы, а также координат его размещения в пространстве на соответствие заданным параметрам.

Анализ шрифтов заключается в проверке шрифтов и кегля используемых для отображения надписей на элементе на соответствие заданным параметрам.

Анализ цветов элементов экранной формы заключается в сравнении цветов текста и фона с заданными значениями.

Восприятие экранной формы связано с оценкой и переработкой сознанием человека информации, размещенной на ней, большая часть которой, как правило, сосредоточена в нескольких элементах экранной формы. Поэтому следует обеспечить рациональное распределение информационной нагрузки между элементами – с одной стороны количество информации, приходящейся на элемент, не должно перегружать сознание человека, а с другой не должно быть слишком малым, чтобы не вызывать излишнее переключение внимания от одного элемента экранной формы к другому. Проверка информационной нагруженности элементов экранной формы заключается в оценке количества информации, приходящейся на экранную форму и сравнении полученного значения с заданной величиной. Это позволяет оценить сложность элемента экранной формы для его восприятия человеком (с точки зрения количества информации).

ЗАКЛЮЧЕНИЕ

Алгоритм оценки и анализа экранной формы на основе настраиваемой системы правил позволяет сделать анализ гибким, в частности, выполнять проверку не всех элементов экранной формы, а ограничиться элементами определенного типа. Кроме того, последовательное применение правил позволяет пользователю определить приоритеты анализа.

Несмотря на концептуальную простоту, предложенный подход дает общий способ (критерий) оценки качества интерфейса.

СПОСОБ ВЗАИМОДЕЙСТВИЯ СУБЪЕКТОВ ЭКОНОМИКИ ВНУТРИ ЦИФРОВОЙ ФИНАНСОВОЙ ЭКОСИСТЕМЫ НА ОСНОВЕ ПЛАТЕЖНЫХ СИСТЕМ

В работе приводится описание общего способа взаимодействия субъектов экономики внутри цифровой финансовой экосистемы.

ВВЕДЕНИЕ

Актуальность статьи состоит в том, что для обеспечения национального суверенитета и глобальной конкурентоспособности требуется формирование модели цифровой экономики. В работе предложена общий способ взаимодействия внутри цифровой финансовой экосистемы, объединяющей рынки и отрасли экономики на базе инновационных платформ и технологий платежных систем и платежных сервисов.

1. ОБЩЕЕ ОПИСАНИЕ СПОСОБА

Современный платежный рынок представлен взаимодействием предложения и спроса. Наряду с традиционным спросом на услуги платежных систем цифровая экономика будет стимулировать спрос на гибридные продукты платежных систем и других отраслей экономики. Диффузия финансовых цифровых инноваций платежных систем в другие сферы экономики происходит на основе платформ и технологий, где формируются компетенции для развития собственно платежного и смежных рынков: роботизация и искусственный интеллект, распределенные реестры, блокчейн, «облачные» вычисления, нейротехнологии, биометрия, которые формируют технологическую составляющую новой цифровой среды. Инновационные платформы и технологии, в свою очередь, формируют новые сегменты рынка: финансового рынка (FinNet) и в его структуре платежного рынка (PayNet). Такими перспективными сегментами платежного рынка на данный момент являются облачные и мобильные сервисы, операции с большими данными, открытые интерфейсы, а также интернет вещей. Общий способ взаимодействия данных субъектов экономики представлен на Рис. 1.

Винокур Андрей Евгеньевич, магистрант факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, gobest.vini@gmail.com.

Научный руководитель: Степанова Маргарита Дмитриевна, кандидат технических наук, старший научный сотрудник, доцент кафедры ИИТ, stepanova@bsuir.by.



Рис. 1 – Схема взаимодействия внутри цифровой финансовой экосистемы

ЗАКЛЮЧЕНИЕ

Представленный общий способ взаимодействия внутри цифровой финансовой экосистемы реализует основные задачи цифровой экономики для платежных систем и их элементов: совершенствование платежных сетей, обеспечивающих коммуникационные потребности экономики по расчетно-платежному обслуживанию государственных и муниципальных структур, субъектов бизнеса и населения с учетом технических требований, предъявляемых цифровыми технологиями; внедрение инновационных технологий и цифровых платформ.

Список литературы

1. Буркальцева Д.Д. Алгоритм внедрения программы «Цифровая экономика», 2017. — С. 141–147
2. Коробейникова О.М. Способы и методы минимизации рисков в локальных платежных системах / О.М. Коробейникова // Финансы и кредит. — 2012. — № 17 (497). — С. 66–74.

ВОЗМОЖНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В СФЕРЕ ПРЕДОСТАВЛЕНИЯ ИНФОРМАЦИИ О ТРАНСПОРТНЫХ СРЕДСТВАХ

В статье рассматривается актуальность информационных технологий в качестве источника информации о транспортном средстве

ВВЕДЕНИЕ

Потенциальный покупатель бывшего в употреблении механического транспортного средства заинтересован в получении информации о том, какие проблемы у транспортного средства были в прошлом, такая информация может помочь принять решение о покупке транспортного средства. Каждое механическое транспортное имеет уникальный идентификационный номер присваиваемый производителем, данный номер наносится на кузов и шасси транспортного средства. Современные информационные технологии позволяют на основе уникального идентификационного номера получить широкий спектр информации.

1. Источники информации

Уникальный идентификационный номер механического транспортного средства состоит из 17 символов, данный номер содержит информацию о производителе, годе выпуска, модели, комплектации и серийный номер. Пользователь может сам расшифровать уникальный идентификационный номер, но данная задача может быть осложнена тем, что некоторые производители меняли свой код, поэтому для пользователя проще воспользоваться приложениями, предоставляющими услуги декодирования уникального идентификационного номера. Хорошим источником информации о истории транспортного средства может быть организация занимающаяся сбором информации и регистрацией транспортных средств. В США такой организацией является NMVTIS (National Motor Vehicle Title Information System), которая была создана для достижения следующих целей:

- предотвращение попадания украденных транспортных средств в продажу;
- защитить потребителей от мошенничества;
- предоставить потребителям защиту от небезопасных транспортных средств;

Демешко Александр Витальевич, магистрант факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, demeshko.alexander.v@gmail.com

Научный руководитель: Поттосин Юрий Васильевич, кандидат технических наук, доцент.

NMVTIS предоставляет следующую информацию о транспортных средствах:

- информацию о текущих и предыдущих участиях транспортного средства в автотранспортных государственных компаниях;
- актуальную информацию о пробеге автомобиля;
- информацию о зарегистрированных повреждениях и иных недостатках транспортного средства;
- информацию о страховании транспортного средства;

NMVTIS не предоставляет данную информацию пользователям напрямую, а использует провайдеров, которые находятся в состоянии конкуренции, что мотивирует их добавлять в предоставляемый пользователю отчет об истории транспортного средства дополнительную информацию. Дополнительная информация может содержать информацию о рекомендованной для покупки цене, основанной на анализе рынка; информацию о годовой стоимости владения транспортным средством; информация о нахождении транспортного средства в качестве залога; участии данного транспортного средства в отзывных компаниях проводимых производителем в связи с производственными дефектами.

Выводы

Имея отчет об истории транспортного средства можно удостовериться в безопасности его приобретения. Системы, предоставляющие информацию о механических транспортных средствах, должны использовать достоверные источники информации. Использование таких систем экономит время, поскольку пользователю не нужно запрашивать данные из разных источников таких, как производитель, диллеры и другие структуры.

МОДЕЛЬ АВТОМАТИЗИРОВАННОЙ ВЕРИФИКАЦИИ КОМПОНЕНТОВ МАШИН ОБРАБОТКИ ЗНАНИЙ

Рассматривается модель автоматизированной верификации компонентов машин обработки знаний, построенных по Технологии OSTIS.

ВВЕДЕНИЕ

Верификация является неотъемлемым этапом разработки программных продуктов. Автоматизация этого процесса позволяет снизить трудозатраты при повторном её проведении, а также позволяет своевременно обнаруживать дефекты, что снижает затраты на их исправление.

В работе рассматриваются системы, построенные по Технологии OSTIS (ostis-системы [1]). В качестве модели представления знаний в ostis-системах используется семантическая сеть с теоретико-множественной интерпретацией.

Машина обработки знаний (МОЗ), построенная по Технологии OSTIS, представляет собой графодинамическую машину, включающую в себя систему агентов и общую память. Отличительной особенностью таких МОЗ является то, что агенты взаимодействуют друг с другом исключительно через общую память.

Для осуществления автоматизированной верификации компонентов МОЗ ostis-систем применяется одноименные средства [2]. Данные средства являются интерпретацией разработанной модели автоматизированной верификации. При этом как модель, так и интерпретация представлены в виде семантических сетей.

I. МОДЕЛЬ АВТОМАТИЗИРОВАННОЙ ВЕРИФИКАЦИИ

Модель автоматизированной верификации компонентов МОЗ задаётся пятёркой $M_{AV} = \{O_{VP}, O_{TC}, O_{TS}, VO, O_A, O_R\}$, где

- O_{VP} – онтология процессов верификации компонентов МОЗ;
- O_{TC} – онтология тест-кейсов;
- O_{TS} – онтология тест-наборов;
- VO – множество объектов верификации;
- O_A – онтология действий, направленных на верификацию компонентов МОЗ.

Под онтологией будем понимать спецификацию (описание) понятий и их взаимосвязей для соответствующей предметной области [1].

Онтология процессов верификации компонентов МОЗ описывает процессы верификации компонентов МОЗ различных видов. В спецификацию процесса верификации входят инициатор

верификации, проверяемые тест-кейсы, результаты верификации (отчёт).

Онтология тест-кейсов описывает тест-кейсы, их классификацию и структуру. В спецификацию тест-кейса входят описание тест-кейса на естественном языке, верифицируемый компонент МОЗ, аргументы, с которыми необходимо запустить его на исполнение, контекст верификации, ожидаемый и фактический результаты, компонент МОЗ, осуществляющий сравнение ожидаемого результата с фактическим, полученным в рамках некоторого процесса верификации.

Один тест-кейс осуществляет верификацию только одного требования. В связи с этим возникает необходимость проверки сразу нескольких тест-кейсов одновременно. Если проверяемые тест-кейсы имеют общие цели и/или условия верификации, их можно объединять в тест-наборы. Описывает спецификацию тест-наборов Онтология тест-наборов.

Онтология действий, направленных на верификацию компонентов МОЗ, описывает действия, выполняемые в рамках автоматизированной верификации, а также действия, упрощающие манипулирование тест-кейсами и тест-наборами, например, осуществляющие удаление тест-кейса/тест-набора.

Выводы

Предлагаемая модель позволяет снизить трудозатраты на проведение верификации по сравнению с ручной верификацией за счёт унификации описания используемых понятий, автоматизации некоторых этапов верификации.

1. Документация IMS [Электронный ресурс]. Минск, 2017. – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 10.02.2020.
2. Дюбина, Е. А. Средства верификации машин обработки знаний интеллектуальных систем / Е. А. Дюбина, И. Т. Давыденко // Информационные технологии и системы 2018 (ИТС 2018) = Information Technologies and Systems 2018 (ITS 2018) : материалы международной научной конференции, Минск, 25 октября 2018 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2018. – С. 96 - 97.

Дюбина Елена Александровна, магистрант кафедры интеллектуальных информационных технологий БГУИР, dziubina.el@gmail.com.

Научный руководитель: Степанова Маргарита Дмитриевна, доцент кафедры интеллектуальных информационных технологий БГУИР, stepanova@bsuir.by.

СТРУКТУРНАЯ АРХИТЕКТУРА ПРИЛОЖЕНИЙ ДЛЯ ОРИЕНТАЦИИ НА МЕСТНОСТИ

В работе проводится описание структурных блоков и их взаимодействия в приложении с использованием камеры и геоданных мобильного устройства.

ВВЕДЕНИЕ

В данный момент практически нет приложений для ориентации на местности с использованием камеры мобильного устройства и его геоданных, в особенности систем, которые позволяют определять видимые здания при направлении на них камеры.

Существуют алгоритмы распознавания образов, где применяется машинное обучение. Однако они достаточно сложны для понимания и оперирования ими. Поэтому существуют необходимость в создании простой архитектуры с достаточно простым основным алгоритмом ориентации.

В данной статье в общих чертах будет описана структурная архитектура подобных приложений, в которой будут указаны структурные блоки, функции которые они выполняют, данные которые используют, а так же взаимодействие этих блоков.

I. ОСНОВНЫЕ ЗАДАЧИ РАЗРАБОТАННОЙ АРХИТЕКТУРЫ

Основной задачей будет создание самого алгоритма, по которому приложение будет вычислять, какие здания являются видимыми, а какие здания не видны. Следующей крупной задачей будет доступ к датчикам мобильного устройства. Третья крупная задача - это доступ к картографическим ресурсам, от которых можно получить информацию о координатах реальных домов и об их высоте. Таким образом можно определить логические блоки, которые обязательно должны присутствовать в создаваемом мобильном приложении.

II. СТРУКТУРНЫЕ БЛОКИ АРХИТЕКТУРЫ

Основные структурные блоки, на которые можно разделить решение задачи ориентации с помощью мобильного устройства:

- *Блок взаимодействия с мобильным устройством*

Отвечает за получение входных данных, высоты пользователя над уровнем моря (земли), местоположением пользователя,

Ермаченок Игорь Сергеевич, магистрант кафедры интеллектуальных информационных технологий БГУИР, ihar.yearmachonak@gmail.com.

Научный руководитель: Сердюков Роман Евгеньевич, кандидат технических наук, доцент кафедры интеллектуальных информационных технологий БГУИР, rserdyukov@gmail.com.

направление взгляда пользователя (азимут устройства), вывод выходных данных

- *блок взаимодействия с картографической информацией*
- *блок решения основной вычислительной задачи*

Работа блока взаимодействия с мобильным устройством имеет конкретный характер и может быть различна на разных платформах и языках программирования. Работа блока взаимодействия с картографической информацией зависит от конкретных сервисов, которых сейчас достаточно. На вход блока решения основной вычислительной задачи поступает информация из двух других блоков. С одной стороны - это азимут, координаты пользователя и высота местоположения пользователя. Эти данные поступают от датчиков мобильного устройства или вводятся пользователем самостоятельно. С другой стороны - это географические координаты и высоты зданий, находящиеся в ближайшем окружении пользователя. Описанные блоки и их взаимодействие представлены на Рис. 1.

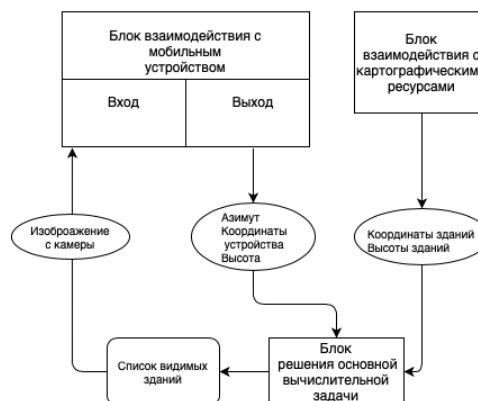


Рис. 1 – Структурные блоки архитектуры

ЗАКЛЮЧЕНИЕ

Разработанная архитектура может применяться только в случае правильного алгоритма ориентации, который сможет использовать входные и выходные данные эффективно.

РАЗДЕЛЕНИЕ ДАННЫХ, КАК УНИВЕРСАЛЬНЫЙ МЕТОД ОПТИМИЗАЦИИ ПРИЛОЖЕНИЙ, РАБОТАЮЩИХ С БОЛЬШИМИ ДАННЫМИ

Рассматриваются универсальные методы оптимизации приложений, работающих с большими данными: разделение данных, его типы и особенности

ВВЕДЕНИЕ

В связи с ростом количества приложений и технологий, работающих с большими данными, возникает потребность в решении задачи оптимизации данных приложений. Для решения этой задачи предлагается использовать метод универсальный для технологий, работающих с большими данными: разделение данных. Преимуществом данного метода является то, что для его реализации не требуется привлечения дополнительных вычислительных мощностей.

I. РАЗДЕЛЕНИЕ ДАННЫХ

Разделение данных - это метод физического разделения данных во время обработки, загрузки или сохранения основных данных.

Типы разделения данных:

- вертикальное разделение данных;
- горизонтальное разделение данных;
- гибридное разделение данных.

Вертикальное разделение данных - тип разделения данных, при котором исходные атрибуты данных разделяется на множество атрибутов хранящихся и обрабатываемых отдельно друг от друга, для идентификации значений атрибутов данных используются атрибуты-ключи. Атрибуты разделены в соответствии со схемой их использования. Например, часто используемые атрибуты могут быть размещены в одном вертикальном разделе, а менее часто используемые атрибуты - в другом.

Горизонтальное разделение данных (часто называемое разбиением). В этой типе каждый раздел является отдельным хранилищем данных, но все разделы имеют одинаковую схему. Каждый раздел называется шардом и содержит определенное подмножество данных, например, все заказы для определенного набора клиентов [1].

Типы горизонтального разделения данных:

Жук Александр Андреевич, магистрант кафедры интеллектуальных информационных технологий БГУИР, san91130324@gmail.com.

Научный руководитель: Бойко Игорь Михайлович, кандидат технических наук, igor_m_boyko@hotmail.com.

- атрибутное разделение данных;
- разделение данных с фиксированным числом разделов.

Атрибутное разделение данных - разделение данных на основании значения одного или нескольких атрибутов разделяемых данных. Главной характеристикой данного типа разделения данных является заранее известные значения атрибутов используемых для разделения данных.

Разделение данных с фиксированным числом разделов - разделение данных, при котором исходный набор данных распределяется между заранее заданным числом разделов на основании заранее определенной функции, например, хэш-функции.

Гибридное разделение данных - это разделение объединяет вертикальное и горизонтальное разделение.

II. УЛУЧШЕНИЯ ДОСТИЖИМЫЕ ПРИ ПОМОЩИ РАЗДЕЛЕНИЯ ДАННЫХ

Использование разделения данных позволяет улучшить масштабируемость хранения и обработки данных, улучшить производительность за счет равномерного распределения нагрузки на вычислительный кластер, улучшить доступность данных, улучшить безопасность при работе с данными, за счет возможности применения различных политик безопасности на различные разделы данных [1].

III. ВЫВОДЫ

Разделение данных является мощным и универсальным инструментом для улучшения качества приложений и технологий, работающих с большими данными.

1. Horizontal, vertical, and functional data partitioning [Electronic resource] / Microsoft Azure. – Mode of access: <https://docs.microsoft.com>. – Date of access: 12.03.2020.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРОВЕДЕНИЯ ФУТБОЛЬНЫХ ТУРНИРОВ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ

ВВЕДЕНИЕ

Стремительное развитие и грамотное применение информационных технологий обуславливает ведение эффективного бизнеса и достижения высокого качества жизни современного общества.

Совмещение сферы автоматизации со сферой мобильных технологий чрезвычайно востребовано и перспективно, что подтверждает растущее число мобильных операционных систем, различных сервисов и приложений. С помощью применения мобильных технологий можно автоматизировать даже такой процесс, как поиск и организацию футбольного турнира. Это позволит спортсменам со всего мира находить наиболее подходящие для них локации и события. Также использование мобильных технологий позволяет избавиться от самостоятельного поиска локации для проведения таких турниров.

Актуальность автоматизации процесса организации футбольных турниров и необходимость создания соответствующего решения ставит ряд задач, которые должна реализовывать система:

- определение критериев поиска футбольных турниров;
- организация поиска турниров по заданным критериям;
- определение критериев подбора турниров;
- создание турнирной сетки;
- управление личными данными участников.

I. АНАЛИЗ СИСТЕМ ПРОВЕДЕНИЯ ТУРНИРОВ

Были проанализированы следующие системы проведения турниров: олимпийская, круговая, швейцарская, смешанная. Подробный анализ перечисленных систем показал, что наиболее подходящей является смешанная система, так как она позволяет при достаточно большом количестве участвующих команд провести соревнования в сравнительно небольшой срок и доволь-

но точно определить соотношения сил команд-участниц.

II. ВЫБОР ИНСТРУМЕНТОВ РАЗРАБОТКИ

Для разработки используются мобильной части используется Flutter - фреймворк от компании Google, который позволяет создавать кроссплатформенные приложения, которые могут использовать один и тот же код. Спектр платформ широк - это веб-приложения, мобильные приложения под Android и iOS, графические приложения под настольные операционные системы Windows, MacOS, Linux.

В качестве языка разработки используется язык программирования Dart.

При построении приложения Flutter транслирует код на Dart в нативный код приложения, которое можно запускать на Android или iOS или другой платформе.

Особенностью работы с Flutter является то, что приложения под разные платформы могут иметь один и тот же код. Поскольку используемые платформы не эквиваленты, то какие-то отдельные части кода необходимо настраивать под определенную ОС, например, под iOS, но тем не менее большая часть кода может совпадать. Это позволяет существенно экономить время и ресурсы на создание приложения под обе платформы.

В качестве серверной части используются сервисы Firebase от компании Google. Платформа содержит множество полезных функций, в том числе серверный код для мобильных сервисов, статистику, а также инструменты для монетизации и расширения аудитории.

Диаграмма компонентов системы представлена на рисунке 1.

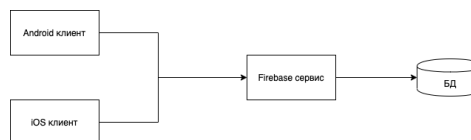


Рис. 1 – Диаграмма компонентов системы проведения турниров

Корольков Кирилл Петрович, магистрант кафедры интеллектуальных информационных технологий БГУИР, korolkir@gmail.com.

Научный руководитель: Романов Владимир Ильич, доцент кафедры каф. ИИТ, rom1954@tut.by.

ИСПОЛЬЗОВАНИЕ ЭКСПЕРТНЫХ СИСТЕМ ДЛЯ АНАЛИЗА И ОЦЕНКИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

В этом документе рассматриваются вопросы использования экспертной системы(ЭС) для предметной области компьютерной безопасности.

ВВЕДЕНИЕ

Аудит компьютерной безопасности является важной частью системы безопасности любой организации. Успех обеспечения ИБ — заключается в комплексном подходе. При этом все средства, методы и мероприятия, используемые для защиты информации, объединяются в единый целостный механизм — систему защиты. От части решение задач обеспечения информационной безопасности организации может быть получено на базе использования экспертных систем, как части системы защиты информации.

I. ИСПОЛЬЗОВАНИЕ ЭС ДЛЯ АУДИТА БЕЗОПАСНОСТИ

Рассмотрим подробнее что же такое «Анализ системы защиты информации» и для чего он применяется. Одно из его определений [1] — это комплексное изучение фактов, событий, процессов, явлений, связанных с проблемами защиты информации, в том числе данных о состоянии работы по выявлению возможных каналов утечки информации, о причинах и обстоятельствах, способствующих утечке и нарушениям режима секретности (конфиденциальности) в ходе повседневной деятельности предприятия.

Аудит безопасности часто используется для обнаружения аномальных событий, которые выходят за рамки мер безопасности в реальном времени. Методы искусственного интеллекта (в частности, методы экспертных систем) могут многое предложить специалистам по компьютерной безопасности. Экспертные система предназначена для автоматизации процедур аудита безопасности и снижения нагрузки на аудиторов. Экспертная система должна уметь взять на себя те функции, которые выполняет специалист-эксперт или выполнить роль ассистента для лица, принимающего решения. Использование экспертных систем позволит управляющей информационной системе получать решение непосредственно от программы и полностью исключить

необходимость использования человека в управляющей системе. С другой стороны, экспертная система может повысить эффективность работы человека, предлагая наиболее верное решение поставленной задачи.

II. ПОСТРОЕНИЕ ЭС ДЛЯ АУДИТА БЕЗОПАСНОСТИ

При проектировании экспертной системы выполняются следующие шаги.

1. Определяются основные угрозы информационной безопасности и производится их классификация.

2. Для каждой угрозы информационной безопасности определяется список уязвимостей, через которые эта угроза может быть реализована.

3. С помощью информационной безопасности для каждой уязвимости определяется список требований, которые должны быть выполнены, чтобы избежать осуществления угрозы.

4. Каждому запросу присваивается вес, выражающий степень важности требований.

Таким образом, экспертная система состоит из модулей, которые включают в себя требования безопасности, предъявляемые к каждой уязвимости..

III. ВЫВОДЫ

Для повышения качества аудита безопасности целесообразно применять экспертные системы. Применение ЭС для обеспечения информационной безопасности на предприятии позволяет существенно повысить уровень информационной безопасности, несколько упростить процесс обнаружения и анализа проблем информационной защиты, а так же использовать опыт экспертов в области информационной безопасности.

1. Курило А.П., Зефилов С.Л., Голованов В.Б. Аудит информационной безопасности.— М.: Издательская группа «БДЦ-пресс», 2006 г.

Лось Павел Николаевич, магистрант кафедры интеллектуальных информационных технологий БГУИР pasha-los96@yandex.ru.

Научный руководитель: Захаров Владимир Владимирович, доцент кафедры интеллектуальных информационных технологий БГУИР, кандидат технических наук, доцент, zvv@open.by.

ПОСТРОЕНИЕ МОДЕЛИ БАЗЫ ЗНАНИЙ ДЛЯ ЭКСПЕРТНОЙ СИСТЕМЫ АУДИТА БЕЗОПАСНОСТИ

В этом документе рассматривается проблема описания и проектирования базы знаний(БЗ) для экспертной системы(ЭС) аудита компьютерной безопасности.

ВВЕДЕНИЕ

Построение модели интеллектуального анализа компьютерной безопасности является частью масштабного процесса, в который входят все задачи, от формулировки вопросов выбора и хранения данных и создания модели до развертывания модели в рабочей среде. В этом документе показаны особенности построения модели анализа компьютерной безопасности.

I. ПРОЕКТИРОВАНИЕ БАЗЫ ЗНАНИЙ ДЛЯ ЭС

Целью базы знаний экспертной системы является представление специфичных для предметной области знаний в форме, которую компьютер может использовать для эффективной работы с этими знаниями.

Структура проектируемой системы показана на рисунке 1. База знаний для анализа представляет собой набор фактов, которые отражают состояние компьютерной системы, информация об уязвимостях программного обеспечения и возможные сетевые проблемы. Правила взаимодействия (Interaction rules) определяют, как различные части системы могут взаимодействовать и влиять на безопасность. Interaction rules определяют семантику ЭС: различные виды уязвимостей и способы их возникновения, поведение программного обеспечения, влияющее на безопасность, конфигурация доступа к сети. Политика безопасности (Security policy) - это совокупность правил, процедур, практических методов и руководящих принципов в области ИБ, используемых организацией в своей деятельности. В разрабатываемой системе политика - это кортежи, которые перечисляют права доступа к данным для пользователей.

Лось Павел Николаевич, магистрант кафедры интеллектуальных информационных технологий БГУИР pasha-los96@yandex.ru.

Научный руководитель: Захаров Владимир Владимирович, доцент кафедры интеллектуальных информационных технологий БГУИР, кандидат технических наук, доцент, zvv@open.by.

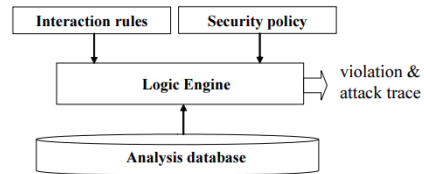


Рис. 1 – Структура ЭС

Онтология состоит из 4 основных сущностей и отношений между ними:

1. Модель аппаратного домена. Является абстракцией для описания проверяемой машины и всей информации о сети, к которой подключена эта машина.

2. Модель активов (Assets) - это активы, наиболее ценные ресурсы в системе. В качестве актива могут выступать различные источники данных: база данных, файл или системная служба. Иногда определение актива может быть расплывчатым, но, как правило, это все, что важно для организации-владельца или пользователя.

3. Модель программного обеспечения. Описывает существующее(установленное) ПО на данной машине. Программное обеспечение может зависеть от другого программного обеспечения, следовательно, любая часть программного обеспечения, которая зависит, например, от неисправной библиотеки, также потенциально уязвима.

4. Модель безопасности пользователя. Описание пользователей, рабочих групп.

II. ВЫВОДЫ

Описание модели предметной области было выполнено, не привязываясь к определенной реализации или операционной системе. При этом описание является достаточным, что бы идентифицировать каждую сущность, ее связи и поведение в системе.

ВЕБ-СИСТЕМА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ПРЕДПРИЯТИЯ

ВВЕДЕНИЕ

С появлением больших компаний, которые насчитывают более 50-100 человек, появляется серьезная проблема с отсутствием единого информационного пространства, взаимодействием и коммуникацией между удаленными сотрудниками, а также автоматизацией бизнес-процессов. Решением этих проблем является внедрение веб-системы информационного обеспечения предприятия (далее корпоративный портал).

I. Анализ подходов к решению ПРОБЛЕМЫ

Существует большое количество точечных решений данных проблем, но только корпоративный портал предоставляет комплексное решение проблемы. К примеру, для решения проблемы взаимодействия и коммуникации между удаленными сотрудниками могло бы быть ПО Skype, а для автоматизации бизнес-процессов использовать Atlassian Jira. Но оба эти решения являются платными и решают только одну конкретную проблему.

В отличие от других подходов к решению, указанных в введении, проблем предлагаемый подход (корпоративный портал) позволяет аккумулировать внутри себя все необходимые функциональности для решения поставленных проблем, предоставляет большое количество интеграций для улучшения производительности труда сотрудника, а также расширенную систему безопасности, где вся корпоративная информация будет находиться под надежной защитой.

II. ПОНЯТИЕ И СПЕЦИФИКА КОРПОРАТИВНОГО ПОРТАЛА

Корпоративный информационный портал — это программный продукт, позволяющий организации выявлять информацию, которая хранится внутри и вне компании, и предоставляющий каждому пользователю унифицированную точку доступа к предназначенной для него информации.

Назначение корпоративного информационного портала определяется в достижении следующих целей:

- обеспечение для каждого сотрудника компании возможности аккумулирования кор-

поративных знаний (информационных ресурсов предприятия);

- сохранение этих знаний как составной части информационных ресурсов предприятия;
- обеспечение коллективного использования сотрудниками компании текущих и архивных корпоративных знаний;
- распространение знаний (текущих и архивных), хранящихся в информационном банке компании в состоянии непрерывного оповещения или информационного обслуживания по запросам;
- создание коммуникационной среды для сотрудников;
- автоматизация бизнес-процессов;

III. Подходы к созданию портала

На данный момент существует большое количество готовых решений по созданию порталов. Большинство решений по созданию портала можно условно подразделить на три группы:

- платформы для создания корпоративных порталов;
- готовые решения на базе платформ;
- SaaS-сервисы (облачные решения).

IV. Выводы

Корпоративный информационный портал — не только инструмент совместной работы, который позволяет компании развиваться. Он предоставляет широчайшие возможности для управления ее корпоративной культурой. При грамотном, продуманном использовании он становится стабильной опорой предприятия в движении к целям, озвученным в его миссии.

1. Погорелик, Т. А. Система управления знаниями предприятия. Экономический вестник Ростовского государственного университета / Погорелик, Т. А. // . — 2009. — Том 7. — С. 163-165.

Мелкумов Артем Эдуардович, магистрант кафедры ИИТ БГУИР, melkumov.artem@gmail.com.

Научный руководитель: Поттосин Юрий Васильевич, доцент, кандидат технических наук.

СРЕДСТВА ПРОГНОЗИРОВАНИЯ ЗАГРУЖЕННОСТИ ПАРКОВОК

Рассматриваются средства прогнозирования загруженности городских парковок. Предлагается использование рекуррентных нейронных сетей с долгой краткосрочной памятью (LSTM).

ВВЕДЕНИЕ

В настоящее время актуальной проблемой среди водителей транспортных средств является поиск парковочного места. Современные навигационные приложения позволяют отслеживать ситуацию на дорогах, однако не показывают доступность парковочных мест. Во многом это связано с тем, что загруженность парковок зависит от множества факторов, что делает построение прогнозов сложной задачей.

Выходом из сложившейся ситуации может стать разработка средств прогнозирования загруженности парковок.

I. ПРЕДЛАГАЕМЫЙ ПОДХОД

Главным требованием к подсистеме является предсказание количества свободных мест в указанный момент времени.

Программное решение в своей архитектуре использует методологию интернета вещей. На месте каждого парковочного места находится беспроводной сенсор, который может находиться в двух состояниях: занят или свободен. На каждой парковке также присутствует базовая станция, собирающая данные с сенсоров.

Данные представлены записями о состоянии датчиков: адрес, время прибытия и отъезда автомобиля, погодные условия и т.п. Собранные данные о прибытии и отбытии автомобилей обладают периодичностью во времени, а также сопоставимы по всем признакам, следовательно, можно говорить, что рассматриваемая последовательность является временным рядом, и для нее применимы методы анализа и прогнозирования временных рядов.

До недавних пор основными методами прогнозирования временных рядов оставались статистические методы. Однако связанные с этими методами математические модели не всегда линейны, и поэтому они не могут прогнозировать сложные явления и процессы, в которых модель данных может быть нелинейной. В этих случаях и приходит на помощь аппарат нейронных сетей.

Рекуррентные нейронные сети (англ. Recurrent neural network) вид нейронных сетей, где связи между элементами образуют направ-

ленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные странственные цепочки.

Нейронные сети с долгой краткосрочной памятью (англ. Long short-term memory) — разновидность архитектуры рекуррентных нейронных сетей, которая хорошо приспособлена к обработке и прогнозирования временных рядов в случаях, когда важные события разделены временными лагами с неопределённой продолжительностью и границами.

LSTM-блоки содержат три или четыре «вентиля», реализованных в виде логической функции для вычисления значения в диапазоне $[0; 1]$. Они используются для контроля потоков информации на входах и выходах памяти блоков. Веса в блоке используются для задания направления оперирования вентилями. LSTM-блоки обычно тренируют при помощи метода обратного распространения ошибки во времени.

Нейронные сети с долгой краткосрочной памятью используются для воспроизведения последовательности значений загруженности O_t на основе прошлых значений, чтобы предсказать значение на один шаг вперед. Использование экзогенных переменных, таких как день недели, погодные условия и т.п. необходимо для повышения точности предсказаний.

II. ВЫВОДЫ

Возможность моделировать временные последовательности и нелинейные зависимости является главным достоинством нейронных сетей с долгой краткосрочной памятью, что делает целесообразным их использование в прогнозировании загруженности парковок. Относительная невосприимчивость к длительности временных разрывов даёт сетям с долгой краткосрочной памятью преимущество по отношению к другим моделям и позволяет повысить качество прогнозирования в задачах подобного рода.

1. Афанасьев, В. Н. Анализ временных рядов и прогнозирование: Учебник. / М. М. Юзбашев // М.: Финансы и статистика. – 2001. – С. 5-15.

Пинчук Алексей Алексеевич, магистрант кафедры интеллектуальных информационных технологий БГУИР, alexey.pinchuk.jr@gmail.com.

Научный руководитель: Романов Владимир Ильич, кандидат технических наук, доцент кафедры интеллектуальных информационных технологий БГУИР, rom1954@tut.by.

СПОСОБЫ СБОРА ДАННЫХ ПОЛЬЗОВАТЕЛЕЙ СОЦСЕТЕЙ

Рассматриваются основные подходы и способы сбора данных пользователей социальных сетей, а также их особенности.

ВВЕДЕНИЕ

Понятие социальной сети использовалось социологами еще в 20-х годах прошлого века для изучения взаимосвязей между участниками различных сообществ. Одним из вариантов стало использование социограмм, на которых отдельные индивиды представлялись в виде точек, а взаимосвязи между ними — в виде линий. Идею использования аппарата теории графов для изучения взаимоотношений и взаимосвязей между людьми подхватили специалисты в области социологии, психологии, антропологии, политологии, экономики.

I. АКТУАЛЬНОСТЬ ПРОБЛЕМЫ СБОРА ДАННЫХ

Современные социальные сети существенно изменили постановку вопроса — сегодня у исследователей имеются ресурсы для изысканий, а стремительное распространение социальных онлайн-сервисов и развитие технологий Больших Данных инициировали интерес к использованию сведений из социальных сетей в различных отраслях. Совместное использование структурных и контентных данных потенциально позволяет применять социальные сети для решения широкого круга бизнес-задач: борьбы с мошенничеством, управления брендом, рекламы товаров и услуг, формирования новых каналов сбыта и др.

II. СРАВНЕНИЕ СПОСОБОВ СБОРА ДАННЫХ ПОЛЬЗОВАТЕЛЕЙ СОЦСЕТЕЙ

Самый простой способ сбора данных — воспользоваться услугами специализированных компаний, собирающих и постоянно обновляющих данные из множества источников. Главное преимущество здесь — быстрота получения информации, что существенно при больших объемах клиентской базы и использовании различных социальных сетей. Недостаток — платная подписка на обновления данных.

Следующий способ — использовать программные интерфейсы, предоставляемые почти всеми популярными социальными сетями. Для различных сетей API отличаются набором до-

ступных данных, ограничениями на количество запросов и стоимостью доступа к интерфейсам. Например, если с помощью программного интерфейса сети «В контакте» можно получить полную информацию о пользователе, то Facebook предоставляет API, возвращающий практически «нулевые» сведения о пользователе. К недостаткам этого метода относится ограничение на количество одновременных запросов и на количество обращений, которые приложение может делать в единицу времени. Кроме того, необходимо постоянно отслеживать изменения в API и обновлять приложение по сбору данных, причем некоторые социальные сети предоставляют важные данные только на платной основе. Преимуществами метода являются возможность получения данных об одном пользователе в структурированной форме (JSON или XML), а также простота интеграции вызовов API в собственное приложение.

Еще один способ — ручной разбор веб-страниц социальных сетей, а также использование готовых краулеров для сбора данных с последующим разбором. В этом случае имеется доступ ко всем открытым данным и отсутствуют ограничения на скорость их сбора. К недостаткам следует отнести сложность реализации — веб-страница каждой социальной сети уникальна, поэтому каждый раз придется разрабатывать свои правила разбора. Недостатками являются также сложность поддержки и необходимость больших вычислительных ресурсов, правда, этот процесс хорошо распараллеливается.

III. ВЫВОДЫ

Каждый способ сбора данных пользователей соцсетей имеет свои плюсы и минусы. Таким образом, наиболее быстрым по времени реализации будет метод, использующий открытый API соцсети, однако данный метод может иметь ограничения по количеству обращений и типу сведений о пользователе.

1. Горчинская, О. Ю. Анализ данных социальных сетей / О. Ю. Горчинская // Открытые системы. СУБД. — 2015. — №3.

Стельмачёнок Максим Олегович, магистрант кафедры интеллектуальных информационных технологий БГУИР, stelmachenokm@gmail.com.

Научный руководитель: Бойко Игорь Михайлович, в. н. с. УП ГИС НАНБ, кандидат технических наук, igor_m_boyko@hotmail.com.

ПРОЕКТИРОВАНИЕ БИБЛИОТЕКИ ВИЗУАЛИЗАЦИИ ГРАФОВЫХ СТРУКТУР НА ОСНОВЕ ТРЕБОВАНИЙ ПОЛЬЗОВАТЕЛЯ

Рассматриваются основные аспекты в ходе проектирования библиотеки для визуализации графовых структур на основе шаблона визуализации, заданным пользователем на некотором декларативном языке.

ВВЕДЕНИЕ

Графовые структуры часто используются в разных сферах науки. Их визуализация упрощает последующий анализ. В зависимости от предметной области и целей анализа ставятся различные требования для их графического отображения. Среди таких требований может быть указана необходимость визуализации определенных уровней структуры некоторым конкретным способом. Однако существует мало инструментов графического отображения графов с заданной гибкой шаблона визуализации.

I. ПРОЕКТИРОВАНИЕ

Библиотека визуализации графовых структур будет позволять отображать требуемый граф на основе шаблона визуализации, заданного пользователем данного программного решения. Пользователь сможет задавать шаблон визуализации с помощью декларативного языка. Данный подход был выбран с целью отделить описание визуализации от описания графовой структуры.

Существует много готовых решений, которые могут качественно отображать граф, применяя какой-то определенный алгоритм визуализации. Поэтому решено было в качестве основы библиотеки использовать шаблон проектирования «Адаптер», используя существующие решения для возможности отрисовки конкретных частей графовых структур.

Пользователь данной библиотеки сможет задавать шаблоны визуализации с помощью определенных декларативных правил. Основными правилами будут являться:

- `get(graphSubsetDescription)`. В качестве аргумента будет передаваться объект, который будет описывать подграф к которому необходимо применить определенный алгоритм визуализации.
- `useLayout(layoutDescription)`. В качестве аргумента будет передаваться объект, который будет описывать алгоритм визуализации. Обязательным является указание

названия алгоритма. Если алгоритм имеет настраиваемые параметры, то их также можно будет указать в передаваемом объекте. В противном случае будут применены параметры по умолчанию для соответствующего алгоритма.

- `useFor(graphStructure)`. В качестве аргумента будет передаваться графовая структура, которую необходимо визуализировать.
- `export(exportType)`. В качестве аргумента будет передаваться формат файла, где будет сохранено конечное изображение графа.
- `build()`. Данный метод будет запускать процесс построения визуализации графа согласно ранее описанным правилам.

Процесс построения визуализации можно описать блок-схемой на рисунке 1.

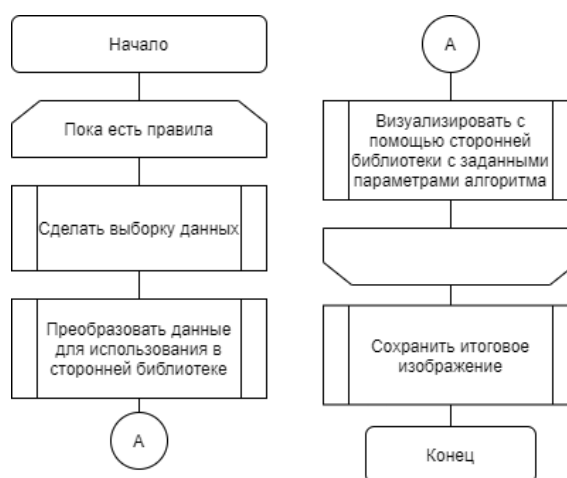


Рис. 1 – Блок-схема процесса построения визуализации

II. ВЫВОДЫ

В качестве языка программирования для реализации данной библиотеки будет использоваться язык JavaScript. Входные данные с описанием графовых структур будут задаваться в формате JSON.

Тарасенко Владислав Александрович, магистрант кафедры интеллектуальных информационных технологий БГУИР, u.tarasenka@yandex.by.

Научный руководитель: Сердюков Роман Евгеньевич, ведущий системный архитектор ООО «СтаркГеймз», кандидат технических наук, доцент, rserdyukov@gmail.com.

СРЕДСТВА АВТОМАТИЗАЦИИ РАЗРАБОТКИ МНОГОКРАТНО ИСПОЛЬЗУЕМЫХ КОМПОНЕНТОВ РЕШАТЕЛЕЙ ЗАДАЧ

В работе рассмотрен подход к построению средств автоматизации разработки многократно используемых компонентов решателей задач на основе Технологии OSTIS

ВВЕДЕНИЕ

Главным недостатком современных подходов к проектированию решателей задач является достаточно трудоёмкий и длительный процесс разработки. Одним из подходов к сокращению сроков и трудоемкости разработки решателей задач является использование библиотек многократно используемых компонентов (МИК) [2]. Однако, для поддержания совместимости и корректности компонентов в рамках библиотеки, а также корректного их применения во время разработки решателей задач, требуются специализированные средства автоматизации. В данной работе будет рассмотрен подход к созданию средств автоматизации разработки МИК решателей задач на основе Технологии OSTIS.

I. БИБЛИОТЕКА КОМПОНЕНТОВ РЕШАТЕЛЕЙ ЗАДАЧ ТЕХНОЛОГИИ OSTIS

В рамках Технологии OSTIS решатель задач представляет собой графодинамическую sc-машину, которая реализует агентный подход для решения поставленных задач [1].

Под *МИК OSTIS* понимается компонент некоторой интеллектуальной системы, который может быть использован в рамках другой интеллектуальной системы [3]. В свою очередь, под *МИК решателя задач* понимается компонент, соответствующий некоторому sc-агенту, который может быть использован в других системах, возможно, в составе более сложных неатомарных sc-агентов. Таким образом, набор совместимых компонентов объединяются в *библиотеку МИК решателей задач*. С использованием таких библиотек процесс разработки решателя задач сводится к интеграции между собой необходимого набора готовых МИК, хранящихся в библиотеке.

Трунц Виталий Владиславович, магистрант факультета информационных технологий и управления БГУИР, trunts.vitalij1110@gmail.com

Научный руководитель: Давыденко Ирина Тимофеевна, доцент кафедры Интеллектуальных информационных технологий БГУИР, кандидат технических наук, davydenko@bsuir.by

II. СРЕДСТВА АВТОМАТИЗАЦИИ РАЗРАБОТКИ МИК РЕШАТЕЛЕЙ ЗАДАЧ

Системы, которые построены на основе Технологии OSTIS, называются *ostis-системами*. [2] Архитектуру любой *ostis-системы* можно декомпозировать на *базу знаний*, *машину обработки знаний (МОЗ)*, *пользовательский интерфейс*.

Средства автоматизации разработки *МИК решателей задач* также является *ostis-системой*. Основным компонентом такой системы, который реализует процесс автоматизации разработки МИК решателей задач является МОЗ, которая, в свою очередь состоит из соответствующих sc-агентов. Примерами sc-агентов МОЗ таких средств являются: *sc-агент сборки многократно используемого компонента*, *sc-агент поиска используемых scr-программ внутри sc-агента*. Преимуществом такого подхода является представление всех компонентов с помощью унифицированной семантической сети, что позволяет интегрировать любые МИК в различные *ostis-системы* путём добавления их в общую sc-память, что существенно сокращает трудоемкость интеграции МИК в разрабатываемую систему.

1. Голенков В. В., Хорошевский В. Ф. Представление и обработка знаний в графодинамических ассоциативных машинах. – Мн. : БГУИР, 2001.
2. Шункевич Д. В., Давыденко И. Т. Средства поддержки компонентного проектирования систем, управляемых знаниями (OSTIS-2015): материалы V Междунар. научн.-техн. конф. – Мн.: БГУИР, 2015
3. Шункевич Д. В. Машина обработки знаний интеллектуальной метасистемы поддержки проектирования интеллектуальных систем (OSTIS-2014): материалы IV Междунар. научн.-техн. конф. – Мн.: БГУИР, 2014.

АНАЛИЗ ПОКАЗАТЕЛЕЙ ПРИМЕНЯЕМЫХ ДЛЯ ОЦЕНКИ АЛГОРИТМОВ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

В статье рассматриваются показатели, применяемые для оценки качества работы алгоритмов сегментации изображений. Даны рекомендации по выбору показателей для оценки алгоритмов при различных целях сегментации.

ВВЕДЕНИЕ

Разработка методов сегментации с высокой точностью является важной задачей при обработке изображений. Поэтому оценка точности алгоритмов сегментации имеет большое значение, и является вопросом методологии оценки.

I. ПОКАЗАТЕЛИ ДЛЯ ОЦЕНКИ АЛГОРИТМОВ СЕГМЕНТАЦИИ

Выделим следующие группы показателей:

- показатели на основе пространственного перекрытия между двумя сегментациями[1];
- показатели основанные на сравнении объема сегментированной области с эталоном;
- показатели основанные на базовых понятиях из теории информации, таких как энтропия и взаимная информация;
- показатели основанные на представлении сегментации как распределения и статистических методах сравнения этих распределений;
- показатели основанные на суммировании расстояния между всеми парами пикселей при сравнении двух сегментаций[2].

II. ТРЕБОВАНИЯ К АЛГОРИТМАМ СЕГМЕНТАЦИИ

В зависимости от цели сегментации могут быть специальные требования к алгоритмам сегментации:

- в некоторых случаях важным требованием является то, что сегментированная область

включает в себя весь действительный сегмент, независимо от того, включает ли в себя ложно сегментированные части. В таких случаях стоит использовать показатели на основе пространственного перекрытия между двумя сегментациями;

- когда алгоритмы должны охватывать объем близкий к объему истинного сегмента настолько возможно. В этом случае используют показатели основанные на сравнении объема сегментированной области с эталоном;
- когда общая форма и выравнивание важны, например, когда цель сегментации идентифицировать повреждения и дать оценку размера. В этом случае следует использовать расстояние Махаланобиса.

ЗАКЛЮЧЕНИЕ

Рассмотренные виды показателей позволяют производить оценку качества алгоритмов при различных целях сегментации изображений. Выбор одного или нескольких показателей требует анализа требований предъявляемых к оцениваемым алгоритмам.

Список литературы

1. Zou, K. Statistical validation of image segmentation quality based on a spatial overlap index / K. Zou [et al]. // Acad Radiol. – 2004. – Vol. 11, no. 2. – P. 178–189.
2. Di Gesu, V. Distance-based functions for image comparison / V. Di Gesu, V. Starovoirov // Pattern Recognition Letters. – 1999. – Vol. 20 – P. 207–214.

Хоменко Артем Сергеевич, магистрант 1 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, arnewmail@yandex.ru.

Научный руководитель: Ивашенко Валерьян Петрович, доцент кафедры интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, ivashenko@bsuir.by

ИССЛЕДОВАНИЕ ПАРАМЕТРОВ ИНТЕРНЕТ-МАГАЗИНА, ВЛИЯЮЩИХ НА ЕГО ЭФФЕКТИВНОСТЬ

Рассматриваются теоретические аспекты, связанные с эффективностью интернет-магазина. Выделяются факторы, влияющие на эффективность.

ВВЕДЕНИЕ

Для определения эффективности интернет-магазина лучше всего использовать такой параметр магазина, как конверсия. Конверсия - отношение числа посетителей сайта, выполнивших на нём какие-либо конверсионные действия (покупку, регистрацию, подписку, переход по рекламной ссылке, посещение определённой страницы), к общему числу посетителей сайта, выраженное в процентном соотношении [1].

В связи с вышесказанным, опираясь на существующие теоретические и практический разработки, можно выделить факторы, влияющие на конверсию магазина.

I. ФАКТОРЫ, ВЛИЯЮЩИЕ НА КОНВЕРСИЮ ИНТЕРНЕТ-МАГАЗИНА

Дизайн. Это визуальное оформление сайта, а именно различные графические элементы, шрифты, цветовые гаммы. Дизайн продающего сайта определяет насколько пользователю будет приятно находиться на данном сайте.

Контент. Это наполнение сайта. Оно может быть текстовым, графическим или мультимедийным. Очень важно, чтобы информация на сайте была достаточно краткой, но в то же время и полезной, так же грамотно написана. Положительно скажется наличие уникальных фотографий и видеороликов о товарах.

Юзабилити. Это удобство и простота использования сайта для целей посетителей. Важно наличие простой и понятной структуры сайта, удобного меню и т.д. Сайт должен быть таким, чтобы у посетителей, при поиске товаров не возникало никаких сложностей в получении результата.

Быстрота. Необходимо оптимизировать сайт так, чтобы он загружался максимально быстро, иначе пользователь просто закроет сайт ещё до его загрузки. Так же, чем сайт лучше оптимизирован, тем он выше в поисковых системах.

Универсальность. Важно, чтобы сайт правильно отображался на различных устройствах,

будь то персональный компьютер, телефон или планшет, так же важна поддержка всех современных браузеров.

Товар. Товар, реализуемый через интернет-магазин должен быть не только интересен покупателю, но и качественно представлен и пропрекламирован.

Цена. Цена должна быть адекватной т.к. никто не купит товар в данном магазине, если у конкурента есть такой же, но дешевле.

Дефицитность товара. Если товар достаточно редкий и его сложно найти и купить в обычном магазине, то покупатель с большей охотой закажет его в интернете.

Уникальность товара. Если продаваемый товар обладает какой-либо уникальной особенностью, которой нет у других подобных товаров, то он привлечёт больше внимания покупателей.

Реклама. Реклама является одним из важнейших факторов увеличения конверсии т.к. это то, как и где пользователь увидит магазин и перейдёт ли на него. Существует множество видов рекламы, например: контекстная, тизерная, таргетированная и т.д.

II. ВЫВОДЫ

В статье были выделены основные факторы, влияющие на эффективность интернет-магазина, все они в равной степени важны и отвечают за привлечение и удержание клиента, а так же за то, какие эмоции он получит от покупки именно в данном магазине и вернётся ли снова. Так же исходя из исследования, можно заметить, что на эффективность влияют не только качество товара и то, как его преподносят, но и технические особенности магазина, что не всегда берётся во внимание.

1. Айзенберг, Б. Добавьте в корзину: ключевые принципы повышения конверсии веб-сайта / Б. Айзенберг, Дж. Айзенберг // М.:Мани, Иванов и Фербер. – 2011. – №2. – С. 517.
2. Круг, С. Как сделать сайт удобным: юзабилити по методу Стива Круга / С. Круг // СПб.: Питер. – 2011. – №2. – С. 210.

Шабатько Виталий Борисович, магистрант кафедры интеллектуальных информационных технологий БГУИР, extrem332@mail.ru.

Научный руководитель: Романов Владимир Ильич, кандидат технических наук, доцент, rom@newman.bas-net.by.

ТРЕБОВАНИЯ К МОДЕЛИ ОБУЧАЕМОГО В АДАПТИВНЫХ ОБУЧАЮЩИХ СИСТЕМАХ

Рассматриваются требования к модели обучаемого в адаптивных обучающих системах, которые необходимо учитывать для обеспечения персонализации процесса обучения.

ВВЕДЕНИЕ

Одним из необходимых компонентов адаптивных обучающих систем является модель обучаемого. Под моделью обучаемого понимают модель, которая хранит некоторую информацию об обучаемом в виде его характеристик, достаточную для обеспечения адаптации процесса обучения к этому обучаемому [1].

I. ТРЕБОВАНИЯ К МОДЕЛИ ОБУЧАЕМОГО

Рисунок 1 изображает взаимосвязь между моделированием обучаемого и адаптацией [2].

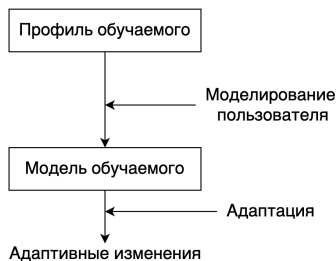


Рис. 1 – Взаимосвязь между моделированием обучаемого и адаптацией

Для определения набора характеристик, которые необходимо учитывать для обеспечения персонализации процесса обучения, проведен анализ требований к подготовке контингента студентов,

обучающихся в технических вузах, их различий с точки зрения начальной подготовки, форм и способов обучения, и, соответственно целей и задач обучения.

Анализ показал, что студенты вузов, особенно на младших курсах, в силу разных причин могут сильно отличаться по уровню начальных знаний, полученных в школах, училищах, техникумах. Поэтому многие студенты имеют проблемы при изучении некоторых дисциплин на младших курсах.

Обучение в рамках некоторых направлений отличается «ранней специализацией», то есть образовательные стандарты включают ряд дисциплин, изучаемых на младших курсах, которые считаются базовыми для выбранной специальности. Не все студенты имеют достаточный теоретический уровень подготовки для изучения этих дисциплин.

Шаплыко Илья Михайлович, магистрант кафедры интеллектуальных информационных технологий БГУИР, shaplyko.ilya@gmail.com.

Научный руководитель: Захаров Владимир Владимирович, кандидат технических наук, доцент, доцент кафедры ИИТ, zvv@open.by.

Специфика учебного процесса во многих вузах предполагает обучение не только русскоговорящих студентов, но и иностранных. Иностраным студентам сложнее понимать специфические термины и определения различных курсов, соответственно для этой группы студентов необходим особый подход к формированию содержания и форме представления учебного материала. Материал должен быть адаптирован для целевых категорий разделяемых на русскоговорящих и иностранцев.

Существенную роль играют также психологические особенности, обусловленные типом мышления. Обучаемые с разным типом мышления по-разному воспринимают информацию, причем роль играет не только сама информация, но и способы и последовательности ее представления.

II. Выводы

Таким образом, на основе анализа можно определить требования к модели обучаемого. Модель должна:

- содержать данные об обучаемом, достаточные для выбора стратегии обучения в начале процесса обучения;
- учитывать изменение уровня знаний пользователя в процессе обучения;
- позволять формировать содержания и формы представления информационных элементов предметной области конкретному обучаемому;
- позволять применять различные технологии адаптации для наиболее адекватного отображения предметного курса в зависимости от его специфики.

Список литературы

1. Brusilovsky, P. Methods and techniques of adaptive hypermedia / Peter Brusilovsky // Springer. – 1998. – P. 1-43.
2. Nguyen, Loc. Learner model in adaptive learning / Loc Nguyen, Phung Do // World Academy of Science, Engineering and Technology. – 2008. – Vol. 45, no. 70. – P. 395-400.

ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПРЕОБРАЗОВАНИЯ ПОИСКОВОГО ЗАПРОСА НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Рассматривается информационный поиск. Предлагается использование моделей машинного обучения для упрощения формирования запроса к БД.

ВВЕДЕНИЕ

Для данной задачи предлагается использовать два последовательно применённых алгоритма: первый решает задачу классификации (для определения типа искомого объекта), второй задачу извлечения именованных сущностей (для установления соответствия искомым слов атрибутам соответствующего типа). Далее предлагается построить объект, с помощью которого осуществляется запрос к БД;

I. ОПИСАНИЕ ПРОЦЕССА ПОИСКА

Имея запрос пользователя на естественном языке, необходимо понять, объект какого типа он пытается найти, то есть решить задачу классификации. Для этого предлагается использовать векторное представление.

Векторное представление (Word Vector Embedding) – сопоставление словам из словаря векторов из R^n для n , значительно меньшего количества слов в словаре [1].

Определённый тип искомой сущности содержит свой набор атрибутов и для того, чтобы определить соответствие атрибутов словам из поискового запроса на естественном языке, предлагается использовать определение именованных сущностей.

Определение именованных сущностей (Named Entity Recognition – NER) – получение структурированного представления информации по тексту на естественном языке, в нем содержащейся [2].

Для создания моделей необходима обучающая выборка, которую предлагается брать из БД, в которой хранится искомая информация в уже структурированном и упорядоченном виде: таблицы соответствуют типам, а поля и их значениям.

Результаты работы моделей предлагается использовать для формирования запроса к БД посредством ORM, используя оператор подобия.

Юхневич Павел Витальевич, магистрант кафедры интеллектуальных информационных технологий БГУИР, yukhnevichpavel@gmail.com.

Научный руководитель: Сердюков Роман Евгеньевич, доцент кафедры интеллектуальных информационных технологий БГУИР, кандидат технических наук, rserdyukov@gmail.com

II. РЕЗУЛЬТАТ РАБОТЫ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Для демонстрации предлагаемого подхода был проведён эксперимент, результаты которого (табл.1.) наглядно показывают работу упомянутых выше моделей. Была выбрана предметная область, содержащая 3 класса сущностей, общим объёмом 500 объектов, при этом обучение осуществлялось на 100 объектах.

Таблица 1 – Результат работы моделей

Запрос	Классификация	NER
Android 10	smartphone	os: Android, os_version: 10
notebook xiaomi	notebook	type: notebook, brand: xiaomi
xiaomi	smartphone	brand: xiaomi
apple invention	news	text: apple invention

Полученные результаты (например, результат классификации по запросу «xiaomi» и «xiaomi notebook») указывает на то, что в первом случае, определённый тип чаще других содержит данное ключевое слово, во втором же случае тип является другим, так как пользователь задал его явно в запросе.

Примечателен также и последний запрос, который мог бы быть отнесён к классу «smartphone», но был отнесён к классу «news», так как слово «invention» встречается там и не встречается в «smartphone».

III. ВЫВОДЫ

Предлагаемый подход позволяет определить тип искомой сущности и отобразить слова из поискового запроса на соответствующие атрибуты данного типа до обращения в базу данных, тем самым снизив число запросов.

1. Журавлев, Ю. И. Распознавание. Математические методы / Ю. И. Журавлев, В. В. Рязанов, О. В. Сенько // Информатика. – 2006. – №2. – С. 30-40.

СРЕДСТВА АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТОВЫХ ОБРАЩЕНИЙ В СЛУЖБУ ПОДДЕРЖКИ

Рассматриваются основные подходы и способы автоматической обработки текстовых обращений в службу поддержки, а также их особенности.

ВВЕДЕНИЕ

Служба поддержки — понятие, обобщающее собой и охватывающее множество услуг, посредством которых предприятия и организации обеспечивают помощь пользователям. Служба поддержки – структура, которая требует постоянного внимания и контроля. Лишь действующий по определенным требованиям и установкам сервис может считаться эффективным и действенным. С учетом «человеческого» фактора и прочих аспектов важно выполнить максимальную автоматизацию процессов.

I. АКТУАЛЬНОСТЬ ПРОБЛЕМЫ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТОВЫХ ОБРАЩЕНИЙ В СЛУЖБУ ПОДДЕРЖКИ

Под автоматизацией применительно к службе поддержки принято считать использование выверенного, настроенного и отлаженного программного обеспечения (ПО), которое позволяет выполнять определенные задачи.

Первая линия службы поддержки — это элемент структуры контактного центра службы поддержки, цель которого — идентификация клиента и его проблемы для дальнейшей её обработки.

Практика показывает, что около 50–70% обращений решаются именно на первой линии службы поддержки, следовательно использование специального ПО для автоматизация обработки данных обращений позволит существенно сократить затраты на содержание службы поддержки, а также минимизирует риск ошибки связанный с «человеческим» фактором.[1]

II. СПОСОБЫ ОБРАБОТКИ ТЕКСТОВЫХ ОБРАЩЕНИЙ В СЛУЖБУ ПОДДЕРЖКИ

Самый простой способ автоматизации контактного центра службы поддержки - поиск по ключевым словам. Он заключается в поиске в обращении клиента заранее указанных ключевых или шаблонных слов, на которые уже известен ответ. Данный вариант реализации может быть полезен для обработки очень простых обращений, например, когда клиент не знает, как выпол-

нить ту или иную операцию и ему нужна простая и быстрая консультация.

Более сложный способ реализации - ответ на открытый вопрос. Данный вариант автоматизации позволяет найти точный ответ на любой вопрос в каком-либо источнике данных, например, в руководстве пользователя к программному продукту, или по базе часто задаваемых вопросов. Для реализации данного метода может быть использован полнотекстовый поиск по базе данных.

Ещё один способ - построение целеориентированного диалога. Эта реализация позволяет вести диалог с клиентом для достижения определённой цели. Основное преимущество данной реализации – можно объединить вопросы из различных тем и с различными целями в одном модуле, тем самым получив некоторую универсальную вопросно-ответную систему для решения сразу нескольких задач. Для реализации данного способа может использоваться нейросетевая модель sequence2sequence.[2] Базовая sequence-to-sequence модель состоит из двух рекуррентных нейронных сетей: кодер, который обрабатывает входные данные, и декодер, который генерирует данные вывода. Главное преимущество данной модели состоит в том что она позволяет достичь высокого качества отображения большого множества схожих последовательностей (набора вопросов) в другое множество (в набор ответов).

III. ВЫВОДЫ

Использование специального ПО для автоматизации обработки обращений пользователя в службу поддержки позволит существенно сократить затраты на её содержание. Для автоматизации обработки запросов могут быть использованы как традиционные методы поиска информации, такие как полнотекстовый поиск, так и методы нейросетевой обработки запроса, при которых пользователь может вести осмысленный диалог с системой.

1. Madeline, Bodin. The Call Center Dictionary / Madeline. Bodin // Software. – 2002.
2. Russell, Stuart. Artificial Intelligence: A Modern Approach / Russell. Stuart // Deep Learning. – 2013.

Якимцов Иван Дмитриевич, магистрант каф. ИИТ, ivan.yakimtspv@gmail.com.

Научный руководитель: Крапивин Юрий Борисович, доцент кафедры интеллектуальных информационных технологий БГУИР, ybox@list.ru.

Секция "Системы управления"

Председатель: канд. техн. наук, доцент, зав. каф. СУ Марков А.В.
Члены жюри: канд. тех. наук, проф. Лукьянец С.В.
канд. тех. наук, доцент Хаджинов М.К.
Секретарь инженер Дежурко Е.Н.

ПРИМЕНЕНИЕ КОДОВ «DATA MATRIX» ДЛЯ МАРКИРОВКИ ПРОДУКЦИИ. СИСТЕМА КОНТРОЛЯ КАЧЕСТВА ПЕЧАТИ

Увеличение объемов производства неизбежно ведет к росту конкуренции на рынке товаров, и как следствие, появлению недобросовестных производителей. Для передачи конечным пользователям информации о поставщике, сырье, территории изготовления, маршрутах перевозки, дате производства, партии продукции, а также шифрования множества других параметров, появилась необходимость создать инструмент, который позволит собрать все необходимые данные в компактном виде. В работе будет рассмотрен один из типов кодирования данных – Data Matrix код. Также будут описаны компоненты системы управления производством, позволяющие проверять качество сгенерированного и нанесенного кода.

В настоящее время на производстве для кодирования данных о продукте используются коды различных типов. Различают две основные группы штриховых кодов – одномерные (линейные) и двумерные. Среди кодов второго типа наибольшую популярность приобрел код «Data Matrix» (DM) – двумерный матричный штрихкод, состоящий из светлых и темных участков (квадратов). По сравнению с одномерными кодами двумерные позволяют хранить больший объем информации, занимают меньше места и считываются даже при частичном повреждении кода. [1] QR- и DM- коды могут исправлять ошибки согласно алгоритму Рида-Соломона.

QR-коды имеют четыре уровня восстановления поврежденных данных: уровень L (низкий) 7% кодовых слов могут быть восстановлены; уровень M (средний) 15%; уровень Q (квартиль) 25%; уровень H (высокий) 30%. Более высокий уровень может быть выбран для неблагоприятной среды, где код, скорее всего, испачкается. Чем выше уровень, тем выше способность исправлять ошибки, но тем больший геометрический размер на наносимой поверхности будет иметь QR-код.

С другой стороны, во всех версиях DM-кодов возможность исправления ошибок достигает более 33%, что выше, чем у QR-кодов (до 30%). Следовательно предполагается, что коды DM более безопасны и надежны. [2]

Данный тип кодов часто используют при маркировке небольших предметов, например микросхем, а также в пищевой, оборонной промышленности, рекламе и других сферах.

При создании Data Matrix кодов используется сложная арифметика полей Галуа и код Рида-Соломона. Матрица состоит из двух частей: шаблона поиска и закодированных данных. Размер матрицы прямо пропорционален размеру входных данных. Вокруг кода обязательно должна быть свободная зона, отделяющая код от остального изображения.

Путем использования высокоуровневого и низкоуровневого кодирования по определенным

алгоритмам происходит кодирование данных. Код разбивается на составные элементы - квадраты размером 3x3 ячейки без правого верхнего угла. После чего квадраты располагаются в определенном порядке (в зависимости от длины кодируемого сообщения). Далее происходит преобразование данных в двоичном виде и заполнение поля кода.

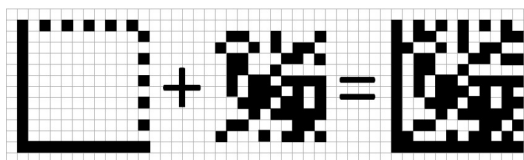


Рис. 1 – Схема образования Data Matrix кода

Печать DM-кодов может производиться различными способами:

- 1. лазерное нанесение непосредственно на продукцию;
- 2. ударно-точечные маркировочные системы.

После нанесения маркировки товар поступает на следующие этапы производства, где может быть подвержен термическому, химическому, физическому воздействию. Во избежание поступления товара с некачественной маркировкой применяют системы технического зрения, позволяющие определить уровень качества печати – верификаторы.

Одним из производителей компонентов для проверки качества печати является компания «Omron». В номенклатуре оборудования данного поставщика присутствует большое количество готовых решений, которые активно тестируются и внедряются на производствах.

Система для считывания штрих-кодов состоит из камеры технического зрения, контроллера технического зрения, специальной подсветки и датчика метки. В качестве дополнений можно использовать линзы для лучшей фокусировки. На Рис. 2 приведена структура компонентов

системы технического зрения считывания и распознавания кодов.

Ведущую роль в обработке данных выполняет контроллер технического зрения. Это устройство оснащено операционной системой Windows 10, имеет различные порты входов/выходов для подключения периферийных устройств, таких как датчики, камеры, коммутаторы, панели операторов.

Для обнаружения и фотографирования кодов ДМ применяют камеру технического зрения. Расположение камеры напрямую зависит от размеров кодов и типа материала этикетки. В случае, если этикетка выполнена из блестящего материала, может возникнуть отражение и преломление света, которое затруднит считывание кода. Для устранения бликов и придания лучшего изображения применяется подсветка.

Скорость обработки зависит от размера захватываемой области изображения. Чем меньше площадь фото, тем с большей скоростью можно получать изображения. При помощи специального программного обеспечения можно настраивать область обнаружения ДМ-кодов согласно требованиям.

Системы технического зрения на базе компонентов «Omron» поддерживают большинство распространенных, быстрых и современных промышленных протоколов передачи данных, таких как: Industrial Ethernet, EtherCAT и т.д. Также

одним из важных достоинств является то, что компания «Omron» предоставляет единую платформу для программирования и параметризации компонентов систем управления - «Sysmac», преимущество которой является простая интеграция систем технического зрения в уже эксплуатируемые системы промышленной автоматизации [3]. Таким образом, применение на производстве ДМ кодов позволяет сократить ошибки при маркировке продукции на 3-5%. Соответственно уменьшается процент отбракованных изделий, что при массовом масштабе производства, характерном для пищевой промышленности, весьма существенно. Данный фактор дает возможность предприятиям сэкономить значительные финансовые средства. А системы технического зрения, построенные на основе современной компонентной базы средств автоматизации (например, от компании «Omron»), гарантируют ещё более существенное уменьшение отбраковки производимой продукции.

1. Волхонский В. В. Системы контроля и управления доступом. Штриховые коды. – СПб: Университет ИТМО, 2015. – 53 с.
2. Trent Munro. Databatrix и QR-коды: почему 2D-коды все еще актуальны в 2016 году./Trent Munro//URL: <http://blog.matthews.com.au/databatrix-and-qr-codes-why-2d-codes-are-still-relevant-in-2016/> – 7.08.2016.
3. Каталог продукции Sysmac / OMRON. – [P072-RU2-05A+SysmacCatalogue2015], 2015. – 356 с.

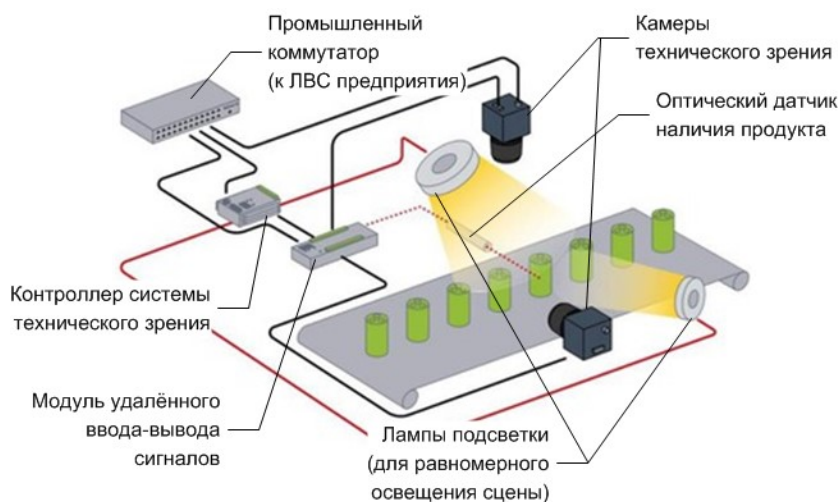


Рис. 2 – Компоненты системы технического зрения системы контроля качества печати маркировки

Архипенко Яна Сергеевна, Парамонова Алиса Егоровна, студенты кафедры систем управления БГУИР, gr622401@mail.ru, .

Научный руководитель: Захарьев Вадим Анатольевич, кандидат технических наук, доцент кафедр систем управления БГУИР, zahariev@bsuir.by.

АЛГОРИТМИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ С РАЗЛИЧНОЙ АРХИТЕКТУРОЙ

Рассматриваются алгоритмы функционирования информационной системы с монолитной и микросервисной архитектурой применительно к задачам организации автоматизированных информационных систем на примере системы продажи проездных документов на железнодорожный транспорт.

ВВЕДЕНИЕ

При создании сложных автоматизированных информационных систем (приобретение проездных документов на различные виды транспорта, а также документов на посадочные места в концертные залы и стадионы и т.п.) важными этапами являются разработка структуры и алгоритмов функционирования таких систем, что влияет на эффективность, производительность, надежность и удобство эксплуатации применяемых аппаратных и программных средств. В настоящей работе исследуется информационная система приобретения проездных документов на железнодорожный транспорт.

В состав системы входит несколько модулей: регистрации пользователей; регистрации пассажиров пользователя; оформления проездных документов на поезда; оформления проездных документов на электрички; оплаты проездных документов; справочной информации.

При построении системы продажи проездных документов необходимо учитывать особенности монолитной и микросервисной архитектуры.

I. МОНОЛИТНАЯ СИСТЕМА

Для системы с монолитной архитектурой характерно нахождение всех модулей системы в одном проекте с общей кодовой базой. Основным достоинством такой системы является удобство в работе: настройка одного модуля, пусть и большого, что проще и быстрее по сравнению с настройкой приложения с микросервисной архитектурой, включающей набор отдельно настраиваемых различных модулей. Основным недостатком в сравнении с микросервисной архитектурой является сложность распределения ресурсов между модулями, так как система с монолитной архитектурой содержит их внутри себя, что не позволяет распределять ресурсы между модулями. В ряде случаев это играет ключевой фактор при построении крупных систем и сильно влияет на отказоустойчивость.

Схема алгоритма функционирования системы продажи документов на железнодорожный транспорт с монолитной архитектурой изображена на рисунке 1.

Когда пользователь отправляет запрос, система через API-шлюз принимает его и обраба-

тывает соответствующими модулями. Если, например, пользователь оформляет проездной документ на поезд, то вся система обрабатывает его запрос, при этом выполняя функции, соответствующие модулю оформления проездных документов на поезда, а остальные модули в это время не работают ввиду отсутствия необходимости, и недоступны извне, так как система обрабатывает конкретный запрос. Конечно же модули взаимодействуют между собой, поэтому после оформления проездного документа, пользователь будет перенаправлен на оплату оформленного заказа, или, к примеру, для оформления заказа система предложит варианты транспорта по некоторым критериям поиска из модуля справочной информации. Здесь стоит отметить преимущество монолитной архитектуры в прямом доступе к остальным модулям. И в дальнейшем при обработке данных по оплате заказа система будет задействована, используя модуль оплаты проездных документов, и в это время остальные модули не используются и не доступны извне.



Рис. 1 – Схема алгоритма функционирования системы с микросервисной архитектурой

Управление ресурсами такой системы происходит следующим образом: что бы система могла обрабатывать несколько запросов одновременно, аппаратное обеспечение выполняет обработку данных в нескольких параллельных

процессах. Тогда для создания отдельного процесса для обработки запроса другого пользователя в отдельном потоке запускается копия системы. Если модуль оплаты заказов используется в два раза чаще каждого из модулей оформления проездных документов, потому что оплачивать необходимо оба вида проездных документов, то для поддержания стабильности работы ресурсы необходимо рассчитывать так, чтобы самый нагруженный модуль системы выдерживал нагрузку и обрабатывал стабильно все запросы пользователей.

II. МИКРОСЕРВИСНАЯ СИСТЕМА

Для системы с микросервисной архитектурой характерна самостоятельность каждого из модулей, так как фактически они являются отдельными приложениями. Это позволяет настраивать и распределять ресурсы для каждого модуля в отдельности. Основным и важнейшим преимуществом такой системы является увеличение отказоустойчивости, когда при недоступности или выходе из строя одного модуля все остальные продолжают работать в своем предопределённом настройкой режиме. Для больших систем с множеством различного функционала зачастую этот фактор может являться наиболее значимым. Схема алгоритма функционирования системы с микросервисной архитектурой изображена на рисунке 2.



Рис. 2 – Схема алгоритма функционирования системы с монолитной архитектурой

Система принимает пользовательский запрос и через API-шлюз направляет его на обработку соответствующему модулю. Таким образом, если пользователь направляет запрос на

оформление проездного документа на поезд, то его запрос поступит на соответствующий модуль, а все остальные модули системы будут доступны и могут обрабатывать запросы других пользователей. При этом модули могут взаимодействовать между собой, и не важно на одном ли физическом устройстве они находятся, поскольку взаимодействуют они как независимые приложения посредством сети Интернет.

Достоинством такой системы является возможность настройки каждого модуля. При ситуации, когда, к примеру, высоконагруженный модуль не справляется с потоком запросов, появляется возможность увеличения производительности такого модуля методом создания копии.

Копии могут находиться на различных аппаратных средствах. К примеру, в регионе, где интенсивность потока запросов выше, установить две копии модуля на более мощном сервере, а в регионе с меньшей интенсивностью – две копии на более слабом серверном оборудовании. В общем случае это уже особенности настройки, но показать ее гибкость важно, потому что это одно из ключевых преимуществ системы с микросервисной архитектурой.

III. ВЫВОДЫ

Иногда именно благодаря настройке принципа функционирования можно значительно увеличить эффективность системы, а микросервисную систему можно более гибко настраивать, так как каждый ее модуль – самостоятельный. Появляется возможность создать копию отдельного модуля, а не полностью всей системы. При этом увеличивается отказоустойчивость системы. Это становится важным фактором для крупных систем с большим набором различных функций, которые удобно разнести в отдельные модули.

Окончательные рекомендации по эффективности рассмотренных структур могут быть даны в результате моделирования системы с учетом конкретных исходных данных относительно размерности системы, интенсивности входящих информационных потоков и возможностей используемых ресурсов.

1. Крис Ричардсон. Микросервисы. Паттерны разработки и рефакторинга / К.Ричардсон – Изд-во: Издательский дом "Питер", Санкт-Петербург, 2016. – 544 с.
2. Сэм Ньюмэн. Создание микросервисов / С.Ньюмэн. – Изд-во: Издательский дом "Питер", Санкт-Петербург 2016. – 304 с.
3. Интернет-ресурс <https://habr.com/ru/>

Бортник Артем Владимирович, магистрант кафедры информационных технологий автоматизированных систем, ttemmik97@gmail.com

Научный руководитель: Лукьянец Степан Валерьянович, кандидат технических наук, профессор, Lukyanetsstv@gmail.com

МЕТОДЫ ПОСТРОЕНИЯ ИНФОРМАЦИОННЫХ ДИАЛОГОВЫХ СИСТЕМ

Рассматриваются стандартные структуры и основные компоненты диалоговых систем и основные подходы к их построению.

ВВЕДЕНИЕ

Диалоговые системы - неотъемлемая составляющая интеллектуальных компьютерных систем, служащих для переработки информации. Основным требованием диалоговых систем является обеспечение более удобной и естественной формы взаимодействия интеллектуальных систем с пользователями.

1. Виды диалоговых систем

По назначению можно выделить три основных типа диалоговых систем: общего назначения (general, чат-боты), задачеориентированные (task-oriented) и способные вести диалог на любую тему (open domain). Упрощенные версии диалоговых систем, такие как чат-боты имеют довольно простую архитектуру и состоят из набора правил и заранее подготовленных ответов.

Современные интеллектуальные диалоговые системы имеют более сложную архитектуру, решают конкретную задачу пользователя и, как правило, взаимодействуют с внешним хранилищем данных.



Рис. 1 – Структура целевой диалоговой системы

Системы общего назначения не пытаются решить конкретную задачу. Их цель в том, чтобы поддержать разговор с пользователем на произвольную тему. Язык разметки базы знаний может включать в себя паттерны вопросов и соответствующие им шаблоны ответов, также предысторию диалогов к ним и название соответствующей темы общения.

II. Подходы к построению диалоговых систем

Существует два основных подхода к построению диалоговых систем — генеративный и по-

исковый. В генеративном подходе ответ на сообщение пользователя порождается с помощью некоторой модели.

В настоящее время наиболее популярный способ построения таких систем — seq2seq-модели. Такие модели состоят из двух рекуррентных сетей: encoder-decoder. Coder строит представление входной последовательности слов. Далее полученное представление (последние выход и значение ячейки сети) копируются в decoder. По полученному представлению decoder пытается восстановить целевую последовательность слов.

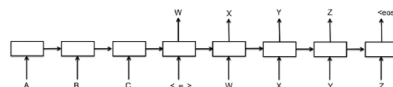


Рис. 2 – Архитектура sequence-to-sequence модели

В поисковом (селективном) подходе ответ на сообщение пользователя выбирается из большого набора готовых ответов. Как правило, для сообщения и всех ответов строятся векторные представления одной размерности, после чего ответ выбирается в соответствии с некоторой метрикой.

В селективных диалоговых моделях для каждой пары вопрос – ответ вычисляется некоторая ранжирующая функция, значение которой тем выше, чем более релевантна текущая реплика диалоговому контексту.

III. Выводы

Таким образом, в докладе рассмотрена стандартная структура диалоговой системы, её основные компоненты и основные подходы к построению систем. Диалоговые системы общего назначения являются более простыми в реализации, однако с их помощью не может быть решено такое количество задач пользователя, как с помощью более сложных систем.

1. Building Advanced Dialog Managers for Goal-Oriented Dialog Systems. 2018

Беленькая Анна Ильинична, магистрант каф. СУ, annabelenkaya897@gmail.com.

Научный руководитель: Захарьев Вадим Анатольевич, доцент кафедры систем управления БГУИР, кандидат технических наук, доцент, zahariev@bsuir.by.

РАСПОЗНАВАНИЕ ОБРАЗОВ С ИСПОЛЬЗОВАНИЕМ СИАМСКОЙ НЕЙРОННОЙ СЕТИ

В работе описываются результаты исследования сиамской нейронной сети. Предлагаемый подход предлагается как альтернатива сверточной нейронной сети, в условиях ограниченного объёма обучаемых данных. Также открыт вопрос использования данного подхода в системах реального времени, в системах распознавания лиц и аутентификации. Описанию исследований этого подхода и посвящён данный тезис.

ВВЕДЕНИЕ

В системах распознавания лиц мы хотим, чтобы мы могли распознавать человека посредством одной фотографии. В случае, если система не может распознать изображение, это означает, что изображение человека не сохраняется в базе данных. Для решения данной проблемы использование только сверточной нейронной сети не возможно, поскольку сверточные нейронные сети не работают на малом тренировочном наборе, а также необходимо обучать модель снова, когда в системе появляется новый человек.

I. СИАМСКИЕ НЕЙРОННЫЕ СЕТИ

Для решения данной проблемы можем использовать сиамскую нейронную сеть. Главной целью сиамской нейронной сети является определение, насколько похожи два изображения.

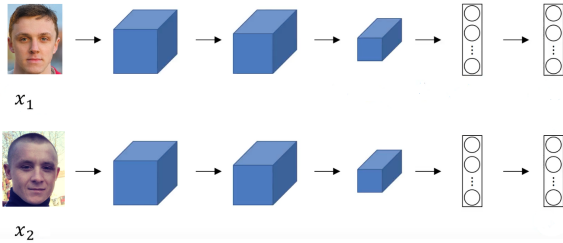


Рис. 1 – Преобразование входных изображений в векторы x_1 и x_2

Чтобы сравнить два изображения, необходимо вычислить расстояние d между векторами этих изображений.

$$d(x_1, x_2) = \|f(x_1) - f(x_2)\|^2 \quad (1)$$

Если это расстояние меньше порогового значения, это означает, что на изображениях одно и то же лицо.

II. ВЫВОД МАТЕМАТИЧЕСКОЙ МОДЕЛИ РАСПОЗНАВАНИЯ ОБРАЗОВ

Для того, чтобы получить конечный вектор из входного изображения, необходимо узнать входные параметры для этого изображения. Для

этого применим градиентный спуск к функции потери триплета. Используя три изображения: корневое (A), позитивное (P , тот же человек, что и корень), а также негативное (N), необходимо найти расстояния $d(A, P)$, и чтобы оно было меньше или равно расстоянию $d(A, N)$. Функция потери триплета минимизирует расстояние между изображением A и P , и максимизирует расстояние между изображением A и N . В этом случае модель может научиться создавать одинаковое кодирование для разных изображений, что означает, что расстояние между разными изображениями будут равны нулю. Для решения данной проблемы добавим α (вес), и чтобы всегда существовал разрыв между позитивным и негативным изображением.

$$d(A, P) + \alpha \leq d(A, N) \quad (2)$$

Вид функция потери триплета:

$$L(A, P, N) = \max(d(A, P) - d(A, N) + \alpha, 0) \quad (3)$$

Тогда функция стоимости – это сумма всех индивидуальных потерь в разных триплетах из всего набора тренировок.

$$COST = \sum_{i=1}^n L(A_i, P_i, N_i) \quad (4)$$

Тренировочный набор должен включать несколько изображений одного и того же человека, чтобы иметь пары A и P , а затем, после обучения, появится возможность распознавать людей посредством одной фотографии.

III. ВЫВОДЫ

Предлагаемый подход по распознаванию образов позволяет точно распознавать лица на фотографии. Данный подход может использоваться в системах реального времени, распознавания людей на фотографиях, аутентификации.

1. Advance AI: Face recognition using Siamese networks [Электронный ресурс]. – <https://mc.ai/advance-ai-face-recognition-using-siamese-networks>

Буяльский Олег Андреевич, магистрант кафедры систем управления БГУИР, oleg.buyalski@gmail.com.

СИСТЕМА НАВЕДЕНИЯ ЛАЗЕРНОГО ЛУЧА

Рассматривается система наведения лазерного луча, состоящая из систем слежения и фокусировки луча ВЭЛ.

ВВЕДЕНИЕ

Лазерное оружие и, в частности, высокоэнергетическое лазерное (ВЭЛ) оружие рассматриваются для различного военного применения в отношении разнообразных платформ, таких как космические, воздушные и наземные системы, и то лишь немногие примеры. Данное оружие, как правило, предполагает использование лазера или другого источника высокоэнергетического пучка электромагнитного излучения для отслеживания и уничтожения намеченной цели. Для достижения целей операции, лазерное оружие должно точно отслеживать намеченную цель и держать луч ВЭЛ на цели до получения ожидаемого результата.

Таким образом, основная цель: система слежения должна обеспечивать обнаружение, захват и сопровождение цели, система фокусировки — наведение лазерного луча на цель и удержание пучка силового лазерного излучения на определенной области объекта.

I. СИСТЕМА СЛЕЖЕНИЯ

Система слежения включает в себя: следящий телескоп, связанный с корпусом, причем следящий телескоп включает в себя датчик слежения для приема электромагнитного излучения, отраженного от воздушной цели; источник электромагнитного излучения для генерирования луча ВЭЛ; дополнительное зеркало для приема электромагнитного излучения от источника и его отражение на основное зеркало луча ВЭЛ через корпус.

Частью системы является процессор, связанный с датчиком слежения и управляющим контроллером луча ВЭЛ, причем процессор обрабатывает первую и вторую части луча ВЭЛ, полученные датчиком для определения взаимосвязи между сопровождающим лучом и лучом ВЭЛ.

Процессор генерирует управляющий сигнал для управления луча ВЭЛ на воздушную цель, основываясь на определенной взаимосвязи.

II. СТРУКТУРНАЯ СХЕМА И ОПИСАНИЕ СИСТЕМЫ СЛЕЖЕНИЯ

Схема установки приведена на рисунке 1.

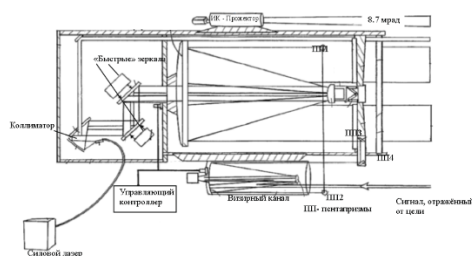


Рис. 1 – Схема установки

Изобретение позволяет использовать лазерный луч, интегрированный в ВЭЛ, который следует по тому же оптическому пути, и отбирается в двух радиальных положениях после отражения от основного зеркала расширителя луча. Когда расширитель луча сфокусирован на цели, единственное пятно диска Эйри, если оно сформировано на датчике слежения, и вектор смещения разлагается путем подсчета количества пикселей от центра поля до центра распадающегося пятна. Когда расширитель луча фокусируется на конечных диапазонах, образуются два пятна диска Эйри и вектор смещения определяется путем измерения количества пикселей от точки, равноудаленной между центрами пятна и центром поля.

Система стабилизации далее дополняется использованием гироскопических инструментов для измерения угловой скорости как камеры слежения, так и основного зеркала.

Фильтр Калмана может использоваться для обработки измерений гироскопов и оптического выравнивания, чтобы сформировать широкополосную базу отсчета с ограниченным дрейфом и низким уровнем шума.

Структурная схема процесса слежения представлена на рисунке 2.

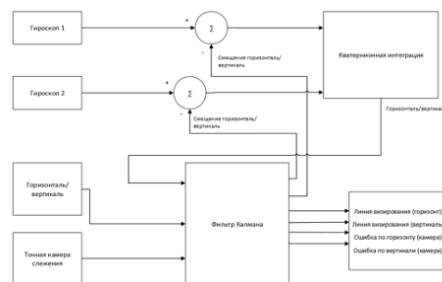


Рис. 2 – Структурная схема процесса слежения

Гироскопы 1 и 2, состоящие из трех элементов, являются основными инструментами для поддержания ориентации датчика по лазерному

каналу. Гироскоп 1 используется для управления выравниванием луча ВЭЛ.

Гироскоп 2 установлен на следящем телескопе и используется для его управления. Сигналы с гироскопов включают в себя тангаж, рыскание и вращение основного зеркала. Сигналы суммируются, соответственно, со сдвигами по тангажу, рысканию и вращению и вводятся в кватернионный интегриционный блок, который представляет, соответственно, инерциальную линию визирования основного зеркала и датчик слежения. Затем сигналы преобразуются в горизонтальную и вертикальную пары, которые выводятся из кватернионного блока и вводятся в фильтр Калмана. Поскольку эти данные искажаются смещениями значений угловой скорости гироскопа, что присуще механизмам, из которых гироскопы сконструированы, фильтр Калмана сравнивает эти значения по горизонту и вертикали с измерениями горизонта и вертикали из блока «горизонталь/вертикаль», которые, как правило, не имеют смещения значений угловой скорости.

III. АЛГОРИТМ ПОИСКА ЦЕЛИ

Поиск цели по реперным точкам происходит по следующему алгоритму: две пары пентапризм формируют реперные периферийные пучки, по которым осуществляется контроль фокусировки луча ВЭЛ на требуемую дистанцию и корректируется наведением на цель.

Энергетическая ось сфокусированного излучения соответствует средней точке между пятнами реперных пучков в поле ТВК.

Прицеливание осуществляется совмещением энергетической оси лазерного излучения с заданным участком цели при помощи быстродействующих зеркал.

Быков Вадим Олегович, студент кафедры систем управления БГУИР, vadimbykov2011@gmail.com

Томашевич Яна Юрьевна, студентка кафедры систем управления БГУИР, tomashevichyana@gmail.com

Научный руководитель: Назаренко Павел Николаевич, кандидат физико-математических наук, ОАО «Пеленг»

Быстродействующие зеркала обеспечивают стабилизацию диаграммы направленности излучения на выход объектива коллиматора и сканирование энергетической оси сфокусированного лазерного излучения относительно визионной оси ТВК.

Принцип фокусировки отображен на рисунке 3.

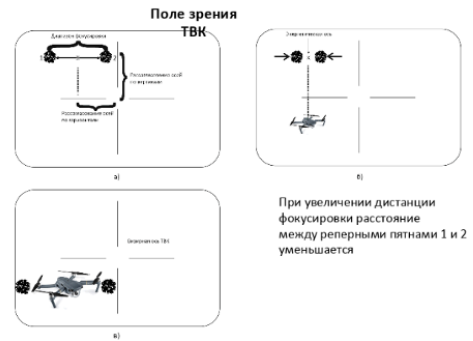


Рис. 3 – Фокусировка лазерного луча по двум реперным точкам

IV. ЗАКЛЮЧЕНИЕ

Таким образом, представлена схема установки и структурная схема системы слежения, которая является основной составляющей системы наведения лазерного луча. Показано, как осуществляется поиск цели по двум реперным точкам.

1. Федоров Б. Ф. Лазеры. Основы устройства и применение.— Издательство ДОСААФ СССР,1988г – 190 с
2. Интернет-ресурс <https://topwar.ru/86572-vyzhiganie-celi-razvitie-vysokoenergeticheskikh-lazerov.html>

СОВМЕСТИМЫЕ СИСТЕМЫ ДОКУМЕНТООБОРОТА

Автоматизация все большего числа процессов и задач в организациях требует интеграции между все большим и большим числом информационных систем, используемых в организациях. Информационные системы обладают разным набором средств интеграции и выбор конкретного способа интеграции зависит от множества факторов: трудоемкости реализации и поддержки работоспособности интеграционных механизмов, требований информационной безопасности, необходимой скорости передачи данных и т.д.

ВВЕДЕНИЕ

Одним из побудительных факторов интеграции систем электронного документооборота (СЭД) с другими приложениями и системами является необходимость поддержки разнообразных бизнес-процессов, когда СЭД используется совместно с другими бизнес-приложениями.

Другой фактор — это необходимость синхронизации нормативно-справочной информации (НСИ), включая учетные записи пользователей, организационно-штатную структуру предприятия, каталоги продукции, справочники контрагентов, адресный регистр и многое другое. Чтобы работать согласованно, все корпоративные приложения должны пользоваться одними и теми же справочными данными, а для этого системы должны уметь этими данными обмениваться.

Нельзя забывать и о взаимодействии с внешним миром. При внедрении СЭД в госсекторе нужна интеграция с системой межведомственного документооборота государственных органов Республики Беларусь (СМДО), в других задачах может понадобиться работа с электронной почтой, при участии в электронных торгах — обмен документами с торговой площадкой и т. д.

Поэтому наличие развитых возможностей интеграции следует считать обязательным требованием к системе электронного документооборота, и это надо учитывать при выборе.

Далее будут рассмотрены способы интеграции СЭД DIRECTUM с иными корпоративными системами.

I. КОННЕКТОР К TELNET-СЕРВЕРУ

Одним из простых средств для интеграции систем внутри организации может являться использование протокола Telnet (сетевое протокола для реализации текстового интерфейса по сети) как достаточно общего, двунаправленного, восьмибитного байт-ориентированного средства связи. Использование протокола Telnet оптимально для настройки интеграции (передачи текстовых данных в условиях отсутствия необходимости в шифровании данных) DIRECTUM с информационными системами, уже использующими Telnet-сервер. Использование Telnet прекрасно подходит для гетерогенных сетей, по-

скольку опирается на концепцию сетевого виртуального терминала (Network Virtual Terminal, NVT).

Таким образом, техническое решение «Коннектор к Telnet-серверу» предназначено для интеграции системы DIRECTUM и других информационных системы организации посредством использования протокола Telnet.

Техническое решение «Коннектор к Telnet-серверу» позволяет настроить DIRECTUM и Telnet-клиент для подключения к Telnet-серверу для задействования всех возможностей протокола Telnet (передачи любых текстовых сообщений, команд, и т.д.) из DIRECTUM.

Техническое решение состоит из Telnet-клиента и компонент DIRECTUM, которые обеспечивают необходимые универсальные настройки для работы с Telnet-клиентом: открытие соединения, авторизацию, закрытие соединения.

Основные компоненты решения:

- Telnet-клиент, с которым устанавливает соединение система DIRECTUM и через который отправляются сообщения в другие информационные системы (на Telnet-сервер);
 - компоненты DIRECTUM.
- Состав и формат данных для передачи определяется для конкретных задач интеграции DIRECTUM и других информационных систем.

II. КОННЕКТОР К SAP PI 3.0

Эффективность электронного документооборота во многом зависит от возможности наладить взаимодействие ЕСМ-системы с другими решениями, используемыми в организации.

ERP-системы являются наиболее часто используемым классом систем, с которыми требуется интеграция.

SAP ERP — мировой лидер среди систем управления ресурсами предприятия. Настройка качественной синхронизации данных между SAP ERP и DIRECTUM является важной бизнес-задачей, стоящей перед многими организациями.

Благодаря интеграции систем исключается дублирование процессов, ускоряется обработка данных и генерация отчетов.

Техническое решение «Коннектор к SAP PI 3.0» предназначено для обмена данными меж-

ду системами ЕСМ-класса DIRECTUM и ERP-класса SAP и позволяет:

- синхронизировать справочную информацию между системами DIRECTUM и SAP;
- отправлять документы DIRECTUM в систему SAP;
- наладить связь объектов SAP и документов DIRECTUM;
- открывать карточки объектов системы SAP из DIRECTUM.

Решение состоит из двух частей. Первая разработана на платформе IS-Builder и основана на возможностях системы DIRECTUM, реализованных в модуле Набор средств интеграции (DIRECTUM Integration Toolset). Вторая часть разработана на платформе SAP NetWeaver PI, и содержит пакет с объектами SLD, пакет с данными Integration Repository и пакет с данными Integration Directory.

Данный коннектор подразумевает несколько способов передачи данных. Если инициатором в процессе интеграции является система DIRECTUM, то взаимодействие может происходить как через веб-сервисы SAP PI, так и через файловую систему или ftp-сервер. Если инициатором в процессе интеграции является система SAP PI, то взаимодействие происходит через веб-сервисы интеграции DIRECTUM.

Опишем процесс передачи данных из DIRECTUM в SAP. Сперва DIRECTUM передает информацию коннектору. Далее коннектор преобразует данные для передачи к комплексному типу. После преобразования данных функции коннектора вызывают метод системы SAP PI с помощью SOAP-запроса для передачи данных в систему SAP PI. В завершении система SAP PI преобразует полученный пакет данных в IDOC и загружает в систему SAP.

Процесс обратной передачи данных из SAP в DIRECTUM сильно схож с предыдущим. В начале система SAP передает пакет с данными (IDOC) в SAP PI. После чего SAP PI преобразует IDOC к комплексному типу. По завер-

шению преобразования SAP PI вызывает метод веб-сервисов интеграции DIRECTUM с помощью SOAP-запроса для передачи данных в систему DIRECTUM. Завершающим этапом веб-сервисы интеграции DIRECTUM загружают полученные данные в систему DIRECTUM.

III. ЗАКЛЮЧЕНИЕ

Коннектор к Telnet-серверу является универсальным решением, которое позволяет создать условия для использования возможностей протокола Telnet. В каждом конкретном случае, например, для интеграции DIRECTUM и других информационных систем через протокол Telnet, должны быть выполнены другие необходимые действия: определены правила формирования сообщений, правила обработки ответов. Решение использует базовые модули системы DIRECTUM и не требует настройки взаимодействия с другими модулями.

Интеграция систем SAP и DIRECTUM, предоставляя возможность максимально удобной и прозрачной работы с объектами обеих систем, позволяет сделать еще один шаг в построении единого информационного пространства предприятия.

Список литературы

1. Макаров С. Особенности интеграции СЭД с другими корпоративными системами [Электронный ресурс] // Режим доступа: <https://www.tezisdok.ru/blog/osobennosti-integratsii-sed-s-drugimi-korporativnymi-sistemami> – Дата доступа: 12.04.2020.
2. Шемсединов Т. Интеграция информационных систем [Электронный ресурс] // Режим доступа: <https://habrahabr.ru/post/117468/> – Дата доступа: 12.04.2020.
3. Коннектор к Telnet-серверу [Электронный ресурс] // Режим доступа: <https://www.directum.ru/solutions/11441471> – Дата доступа: 12.04.2020.
4. Коннектор к SAP PI 3.0 [Электронный ресурс] // Режим доступа: <https://www.directum.ru/solutions/3820390> – Дата доступа: 12.04.2020.

Локтевич Кирилл Михайлович, магистрант кафедры электронных вычислительных средств Белорусского государственного университета информатики и радиоэлектроники, lokkiirill@gmail.com.

Научный руководитель: Павлова Анна Валентиновна, ученый секретарь кафедры систем управления Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, доцент, pavlova@bsuir.by.

ZIGBEE/IEEE 802.15.4 КАК ВАРИАНТ СЕТЕВОГО ШЛЮЗА ДАТЧИКА-ИНДИКАТОРА СОСТОЯНИЯ ЭЛЕКТРИЧЕСКОЙ СЕТИ И ПОТРЕБИТЕЛЯ ЭЛЕКТРИЧЕСКОЙ ЭНЕРГИИ

Основой для связи любых технических устройств служит сетевой шлюз. Сетевой шлюз может работать по различным протоколам и на разной технической базе. В каждом есть свои недостатки и достоинства. В данной статье показан принцип реализации сети на основе протокола Zigbee/IEEE802.15.4 устройств индикации нормального состояния объекта управления (потребителей электрической энергии), определение основных неисправностей, как в сети энергоснабжения, так и в самом потребителе электрической энергии [1].

ВВЕДЕНИЕ

Так как на большинстве объектах в электроснабжении, используемых в настоящее время, уже смонтированы распределительные шкафы с традиционными автоматическими выключателями, то для решения задач управления требуется контроль состояния того или иного объекта для создания компьютеризированных, автоматизированных систем управления. Зачастую это связано с задачами диспетчеризации [1].

Использование датчика-индикатора напряжения и датчика-индикатора тока в качестве индикатора отсутствия нагрузки на автоматическом выключателе дает нам общую картину о ситуации с линией электроснабжения. Зачастую одна линия электроснабжения может иметь несколько ответвлений. Поэтому чтобы увидеть реальную неисправность конкретного устройства требуется использование датчиков-индикатора тока на выходных точках. Для этого необходимо организовать связь. Проводная связь, зачастую, бывает не очень эффективна из-за сложности конструкций здания. В данном случае идеальным решением выступает беспроводная сеть Zigbee, выбор которой и будет обоснован, и рассмотрен в данной работе.

I. ПОСТАНОВКА ЗАДАЧИ ДЛЯ ПОСТРОЕНИЯ БЕСПРОВОДНОЙ СЕТИ

Электрощиты расположены на некотором расстоянии друг от друга, как и сами потребители, требуется обеспечить связь между несколькими узлами для анализа работоспособности технических средств, каждого в отдельности, не прибегая к проводной связи. Установлен исполнительный микроконтроллер или ПЛК - программируемый логический контроллер, который аккумулирует все данные, сигнализирует о ошибках и принимает решения. Требуется производить мониторинг работы данного оборудования с ПК или смартфона. Обеспечить надежность работы сети. Пример реализации поставленной задачи (см.рис.1.)

II. ПРИМЕР СТРУКТУРНОЙ СХЕМЫ СИСТЕМЫ УПРАВЛЕНИЯ И МОНИТОРИНГА ЭЛЕКТРИЧЕСКИХ УЗЛОВ ПРЕДПРИЯТИЯ С ИСПОЛЬЗОВАНИЕМ ДАТЧИКОВ – ИНДИКАТОРОВ НА ОСНОВЕ БЕСПРОВОДНОЙ СЕТИ ZIGBEE

Zigbee – протокол верхнего уровня, работающий на стандарте IEEE802.15.4. Сеть Zigbee конструктивно реализована на элементах: координатор, роутер и конечные устройства. Роутеры создают узлы сети, которые позволяют наращивать протяженность сети и количество ее элементов. Координатор – главный узел, который создает сеть и выбирает канал работы. Координатор определяет канал свободный от помех, и ожидает запросы на подключение. Конечное устройство с датчиком, имеет низкое энергопотребление и работает от батарейки. На перечисленных выше элементах показан пример построение сети Zigbee (см.рис.1).

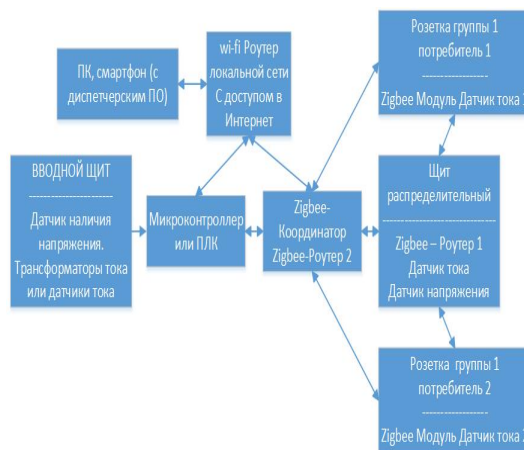


Рис. 1 – Пример структурной схемы системы управления и мониторинга электрических узлов предприятия с использованием датчиков – индикаторов на основе беспроводной сети Zigbee

Из рисунка 1 видно передача данных состояния Zigbee датчика-индикатора тока потребителя 1 и потребителя 2 поступает с разных то-

чек на Zigbee-роутер, который устанавливается рядом с распределительным щитом. Роутер является ретранслятором данных с датчиков он питается от розетки, с подключенным резервным автономным источником, который сработает при отключении сети. Далее передача происходит от Zigbee-роутера 1 к координатору с Zigbee-роутером 2, который напрямую или через шлюзы, подключается с управляющим терминалом – ПК, смартфоном или контроллером).

В блоке где расположен координатор, как видно из рисунка 1, стоит Zigbee-роутер 2 - это сделано для большей надежности. Благодаря ячеистой топологии, сеть Zigbee имеет способность к самовосстановлению и гарантированной доставке пакетов в случаях обрыва связи между отдельными узлами. Так при обрыве одного соединения, создается обходной маршрут, и сигнал дойдёт до получателя по новому пути, так как между узлами существует прямая связь, а не только связь с «центром». Это достигается благодаря специальным алгоритмам маршрутизации сети ZigBee. Такой метод надежный и он используется в самой сети интернет [3].

Микроконтроллер или ПЛК установлен рядом с вводным щитом, в котором также производится анализ с датчиков. Полученные данные микроконтроллер или ПЛК сравнивает с начальной и конечной точкой где и происходит оценка присутствия или отсутствия аварии на определенном узле. Эти данные передаются через wi-fi роутер (с доступом в интернет), как по локальной сети так и через Интернет на ПК или смартфон.

Начальный этап настройки сети Zigbee построен следующим образом. Первоначальным звеном в формировании сети выступает координатор, который определяет радиоканал без содержания помех и ожидает запросы от разных Zigbee устройств. После получения запроса происходит обмен сообщениями. Далее подключение может быть организовано от устройства к устройству, используемое для обмена данными.

Следует учесть также, что у Zigbee сети есть два варианта присоединения процедура MAC ассоциации и повторное сетевое присоединение.

Процедура MAC ассоциации заключается в следующем. Подключаемое устройство отправляет запрос маяк, который попадает в конечную точку. После получения запроса, управляющим устройством, производится сканирование сети и определяется рациональность подключения. Далее - отправка запроса управляющим устрой-

ством, о присоединении, с предъявлением своих прав со значением «повторное присоединение». Когда ответ будет получен, управляющее устройство вышлет адрес. Надо учитывать тот факт, что в MAC ассоциации данные передаются не зашифрованными.

Повторное сетевое присоединение организовано на сетевом уровне. Оно актуально также и для первого присоединения. Устройство, которое имеет сетевой ключ допускается для безопасного обмена пакетами. Ключ также присваивается при настройке. Устройство, которое присоединяется посылает запрос присоединения и обменивается с подключающим устройством пакетами запрос-ответ. Все организуется на сетевом уровне[3].

III. Выводы

Построение беспроводной сети реализовано на протоколе Zigbee так как он имеет ряд преимуществ:

- передача информации продолжается при неисправности одного из узлов благодаря ячеистой (mesh) топологии сети;
- низкое энергопотребление в режиме приёма и передачи;
- длительное время работы на источнике питания;
- высокая чувствительность приёмника и программируемая выходная мощность передатчика [2];
- высокий уровень подавление помех увеличивает надежность в передаче пакетов.

В качестве модулей Zigbee отличное решением это использование чипов Texas Instruments серии CC25xx. Поддержка ZigBee/ZigBee PRO, ZigBee RF4CE, 6LoWPAN, и все решения, основанные на 802.15.4., бесплатное ПО IEEE 802.15.4 MAC и стек Z-Stack™, признанный одним из лучших в отрасли [2].

1. Информационные технологии и системы 2019 (ИТС 2019) : И74 материалы международной научной конференции, БГУИР, Минск, Беларусь, 30 октября 2019 г. – Information Technologies and Systems 2019 (ITS 2019) : Proceeding of The International Conference, BSUIR, Minsk, 30th October 2019/ редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2019. – С. 66-67.
2. Руководство по беспроводному подключению Texas Instruments – май 2011 года – С. 8.
3. Сети ZigBee. Зачем и почему? [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/155037/> Дата доступа: 10.04.2020.

Малашенков Анатолий Александрович, магистрант кафедры электронных вычислительных средств БГУИР, antlik2018@gmail.com.

Научный руководитель: Шемаров Александр Иванович, кандидат технических наук, доцент кафедры электронных вычислительных средств БГУИР, shemarov@bsuir.by.

АЛГОРИТМ АВТОМАТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ТЕМПА РЕЧИ

В работе рассматривается возможность решения задачи определения темпа речи посредством сегментации сигнала и использования частотного анализа сигнала.

ВВЕДЕНИЕ

В работе под характеристикой темпа речи понимается количество слогов внутри просодической единицы на единицу времени. В качестве такого анализируемого фрагмента, как правило, выступает фонетическое слово. Данным термином называется отрезок речевой цепи, объединяемый одним (словесным) ударением, который произносится как единое целое («на улице», «в университете»). Ограничение просодической единицей необходимо для уменьшения влияния пауз в речи.

Задачу идентификации слогов можно свести к задаче определения гласных в речевом потоке. Главное отличие гласных от согласных фонем – наличие резонансной составляющей спектра. Таким образом определение гласных можно свести к анализу резонансных частот спектра гласных, называемых формантами. Форманты гласных русского языка были определены экспериментально во множестве исследований [1] и их можно использовать для создания математической аппроксимации человеческой речи.

I. ПОСТРОЕНИЕ СИНТЕЗАТОРА РЕЧИ

Для моделирования сигнала, близкого к гласным воспользуемся методами линейного предсказания речи [2]. Для симуляции гласных используется комбинация синтезатора частот с рекурсивным фильтром. Для симуляции других звуков – белый шум пропущенный через этот же фильтр. Передаточная функция фильтра будет иметь вид:

$$T(z) = \frac{1}{1 - H(z)} = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}}, \quad (1)$$

где a_k – коэффициенты фильтра, получаемые методом минимизации квадрата ошибки сравнения полученного сигнала с реальными образцами речи.

Схема фильтра представлена на рисунке 1.

II. РАСПОЗНАВАНИЕ ГЛАСНЫХ

Для анализа спектра полученного сигнала воспользуемся библиотекой SFS [3]. В ней реализован метод получения Формантных частот

методом линейного предсказания речи и использует методы автокорреляции и весовые функции для анализа. На выходе получаем матрицу содержащую частоты первых трех формант и амплитуды, что позволяет составить гистограмму и отметить на ней частотное распределение для различных формант и гласных. Из них также можно сделать вывод о ошибочном определении фильтра гласных, анализируя перекрытия гистограмм.

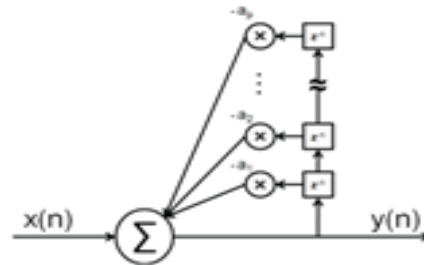


Рис. 1 – Схема БИХ-фильтра для моделирования гласных звуков речи

III. ВЫВОДЫ

Процесс определения темпа речи может рассматриваться, как состоящий из серии статистически независимых событий детектирования гласных, что дает возможность складывать, а не умножать ошибки от факта неверного определения. В результате экспериментов была достигнута точность определения гласных около 80%. Таким образом возможно определение темпа речи путем частотного анализа на основе линейного предиктивного программирования.

1. Информационные процессы, Том 4 №2 / Сорокин В. Н., Цыплихин А.И. – Институт проблем передачи информации, Российская академия наук, Москва, Россия, 2004.
2. Recognition of Vowels in Continuous Speech by Rising Formants / FACTA UNIVERSITATIS (NIS)// SER.: ELEC. ENERG. vol. 23, no. 3, December 2010, 379-393.
3. Официальная документация SFS [Электронный ресурс]. Режим доступа: <https://phon.ucl.ac.uk/resource/sfs/help/intex.php> Дата доступа: 22.03.2020.

Петрович Андрей Юрьевич, магистрант кафедры систем управления Белорусского государственного университета информатики и радиоэлектроники, petrovichandrey17@gmail.com.

Научный руководитель: Захарьев Вадим Анатольевич, доцент кафедры систем управления Белорусского государственного университета, кандидат технических наук, zahariev@bsuir.by.

СИСТЕМЫ УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ ЖИЛЫХ ДОМОВ

Энергопотребление зданий и жилых домов за последние 30 лет увеличилось более чем на 30%, и это количество продолжает расти. Различные исследования доказали, что управление энергопотреблением помогает экономить электроэнергию и значительно сокращать счета за эти услуги. Уже скоро мы можем столкнуться с проблемами недостатка ресурсов и роста стоимости энергии, поэтому надлежащее управление энергопотреблением станет вынужденной необходимостью.

ВВЕДЕНИЕ

Системы энергоменеджмента (СЭМ) используются в энергетическом секторе уже давно, в отличие от систем домашнего энергоменеджмента (СДЭМ), которые начали активно развиваться только в последнее десятилетие. Основная цель использования таких систем в том, чтобы позволить потребителю контролировать и отслеживать количество потребляемой энергии, или потреблять ее более эффективным способом.

I. ОСНОВНЫЕ КОМПОНЕНТЫ

К основным компонентам СДЭМ можно отнести:

- сенсорные и измерительные устройства — различные типы датчиков, которые необходимы для определения параметров: температура, влажность, уровень освещенности, движение, наличие дыма или огня; интеллектуальные счетчики, позволяющие отслеживать цены в реальном времени и оптимально планировать потребление в часы пиковой нагрузки;
- умные приборы — бытовые приборы, имеющие встроенные интеллектуальные функции и позволяющие обмениваться информацией с системой домашнего энергоменеджмента;
- пользовательский интерфейс — устройство, с которым пользователь может взаимодействовать и получать актуальную информацию о энергопотреблении, состоянии оборудования и аналитические отчеты;
- программное обеспечение (ПО) и информационные и коммуникационные технологии (ИКТ) — СДЭМ использует ПО, которое отвечает за принятие решений, а также за оптимальное планирование потребления. ИКТ является связующим звеном между СДЭМ и ее устройствами, на основе различных протоколов связи.

Романюк Иван Анатольевич, магистрант кафедры электронных вычислительных средств Белорусского государственного университета информатики и радиоэлектроники, baho0707@gmail.com.

Научный руководитель: Марков Александр Владимирович, заведующий кафедрой систем управления Белорусского государственного университета, доктор технических наук, доцент, markov@bsuir.by.

II. КЛАССИФИКАЦИЯ

СДЭМ классифицируют по следующим особенностям:

- функциональность: информационные (ограниченные только представлением информации) и управляющие, которые помимо представления информации, могут управлять оборудованием;
- пользовательский интерфейс: на основе типа и метода взаимодействия с пользователем. Наиболее популярный метод — с помощью сенсорного экрана;
- используемое оборудование;
- программное обеспечение.

III. ВЫВОДЫ

Установка СДЭМ — беспроигрышная ситуация как для предприятий, так и для частных домов. В настоящее время рынок СДЭМ очень быстро развивается, и общественность все больше и больше осознает их преимущества. Поставщики подобных систем должны лучше понимать меняющиеся потребности и опасения потребителей, а также сделать внедрение и использование более простым и доступным. Вполне вероятно, что следующий энергетический кризис вызовет всплеск спроса на подобные решения.

Список литературы

1. J. I. Lee. A study on the use cases of the smart grid home energy management system / C. S. Choi, W. K. Park, J. S. Han and I. W. Lee // ICTC 2011, Seoul. — 2011. — P. 746–750.
2. M. Amer. Smart home energy management systems survey / A. Naaman, N. K. M'Sirdi, A. M. El-Zonkoly // Renewable Energies for Developing Countries (REDEC), 2014 International Conference on, Beirut. — 2014. — P. 167–173.
3. B. Asare-Bediako. Home energy management systems: Evolution, trends and frameworks / W. L. Kling and P. F. Ribeiro // 2012 47th International Universities Power Engineering Conference (UPEC), London. — 2012. — P. 1–5.

ДИНАМИЧЕСКОЕ ПЛАНИРОВАНИЕ ТРАЕКТОРИИ КОЛЛАБОРАТИВНОГО РОБОТА

Представлен метод динамического планирования траектории коллаборативного робота – манипулятора, который реализует построение оптимальной траектории в процессе выполнения производственных задач, а также учитывает особенности кинематики как при планировании так и во время движения.

ВВЕДЕНИЕ

Совместная деятельность робота и человека в производственной ячейке какого-либо технологического процесса предполагает сложный алгоритм для задач управления, оптимизации траектории робота с учетом движений человека, в динамически изменяющейся рабочей среде. В данном случае стандартные алгоритмы и методы не подходят, так как не способны учитывать изменение времени выполнения операций роботом, вызванное необходимостью модифицировать, адаптировать движения робота для обеспечения безопасности человека при совместной работе. Известные методы планирования траектории либо обладают высокой вычислительной сложностью, либо не позволяют найти траекторию, оптимальную по длине пути, что не позволяет использовать их в системах, где необходимо планирование траектории в режиме реального времени с условием минимизации перемещения.[1]. Чтобы решить эту проблему, некоторые авторы [2] придерживались иерархического интегрированного подхода, который опирается на четкое различие между задачами и реализацией планирования движения.

1. АЛГОРИТМ ДИНАМИЧЕСКОГО ПЛАНИРОВАНИЯ ТРАЕКТОРИИ

Представленный алгоритм планирования траектории робота - манипулятора учитывает возникновение временных и пространственных ограничений и препятствий и устранение неопределенности, вызванной изменением положения человека. Ключевой особенностью этой работы является моделирование ожидаемого поведения человека на разных уровнях абстракций, связанных с временной неопределенностью и кинематикой как во время планирования траектории, так и во время выполнения. Разработан алгоритм планирования траектории коллаборативного робота, обеспечивающий адаптацию поведения робота к задачам человека в динамике, а также гарантирующий его безопасность.

На рисунке 1 показаны основные модули, составляющие предлагаемую структуру, и последовательность шагов, реализующих интегрированный подход управления.



Рис. 1 – Схема интеграции процессов планирования траектории и перепланирования во время выполнения

Во-первых, рассматриваемый производственный процесс анализируется с целью выявления возможных сценариев коопераций человека и робота (этап 1). На данном этапе идентифицируются совместные задачи, необходимые для реализации производственного процесса, ресурсы, которые могут выполнять эти задачи (человек, робот или одновременно), и эксплуатационные ограничения (например, ограничения приоритета или синхронизации). Затем для всех возможных совместных задач робота, то есть одновременных задач человека и робота, вычисляется набор траекторий робота (этап 2). В частности, планировщик движения отвечает за со-здание и выполнение траекторий робота и гарантирует безопасность оператора-человека, регулирующего скорость робота. Он опирается на оперативный и статистический анализ, способный идентифицировать объем, занимаемый человеком с определенной вероятностью во время выполнения заданий (Human Occupancy Volume-NOV) [3]. Учитывая NOV, характеризующийся различной вероятностью занятости, планировщик движения генерирует для каждой пары одновременных задач человек-робот набор возможных траекторий. Различные траектории, по которым робот может следовать, будут входить в объем, занимаемый человеком на разных уровнях, поэтому он характеризуется различным временем выполнения и допустимым интервалом. Идентифицированные задачи робота и человека вместе с соответствующей временной информацией (время выполнения и ее изменчивость) кодируются в модели временного планирования (этап 3). Эта информация позволяет планировщику задач характеризовать временную неопределенность, касающуюся фактической продолжительности задач человека и

робота. Учитывая эту модель, планировщик задач генерирует гибкий во времени план (этап 4), координирующий действия робота и человека, выбирая наиболее подходящие траектории в соответствии с ожидаемой рабочей средой. Планировщик задач синтезирует гибкие планы, учитывая временную неопределенность. Затем исполнитель плана выполняет его (этап 5), учитывая изменение траектории робота и человека. Надежность выполнения плана достигается за счет временной гибкости и механизма перепланирования, которые позволяют контроллеру адаптировать / изменять план и поведение робота в соответствии с фактическим поведением человека. Выбранные траектории робота (этап 6) выполняются планировщиком движения, который также реализует низкоуровневое разделение скорости и мониторинг изменения рабочей среды, чтобы избежать столкновений с человеком (этап 7).

II. СИСТЕМА ДИНАМИЧЕСКОГО ПЛАНИРОВАНИЯ ТРАЕКТОРИИ МАНИПУЛЯТОРА

Система динамического планирования траектории манипулятора должна решать следующие задачи: постановка задач, которые должен выполнить человек и робот; выбор наиболее подходящей траектории движения робота из набора траекторий; работа с временной неопределенностью во время генерации и исполнения плана; мониторинг исполнения и, в случае необходимости, управление возможными сбоями посредством перепланирования.

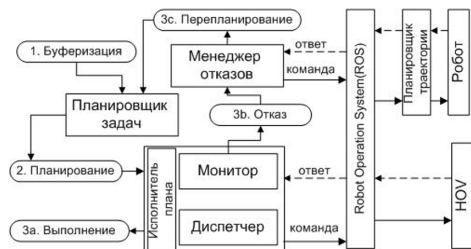


Рис. 2 – Архитектура системы с интегрированным управлением

На рисунке 2 представлена архитектура системы с интегрированным управлением, этапы которого представлены выше. Представлены основные процессы в системе, а также роль промежуточного программного обеспечения на основе Robotic Operating System (ROS) при планировании траектории. Используя основанный на временной шкале подход, система интерпретирует область планирования, относящуюся к сценарию HRC, рассматривая человека как неконтролируемый элемент, а робота - как частично контро-

лируемый элемент. Поведение человека моделируется как неуправляемое с нижними и верхними границами продолжительности задачи в соответствии с информацией, полученной в результате внутреннего анализа планировщика движения. Точно так же поведение робота моделируется как частично управляемое из-за совместного присутствия человека, что может повлиять на выполнение задачи робота-манипулятора в зависимости от положения человека во время операции. Уровень реализации моделирует внутренние ограничения, которые позволяют роботу фактически выполнить поставленную задачу. Временная характеристика задач движения робота использует информацию, собранную планировщиком движения (этап 1 на рис. 1), и инкапсулирует информацию о доступных траекториях во время выполнения. Согласно заданным правилам синхронизации моделируются возможное выполнение задач человеку и роботу и связанные с ними эксплуатационные требования. Предложенный подход интеграции задач и планирования движения позволяет планировщику задач генерировать конкретный сценарий коопераций и связанной модальности взаимодействия, решая наиболее подходящий способ выполнения задач робота, чтобы найти оптимальный баланс между безопасностью человека и пропускной способностью производственного процесса.

ЗАКЛЮЧЕНИЕ

Разработан метод динамического планирования траектории коллаборативного робота, в котором интегрированы планирование движения робота и планирование задач, а их интеграция используется для максимального учета динамики рабочей среды, в которой промышленный робот находится вместе с человеком. Представленная схема интеграции процессов планирования траектории и перепланирования во время выполнения гарантирует устранение временной неопределенности.

1. Michalos, G., Kaltsoukalas, K., Aivaliotis, P., Sipsas, P., Sardelis, A., Chryssolouris, G.: Design and simulation of assembly systems with mobile robots. *CIRP Annals - Manufacturing Technology* 63(1) (2014) 181 – 184.
2. Wolfe, J., Marthi, B., Russell, S.J.: Combined task and motion planning for mobile manipulation. In: *Proceedings of the 20th International Conference on Automated Planning and Scheduling, ICAPS 2010, Toronto, Ontario, Canada, May 12-16, 2010.* (2010) 254–258.
3. Pellegri-nelli, S., Moro, F.L., Pedrocchi, N., Tosatti, L.M., Tolio, T.: A probabilistic approach to workspace sharing for human-robot cooperation in assembly tasks. *CIRP Annals - Manufacturing Technology* 65(1) (2016) 57 – 60.

Снисаренко Светлана Валерьевна, аспирант кафедры систем управления, kafsu@bsuir.by
Научный руководитель: Кузнецов Александр Петрович, доктор технических наук, профессор, kafsu@bsuir.by

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ГРУППОЙ ЛИФТОВ

В работе рассматриваются особенности имитационной модели системы управления группой лифтов и особенностей применения процессно-ориентированной библиотеки дискретного моделирования *Dessert*.

ВВЕДЕНИЕ

Системы управления группой лифтов имеют большую значимость в вертикальной транспортировке из-за потребности в согласовании работы лифтов в группе по вызовам для повышения производительности лифтов, уменьшения времени ожидания кабины пассажирами, сокращения количества холостых пробогов и расхода энергии. Из-за невозможности проведения экспериментов с реальной группой лифтов была разработана имитационная модель.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ С ПРИМЕНЕНИЕМ DESSERT

В результате анализа методов имитационного моделирования разработана библиотека классов C#, моделирующая группу лифтов. В основе модели лежит библиотека для дискретного моделирования *Dessert*.

Dessert предоставляет следующие базовые типы для описания модели [1]:

- ресурсов (классы *Resource* и *Preemptive Resource*);
- контейнеры (*Container*), которые моделируют производство и потребление однородной недифференцированной массы;
- хранилища (*Store<T>* и *FilterStore<T>*), которые позволяют взаимодействовать с дискретными объектами;
- модули сбора статистики (*Tally*) о ресурсах и процессах.

Поведение активных сущностей и других процессов в системе (алгоритм работы лифта, прибытие пассажиров на этаж) может быть представлено функцией-генератором в бесконечном цикле. Все процессы взаимодействуют с имитационной средой *SimEnvironment* и друг с другом через события.

Особенностью лифтовой группы является то, что лифт, являясь ресурсом, не стационарен и имеет собственную логику поведения (движения) и не может быть представлен объектами

Струц Дмитрий Алексеевич, магистрант 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, dmitry.a.struts@gmail.com.

Научный руководитель: Марков Александр Владимирович, доцент кафедры систем управления Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, markov@bsuir.by.

Resource. В спроектированной модели лифт рассматривается как активная сущность со сложным поведением. Для описания поведения лифта был использован конечный автомат [2].

Структура модели представлена диаграммой классов на рисунке 1.

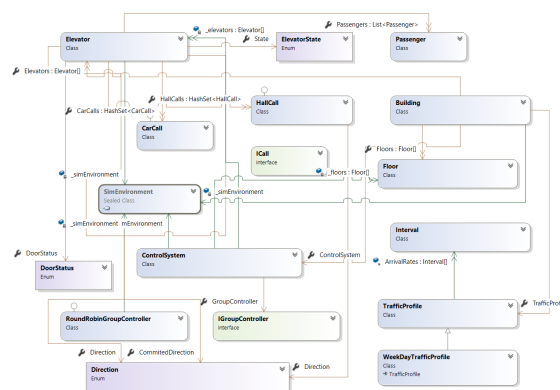


Рис. 1 – Диаграмма классов имитационной модели

ВЫВОДЫ

Были проанализированы возможности и особенности моделирования системы управления группой лифтов с помощью библиотеки *Dessert*. В результате был разработан программный прототип модели и представлена ее структура.

1. Dessert, an Open-Source .NET Framework for Process-Based Discrete-Event Simulation [Электронный ресурс] – Режим доступа: https://www.thinkmind.org/download.php?articleid=computation_tools_2014_1_10_80003
2. Наумов Л.А., Шалыто А.А. Искусство программирования лифта. Объектно-ориентированное программирование с явным выделением состояний // Информационно-управляющие системы. 2003. №6. [Электронный ресурс] – Режим доступа <https://cyberleninka.ru/article/n/iskusstvo-programirovaniya-lifta-obektno-orientirovannoe-programirovanie-s-yavnym-vydeleniem-sostoyaniy>

СИСТЕМЫ ВИДЕОНАБЛЮДЕНИЯ С ПРИМЕНЕНИЕМ ТЕПЛОВИЗИОННЫХ ТЕХНОЛОГИЙ

Лесные пожары являются мощным природным и антропогенным фактором, существенно изменяющим функционирование и состояние лесов. Лесные пожары наносят урон экологии, экономике, а зачастую под угрозой оказываются и человеческие жизни. Для стран, где леса занимают большую территорию, лесные пожары являются национальной проблемой, а ущерб, наносимый реальному сектору экономики, исчисляется десятками и сотнями миллионов долларов в год.

ВВЕДЕНИЕ

В комплексе мероприятий по предотвращению лесных пожаров наиболее актуальной задачей являются раннее обнаружение и определение местоположения очага возгорания. Для этого традиционно используются наземное наблюдение, авиация и космический мониторинг.

Данные методы обнаружения лесных пожаров имеют свои преимущества, однако в каждом из них присутствуют недостатки, способные существенно ограничить способы их применения. Именно по этой причине до сих пор применяется комплексное их использование. Естественно экономические затраты на контроль территорий данными методами весьма высоки. В связи с этим и необходима система, способная осуществлять постоянный контроль территорий с наименьшими затратами.

I. СТРУКТУРА СИСТЕМЫ

Система состоит из сети тепловизорных автоматических модулей обнаружения, устанавливаемых на вышках в локальных пунктах наблюдения и пункта управления. Функционально система состоит из следующих основных узлов: ИК-ТВ автоматических модулей обнаружения; радиосети приема-передачи цифровой информации; центрального пункта приема-передачи и обработки информации. Узлы сконструированы с применением микропроцессорной техники и работают под управлением соответствующих программных модулей в автоматическом режиме. Структурные схемы контролируемого пункта и пункта управления приведены на рисунках 1 и 2. ПУ осуществляет обработку и отображение на электронной карте местности данных в удобном для оператора виде. На ПУ также принимается решение о выдаче аварийной сигнализации или о необходимости получения дополнительной информации в виде телевизионной картинки с интересующего направления. Нахождение и пространственное выделение тепловых источников на территории осуществляется оборудованием контролируемого пункта автоматически, а получение телевизионной картинки по команде с ПУ.

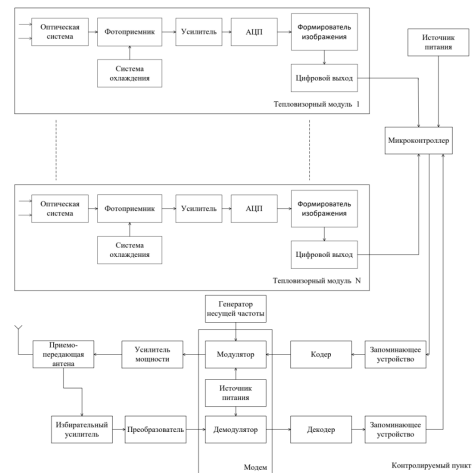


Рис. 1 – Структурная схема контролируемого пункта системы.



Рис. 2 – Структурная схема пункта управления системы.

II. СТРУКТУРА СИГНАЛОВ

Тепловизионная система работает по алгоритму адаптивной дискретизации с применением кодирования кодом Бергера и M- последовательностью 7-ой степени. Со стороны ПУ на КП в системе передается сигнал. В его состав входят: – синхрокод, предназначенный для синхронизации оборудования ПУ и КП; – адрес опрашиваемого контролируемого пункта. Система, работающая по алгоритму адаптивной дискретизации, определяет несуществующие координаты, т.е. определяются и передаются показания лишь тех тепловизионных модулей, которые изменили свое значение на заранее установленную величину. В соответствии с этим, по данному алгоритму система позволяет передавать данные в двух режимах: – адресный; – безадресный. Второй используется в случаях, когда общая длина посылки меньше,

чем длина посылки, состоящей из адресов активных ТВМ, чьи показания необходимо передать и адресов этих ТВМ. Поэтому со стороны КП на ПУ будет использоваться два вида посылок в зависимости от режима работы. В состав сигнала входят: – адрес отвечающего контролируемого пункта; – код режима работы; – данные от всех тепловизорных модулей на КП; – код конца, может отсутствовать, так как на ПУ известно количество тепловизорных модулей на КП. Так как режима может быть два, достаточно одного бита для кодирования режима работы.

III. ТЕПЛОВИЗОРНЫЙ АВТОМАТИЧЕСКИЙ МОДУЛЬ ОБНАРУЖЕНИЯ

Тепловизорный-модуль - основной узел системы. Он выполняет функции сканирования территорий и поиска, идентификации и выделения сигналов очагов возгорания в ИК диапазоне по их собственному тепловому излучению; функции получения цифровой телевизионной картинки; функции получения, обработки и идентификации информации от дополнительных датчиков параметров среды: метеорологических, радиационных и т.д. По общему принципу работы тепловизоров, функциональная схема которого изображена на рисунке 3, инфракрасное излучение концентрируется системой специальных линз и попадает на фотоприёмник, который избирательно чувствителен к определённой длине волны инфракрасного спектра. Попадающее на него излучение приводит к изменению электрических свойств фотоприёмника, что регистрируется и усиливается электронной схемой. Полученный сигнал подвергается цифровой обработке, и это значение передаётся на блок отображения информации. Блок отображения информации имеет цветовую палитру, в которой каждому значению сигнала присваивается определённый цвет. После этого на мониторе появляется точка, цвет которой соответствует численному значению инфракрасного излучения, которое попало на фотоприёмник. Сканирующая система (зеркала или полупроводниковая матрица) проводит последовательный обход всех точек в пределах поля зрения прибора, и в результате получается видимая картина инфракрасного излучения объекта. Таким образом, на мониторе тепловизора мы видим значения мощности инфракрасного излучения в каждой точке поля зрения тепловизора, отображённые согласно заданной цветовой палитре (черно- белой или цветной). Тепловизорная телескопическая система выполняет функ-

ции спектральной и пространственной селекции очагов тепла в ИК-диапазоне и выделения сигналов помех в видимом диапазоне. Она выполнена в виде объектива Кассегрена с фокусным расстоянием 500 мм, светосилой 0,4 и диаметром основного зеркала 200 мм. Угловое поле зрения ИК-канала – 7 минут. Спектральный диапазон – 2.8 – 5.2 мкм. В состав телескопической системы входит объектив и датчик видимого диапазона с угловым полем зрения – 1 градус. По амплитуде сигнала с датчика видимого диапазона программным обеспечением микропроцессорного модуля принимается решение об отсечке полезного сигнала ИК-канала от ложного. Это повышает помехозащищённость системы и снижает вероятность ложных срабатываний от солнечных бликов.



Рис. 3 – Обобщенная структурная схема тепловизора

ЗАКЛЮЧЕНИЕ

Таким образом, была разработана автоматизированная система видеонаблюдения с применением тепловизионных технологий для автоматического дистанционного обнаружения координат тепловых источников по собственному излучению в ИК-диапазоне спектра на неоднородно меняющемся фоне при различных метеорологических ситуациях, визуализация ситуации по направлению обнаружения цифровым телевизионным каналом.

1. Antsipov G. V. AN AUTOMATED REMOTE INFRARED AND TELEVISION SYSTEM OF FOREST FIRE AND ECOLOGICAL MONITORING. International forest fire news No23 (December, 2000), p.p. 92-96.
2. Шамаль В. А. Автоматизированная круглосуточная инфракрасно- телевизионная дистанционная система обнаружения лесных пожаров. Лесное хозяйство. / Атрощенко О. А., Мыслейко И. Г., Новик А. Н., Белый И. В., Бельский А. В. – Минск : БГТУ 2000, с.74-81.
3. Инфракрасная термография в энергетике Афонин А. В. [и др.] Под ред. Ньюпорта Р. К., Таджибаева А. И.Т. 1. Основы инфракрасной термографии. – Санкт-Петербург: СПЭ-ИПК, 2000. 240 с.
4. Л. Шапиро, Дж. Стокман. Компьютерное зрение = Computer Vision. – М.: Бином. Лаборатория знаний, 2006. – 752 с. – ISBN 5-94774- 384-1.

Тарасюк Евгений Васильевич, аспирант кафедры систем управления, tarasiuk-ev@bsuir.by

Научный руководитель: Шилин Леонид Юрьевич, доктор технических наук, профессор, dekfitu@bsuir.by

ПРОТОКОЛ DHCP ДЛЯ АВТОМАТИЧЕСКОЙ НАСТРОЙКИ СЕТЕВЫХ УСТРОЙСТВ

Рассмотрены достоинства создания протокола DHCP, принцип работы протокола на DHCP-сервере и последовательность развёртки службы DHCP на DHCP-сервере.

ВВЕДЕНИЕ

Протокол DHCP семейства протоколов TCP/IP, свертывает ошибки конфигурации, вызванные ручной конфигурацией IP-адресов, например конфликты адресов, вызванные назначением IP-адреса более чем одному компьютеру одновременно. Использование DHCP-сервера и настройка службы DHCP является основной задачей при настройке большого количества рабочих мест на предприятиях.

I. ПРИНЦИП РАБОТЫ ПРОТОКОЛА DHCP. ОБНАРУЖЕНИЕ И ПРЕДЛОЖЕНИЕ

DHCP работает по схеме клиент-сервер. Процесс получения настроек происходит в несколько этапов и описывается схемой DORA (Discover-Offer-Request-Acknowledge):

Discover (Обнаружение). Клиент DHCP подключается к сети и приступает к инициализации (состояние INIT). Клиент ищет подходящий DHCP-сервер, для чего отправляет запрос DHCPDISCOVER на широковещательный адрес 255.255.255.255. В запросе указывает адрес 0.0.0.0, поскольку своего адреса у него еще нет и свой MAC-адрес. Запрос доставляется всем компьютерам, находящимся в данном сегменте сети, но отвечают на него только DHCP-сервера.



Рис. 1 – Схема получения сетевых настроек клиента

Offer (Предложение). DHCP-сервер, получивший запрос DHCPDISCOVER, анализирует его содержимое, выбирает подходящую конфигурацию сети и отправляют ее в сообщении DHCPOFFER. Если в сети находятся несколько

DHCP-серверов, то клиент получает несколько ответов DHCPOFFER и выбирает из них один, как правило полученный первым.

II. ПРИНЦИП РАБОТЫ ПРОТОКОЛА DHCP. ЗАПРОС И ПОДТВЕРЖДЕНИЕ

Request (Запрос). Получив ответ сервера, клиент отвечает сообщением DHCPREQUEST, в котором запрашивает у сервера предоставленные настройки. В сообщении DHCPREQUEST содержится та же информация, что и в DHCPDISCOVER, а также IP-адрес выбранного DHCP-сервера. DHCPREQUEST отправляется на широковещательный адрес и те DHCP-сервера, чей адрес отсутствует в сообщении, понимают что их предложение отвергнуто.

Acknowledge (Подтверждение). DHCP-сервер, адрес которого указан в DHCPREQUEST, получает сообщение. Он фиксирует привязку для клиента и отвечает сообщением DHCPACK, подтверждая выданные клиенту настройки. DHCPACK отправляется на MAC-адрес клиента, указанный в DHCPREQUEST. Клиент получает сообщение DHCPACK, проверяет настройки и применяет конфигурацию (состояние BOUND), которая была получена в сообщении DHCPOFFER.

III. ВЫВОД

Сервис DHCP является довольно гибким инструментом для динамической конфигурации хостов. Протокол DHCP позволяет в разы ускорить конфигурирование рабочих мест на предприятиях и обезопасить сеть от конфликта IP-адресов.

1. DHCP: искусство управления IP-адресами, интернет ресурс: <http://citforum.ru/nets/tcp/dhcp.shtml>
2. Протокол динамического конфигурирования ЭВМ DHCP, интернет ресурс: <http://book.itep.ru/4/4/dhcp.htm>
3. Каретко Н.Н. Основы системного администрирования. Минск, 2020. - 85 с.

Трофимов Дмитрий Сергеевич, магистрант 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, dmitrii.trofimov1387@gmail.com

Научный руководитель: Павлова Анна Валентиновна, доцент кафедры систем управления Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, pavlova@bsuir.by

Секция "Автоматизированные системы обработки информации"

Председатель: канд. физ.-мат. наук, доцент, зав. каф. ИТАС Навроцкий А.А.
Члены жюри: канд. тех. наук, Ломако А.В.
д-р тех. наук, проф. Муха В.С.
канд. тех. наук, доцент Герман О.В.
Секретарь ст. преп. Боброва Т.С.

МЕТОДЫ СТАТИЧЕСКОГО АНАЛИЗА ПРОГРАММНОГО КОДА

Рассматривается возможность оптимизации и повышения защищенности приложений посредством статического анализа программного кода.

ВВЕДЕНИЕ

Статический анализ кода это процесс выявления ошибок и недочетов в исходном коде программ. Статический анализ можно рассматривать как автоматизированный процесс обзора кода. Существенный недостаток методологии мануального обзора кода, это крайне высокая цена, поэтому компромиссным решением являются инструменты статического анализа кода, которые постоянно обрабатывают исходные тексты программ и выдают программисту рекомендации обратить повышенное внимание на определенные участки кода.

1. ОСНОВНЫЕ ЗАДАЧИ И ПРИЕМУЩЕСТВА

В целом задачи, решаемые с помощью методов статического анализа кода можно разделить на три категории:

1. Выявление ошибок в программном коде;
2. Рекомендации по оформлению программного кода. Некоторые платформы статического анализа позволяют проверять программный код на соответствие заданному стандарту оформления программного кода;
3. Подсчет метрик, позволяющих получить численное значение того или иного свойства программного обеспечения.

Главное преимущество статического анализ программного кода состоит в возможности существенного снижения стоимости устранения дефектов в программном обеспечении. Чем раньше ошибка выявлена, тем меньше стоимость ее исправления.

Применение методов статического анализа программного кода позволяет выявить большое число ошибок на этапе конструирования, или написания программного кода, что существенно снижает стоимость разработки всего проекта.

Статический анализ программного кода обладает рядом других преимуществ. Одним из них является полное покрытие кода, что позволяет находить дефекты и уязвимости в обработчиках редких ситуации, мануальное тестирование которых затруднено по тем или иным причинам. Статический анализ является независимым от среды использования и комплятора, что в свою очередь позволяет обнаруживать ошибки,

которые могут проявиться в процессе длительного использования ПО.

II. ОБЗОР МЕТОДОВ НА ПРИМЕРЕ ИНТЕРВАЛЬНОГО АНАЛИЗА

Интервальный анализ вычисляет для каждой целочисленной переменной верхнюю и нижнюю границы для ее возможных значений. Интервалы являются интересными результатами анализа, поскольку эти результаты могут использоваться для оптимизации и обнаружения ошибок, связанных с проверкой границ массива, числовых переполнений и представления целых чисел. Данный случай включает решётку бесконечной высоты, и мы должны использовать специальный метод для обеспечения сходимости к фиксированной точке [1].

Решетка, описывающая одно абстрактное значение, определяется следующим образом:

$$Intervals = lift(\{[l, h] \mid l, h \in N \wedge l \leq h\}), \quad (1)$$

где

$$N = \{-\infty, \dots, -2, -1, 0, 1, 2, \dots, +\infty\} \quad (2)$$

это множество целых чисел с бесконечным количеством конечных точек и порядком интервалов, определенных путем включения:

$$[l_1, h_1] \sqsubseteq [l_2, h_2] \Leftrightarrow l_2 \leq l_1 \wedge h_1 \leq h_2 \quad (3)$$

Эта решётка не имеет конечной высоты, поскольку она содержит, к примеру, следующую бесконечную цепь:

$$[0, 0] \sqsubseteq [0, 1] \sqsubseteq [0, 2] \sqsubseteq [0, 3] \sqsubseteq [0, 4] \sqsubseteq [0, 5] \dots \quad (4)$$

Это переносится на решетку абстрактных состояний

$$States = Vars \rightarrow Intervals \quad (5)$$

До указания правил ограничения необходимо определить функцию eval, которая выполняет абстрактную оценку выражений:

$$eval(\sigma, X) = \sigma(X) \quad (6)$$

$$eval(\sigma, I) = [I, I] \quad (7)$$

$$eval(\sigma, input) = [-\infty, +\infty] \quad (8)$$

$$eval(\sigma, E_1 \text{ op } E_2) = \widehat{op}(eval(\sigma, E_1), eval(\sigma, E_2)) \quad (9)$$

Абстрактные арифметические операторы определяются как:

$$\hat{op}([l_1, h_1], [l_2, h_2]) = \left[\begin{array}{l} \min_{x \in [l_1, h_1], y \in [l_2, h_2]} x \\ \text{op } y \quad \max_{x \in [l_1, h_1], y \in [l_2, h_2]} x \text{ op } y \end{array} \right] \quad (10)$$

Например:

$$\hat{+}([1, 10], [-5, 7]) = [1 - 5, 10 + 7] = [-4, 17] \quad (11)$$

Функция JOIN является обычной для форвардного анализа:

$$JOIN(v) = \prod_{w \in pred(v)} \llbracket w \rrbracket \quad (12)$$

Теперь мы можем указать правило ограничения для присвоений:

$$\begin{aligned} X = E : \llbracket v \rrbracket &= JOIN(v)[X \\ &\mapsto eval(JOIN(v), E) \end{aligned} \quad (13)$$

Для остальных узлов ограничение является тривиальным:

$$\llbracket v \rrbracket = JOIN(v) \quad (14)$$

Интервальный анализ решётки имеет бесконечную высоту, так что применение алгоритмов

фиксации точки может никогда не заканчиваться: для решётки Lp последовательность приближений

$$f^i(\perp, \dots, \perp) \quad (15)$$

никогда не должна сходиться. Мощным инструментом для обхода подобного рода задачи является применение методов расширения и сужения.

III. Выводы

Существует множество методов статического анализа программного кода, однако ни один из них не является ультимативным, поэтому выбор того или иного метода статического анализа программного кода зависит от большого числа факторов таких, как среда разработки и применения, архитектурные особенности продукта и т.д.. Метод интервального анализа используется для решения проблемы неопределенного потока мощности в централизованном кластере, так что в интервал включаются все решения, что в свою очередь позволяет покрыть все возможные варианты использования.

1. Андрес Меллер и Махаэль И. Шварцбах, Статический анализ программ / А. Меллер, М. И. Шварцбах // 2019. – С. 75-77.

Азаренко Алексей Васильевич, магистрант кафедры информационных технологий и управления БГУИР, lehaazarenko@mail.ru.

Научный руководитель: Навроцкий Анатолий Александрович, заведующий кафедрой информационных технологий автоматизированных систем БГУИР, кандидат физико-математических наук, доцент, navrotsky@bsuir.by.

СРАВНЕНИЕ АЛГОРИТМОВ ВЕКТОРИЗАЦИЯ ТЕКСТА В СИСТЕМАХ КЛАССИФИКАЦИИ

Рассматривается задача векторизации текстов в системах классификации. Сравнивается подход к классификации текста на основе подхода Word2Vec и Bag of Words.

ВВЕДЕНИЕ

Классификация текстов является одной из основных задач компьютерной лингвистики, так как к ней сводятся некоторые другие задачи: определение темы текстов, автора текста, эмоциональной окраски и др. Первым этапом решения задачи классификации текстов является представление текста в виде вектора. От качества решения данной задачи зависит время и качество решения задачи классификации целиком. Среди методов векторизации текстов, разработанных на данный момент, самыми распространёнными являются Word2Vec и Bag of Words. В связи с важностью качественного представления текста в виде вектора, актуальным является вопрос выбора лучшего из них.

I. ПОСТАНОВКА ЗАДАЧИ

Формально постановку задачи векторизации можно записать следующим образом. Имеются текстовый документ состоящий из множества слов $D = d_1, \dots, d_n$. Необходимо построить функцию, которая преобразует множество слов D в числовой вектор $X = x_1, \dots, x_n$.

II. КРИТЕРИИ СРАВНЕНИЯ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ

Для оценки качества векторизации текстовых документов в системах классификации можно оценить качество работы алгоритмов классификации при использовании исследуемых алгоритмов векторизации. Для выполнения сравнения использовались метод опорных векторов и классификатор Байеса. Основным критерием при оценке качества классификации является комбинация точности и полноты. Пусть TP – это истинно положительное решение; TN – это истинно отрицательное решение; FP – ложно положительное решение; FN – ложно отрицательное решение. Тогда точность вычисляется по формуле:

$$p = \frac{TP}{TP + FP}$$

Полнота вычисляется следующим образом:

$$r = \frac{TP}{TP + FN}$$

Азарко Владислав Вячеславович, Аспирант кафедры информационных технологий автоматизированных систем БГУИР, azarkovlad@gmail.com.

Научный руководитель: Гуринович Алеватина Борисовна, заместитель декана ФИТУ, кандидат технических наук, доцент, gurinovich@bsuir.by

III. ЭКСПЕРИМЕНТЫ ПО СРАВНЕНИЮ МЕТОДОВ

В статье [1] описаны результаты многих экспериментов по сравнению вышеописанных классификаторов с использованием подходов Word2Vec и Bag of Words. Точность и полнота работы классификаторов с использованием подхода Bag of Words описана в таблице 1, с использованием Word2Vec – в таблице 2.

Таблица 1 – Результаты эксперимента с использованием алгоритма Bag of Words

Метод	Точность	Полнота
Метод опорных векторов	80-85%	83-87%
Классификатор Байеса	80-95%	70-85%

Таблица 2 – Результаты эксперимента с использованием алгоритма Word2Vec

Метод	Точность	Полнота
Метод опорных векторов	85-89%	89-92%
Классификатор Байеса	80-85%	74-80%

IV. ВЫВОДЫ

В соответствии с результатами различных исследований, наилучшими методами для классификации текста по критериям точность и полнота являются свёрточные нейронные сети и метод опорных векторов. Использование алгоритмов векторизации значительно влияет на качество работы классификаторов, однако для каждого конкретного классификатора целесообразно выбирать подходящий алгоритм классификации.

1. A Text classification problem and features set / Polyakov I.V. // Vestn. NGU 2015,

МЕТОДЫ ДЛЯ РАСПОЗНАВАНИЯ ЖЕСТОВ

В данной работе рассматривается задача распознавания жестов на цифровых изображениях и видеопоследовательности в режиме реального времени.

ВВЕДЕНИЕ

В настоящее время успешно развивается направление, связанное с интеллектуализацией методов обработки и анализа данных. Задача распознавания лиц и жестов является одной из первых практических задач, которая послужила стимулом для развития теории распознавания объектов. Распознавание лиц и жестов находит применение в различных сферах человеческой деятельности.

I. АЛГОРИТМ РАСПОЗНАВАНИЯ ЖЕСТОВ НА ЦИФРОВЫХ ИЗОБРАЖЕНИЯХ

Целью данной работы является распознавание жестов на цифровых изображениях. Для решения этой задачи предложен новый алгоритм, основанный на применении вейвлет-преобразования и метода главных компонент. Предложенный алгоритм состоит из двух процессов: извлечения и сохранения признаков известных жестов в базе данных и распознавания жестов. Процесс извлечения и сохранения признаков известных жестов происходит следующим образом:

Шаг 1. Преобразование изображения области жеста в полутоновое изображение.

Шаг 2. Изменение размера области жеста до 64x64 пикселей.

Шаг 3. Применение к полученному на шаге 2 изображению вейвлет-преобразования для извлечения признаков жеста (вейвлет-коэффициентов).

Шаг 4. Сохранение извлеченных признаков в базе данных. В процессе распознавания неизвестного жеста осуществляются шаги 1–3, затем полученные признаки сравниваются с признаками, хранящимися в базе данных, на основе применения метода главных компонент. Функциональная схема предложенного алгоритма представлена на рис 1.

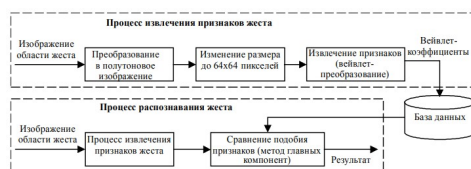


Рис. 1 – Функциональная схема алгоритма распознавания жестов

Арутюнова Таисия Арсеновна, магистрантка кафедры информационных технологий автоматизированных систем БГУИР, taya.art.fun@gmail.com.

Научный руководитель: Муха Владимир Степанович, доктор технических наук, профессор, mukha@bsuir.by.

II. АЛГОРИТМ РАСПОЗНАВАНИЯ ЖЕСТОВ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ

В данной работе также рассматривается задача распознавания жестов на видеопоследовательности в режиме реального времени. Для решения этой задачи предложен оригинальный комплексный алгоритм, основанный на применении метода Виолы – Джонса, алгоритма SAMShift, вейвлет-преобразования и метода главных компонент. Процесс распознавания жестов на видеопоследовательности происходит следующим образом:

Шаг 1. Запрос очередного видеофрайма. Преобразование видеофрайма в полутоновое изображение. Применение к полутоновому изображению метода Виолы – Джонса для поиска области руки.

Шаг 2. Если область руки обнаружена, то выполняется шаг 3. В обратном случае осуществляется возврат на шаг 1.

Шаг 3. Запрос очередного видеофрайма. Отслеживание области руки на основе применения алгоритма SAMShift.

Шаг 4. Если отслеживание осуществлено, то выполняется шаг 5. В обратном случае происходит возврат на шаг 1.

Шаг 5. Выполнение процесса распознавания жеста (рис. 1).

Шаг 6. Возврат на шаг 3.

III. ВЫВОДЫ

Были представлены алгоритм распознавания жестов на цифровых изображениях и алгоритм распознавания жестов на видеопоследовательности, которые решают задачу распознавания жестов.

1. Ярмолик, В. Н. Физически неклонированные функции / В. Н. Ярмолик, Ю. Г. Вашинко // Информатика. – 2011. – №2. – С. 20-30.

ОБЗОР И АНАЛИЗ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УЧЕТА МАТЕРИАЛЬНЫХ ЦЕННОСТЕЙ УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ

Система управления материальными ценностями – информационная система, обеспечивающая автоматизацию управления процессами складской работы учреждения.

I. ВВЕДЕНИЕ

В последнее время в сфере образования наблюдается тенденция использования информационных технологий во всех процессах управления учебных заведений. Внедрение последних разработок и технологий в области управления позволяет в значительной степени увеличить производительность труда сотрудников.[1] В общем случае, систему управления можно рассматривать в виде совокупности взаимосвязанных управленческих процессов и объектов. Обобщенной целью автоматизации управления является повышение эффективности использования потенциальных возможностей объекта управления.

II. ПОСТАНОВКА ЗАДАЧИ

Рассматривая специфику ведения учета, составление заявок в университете, можно сделать вывод, что бумажный документооборот влечет за собой ряд проблем, связанных с возможностью утери заявок, отказом в одобрении заявления в связи с ошибками оформления, невозможность редактировать заявления и долгая обратная связь с сотрудником, который оформлял данный документ. Однако система, которая бы могла в режиме реального времени давать доступ к базе данных этих заявлений, где можно отслеживать дату создания заявления и редактирования, то она бы позволила решить данные проблемы.[2]

III. ОБЗОР И АНАЛИЗ СИСТЕМ

Автоматизированная система учета материальных ценностей ВУЗа должна решать следующие задачи: – повышение эффективности и до-

стоверности учета движения материальных ценностей; – уменьшение вероятности ошибок, вызванных «человеческим фактором»; – сокращение трудозатрат и временных издержек по учету движения материальных средств; – уменьшение вероятности воровства и несанкционированного использования материальных средств; – оптимизация процесса по ремонту и техническому обслуживанию материальных средств; – улучшение контроля за составом, местонахождением и перемещением материальных ценностей.

IV. ВЫВОДЫ

Рассматривая специфику ведения учета, составление заявок в университете, можно сделать вывод, что бумажный документооборот влечет за собой ряд проблем, связанных с возможностью утери заявок, отказом в одобрении заявления в связи с ошибками оформления, невозможность редактировать заявления и долгая обратная связь с сотрудником, который оформлял данный документ. [3] Однако система, которая бы могла в режиме реального времени давать доступ к базе данных этих заявлений, где можно отслеживать дату создания заявления и редактирования, то она бы позволила решить данные проблемы.

СПИСОК ЛИТЕРАТУРЫ И ССЫЛКИ НА ЦИТИРУЕМЫЕ ИСТОЧНИКИ

Список литературы

1. Розина, Т. М. Распределительная логистика / Т. М. Розина. – Мн. : Выш. Шк., 2012. – 319 с.
2. Состав университета [Электронный ресурс]. – Режим доступа : <https://www.bsuir.by/>
3. Бесекерский В.А., Попов Е.П. «Теория систем автоматического управления. – 4-е изд., перераб. и доп. – СПб.: Профессия, 2003. – 747 с.

Базыльчик Роман Игоревич, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, romanikum@gmail.com.

Научный руководитель: Ломако Александр Викторович, доцент информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, lavlot@bsuir.by.

МЕТОДЫ РАСПОЗНАВАНИЯ ТИПОВ СТАТЕЙ

Рассматривается задача распознавания типов статей. Описываются методы, используемые при распознавании. Проводится сравнительный анализ таких методов.

ВВЕДЕНИЕ

Прогресс в области информационных технологий обусловил широкое распространение обработки в реальном времени больших потоков текстовых данных. В современных информационных технологиях роль такой процедуры, как извлечение информации, всё больше возрастает — из-за стремительного увеличения количества неструктурированной информации. Поэтому актуальна проблема создания моделей и алгоритмов, позволяющих эффективно обрабатывать большие потоки данных, особенно в условиях ограниченных временных и других ресурсов. В основе распознавания типов статей лежит классификация текста. В данной статье рассмотрим основные принципы работы двух методов классификации текста.

I. ПЛОСКАЯ КЛАССИФИКАЦИЯ

Каждый тип статьи, который необходимо установить, назовем измерением. Имея в первом измерении N групп, а во втором M групп, определим $(N * M)$ новых классов, которые представляют собой всевозможные сочетания этих групп. Далее в соответствии с выбранным алгоритмом классификации выполним классификацию текстов. Иллюстрация данного алгоритма для $N = 2$ и $M = 3$ представлена на рисунке 1.

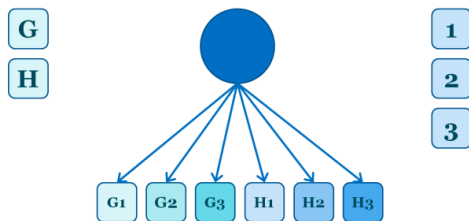


Рис. 1 – Плоская классификация

Этот метод обладает следующими особенностями. Во-первых, при построении модели на каждый класс приходится относительно небольшой объем входных данных. Во-вторых, при больших числах N и M точность классификации может оказаться низкой, так как между некоторыми классами будут существовать лишь незначительные различия. Это затруднит определение

класса, к которому относится рассматриваемый объект. Данный метод, на самом деле, является лишь модификацией классической классификации, когда каждому объекту сопоставляется вектор из нескольких значений, обозначающих принадлежность объекта соответствующему классу.

II. ИЕРАРХИЧЕСКАЯ КЛАССИФИКАЦИЯ

Данный подход заключается в том, что сначала классификация производится по одному признаку, а затем для получившихся классов независимо друг от друга выполняется классификация по второму признаку. Каждому объекту присваивается несколько классов, в соответствии с количеством извлекаемых признаков. При этом в случае двух измерений возможны два варианта алгоритма, в зависимости от того, по какому признаку классификация производится в первую очередь. Рисунок 2 иллюстрирует данный подход.

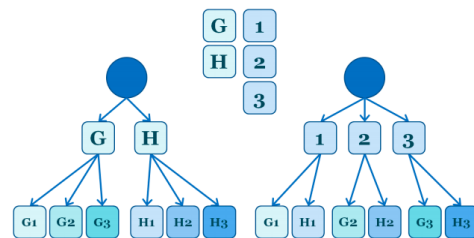


Рис. 2 – Иерархическая классификация

Особенность этого метода заключается в том, что на первом уровне на каждый класс приходится больший объем данных, чем на втором, что может сказаться на точности алгоритма.

III. ВЫВОДЫ

Рассмотренные методы классификации текста позволяют работать со статьями, для определения их типа. Иерархическая классификация обычно сводится к задаче плоской классификации, также у иерархической структуры классов есть возможность налету определять ошибки.

1. Hierarchical Classification of Web Content. / Dumais, S., Chen, H. // Athens, Greece, 2000.

Бартош Владислав Иванович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, vlad.bartosh15@gmail.com.

Научный руководитель: Севернёв Александр Михайлович, доцент кафедры информационных технологий автоматизированных систем, кандидат технических наук, доцент, severnev@bsuir.by.

ПРИМЕНЕНИЕ АКСЕЛЕРОМЕТРИЧЕСКОГО МЕТОДА ДЛЯ РАННЕЙ ДИАГНОСТИКИ ЗАБОЛЕВАНИЙ ЦЕНТРАЛЬНОЙ НЕРВНОЙ СИСТЕМЫ

Применение акселерометрического метода для ранней диагностики заболеваний центральной нервной системы, сопровождающихся тремором.

ВВЕДЕНИЕ

В последние годы с увеличением численности пожилого населения в развитых странах отмечается неуклонный рост нейродегенеративных заболеваний (например, болезнь Паркинсона), а также наблюдается тенденция к преждевременному старению мозга у представителей более молодого возраста в связи с постоянными переутомлением и стрессами, неправильным питанием и вредными привычками.

Диагностика заболеваний центральной нервной системы (ЦНС) на ранних стадиях важна в связи с возможностью начала своевременного лечения, приостановки развития заболевания и дегенеративных процессов, происходящих в центральной нервной системе.

1. АКСЕЛЕРОМЕТРИЧЕСКИЙ МЕТОД ДИАГНОСТИКИ

При дрожательно-ригидной и дрожательной формах болезни Паркинсона одним из характерных проявлений заболевания является тремор. В настоящее время основным способом оценки тремора при болезни Паркинсона и его динамики является клинический метод, основанный на описании клинической картины и оценки степени выраженности тремора в баллах по шкале UPDRS [1].

Для инструментальной диагностики тремора чаще всего используются следующие методы:

- электромиография (регистрируются электрические потенциалы, генерируемые мышечными волокнами в процессе сокращения);
- акселерометрия (измеряет ускорение движения конечностей);
- гироскопия (измеряются угловые скорости движения конечностей);
- тензометрия (регистрируется непосредственный тактильный контакт с тензометрическим датчиком);
- видеорегистрация и другие.

Акселерометрический метод относится к кинематическим методам, основной чертой которых является непосредственная регистрация колебательных движений с помощью миниатюрных сенсоров (датчиков) изготовленных по МЭМС-технологии. Суть метода заключается в измерении ускорения (проекции ускорения)

вдоль осей чувствительности X, Y и Z датчика. Акселерометрические датчики по типу конструкции подразделяют на: пьезоэлектрические, пьезорезистивные (силиконовые, пленочные) и емкостные.

Конструкция емкостного МЭМС акселерометра основана на измерительной ячейке, внутри которой размещена консоль с подвешенной инертной массой. На внутренние поверхности корпуса и поверхности массы нанесены электроды, что превращает конструкцию в систему из двух конденсаторов (рис. 1).

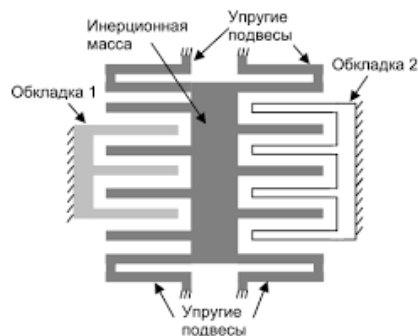


Рис. 1 – Конструкция емкостного акселерометрического датчика

Под действием ускорения инертная масса колеблется на упругих подвесах, что приводит к изменению расстояния между обкладками обоих конденсаторов и изменению их емкости. Изменение емкости конденсаторов вызывает изменение потенциалов на их обкладках, которое образует сигнал пропорциональный приложенному ускорению.

Акселерометрический метод используется для дифференциальной диагностики различных видов тремора, выявления характерных паттернов дрожательных гиперкинезов и может применяться в сочетании с электромиографией, электроэнцефаллографией и другими инструментальными методами.

Электронейрографические (ЭНГ) и электромиографические (ЭМГ) методы не обеспечивают получение полной информации о характере тремора и требуют дополнения клиническими тестами. В свою очередь акселерометрический метод позволяет исследование ведущих характеристик тремора - амплитуды, спектра частот и мощности тремора в определенной точке (рис. 2).

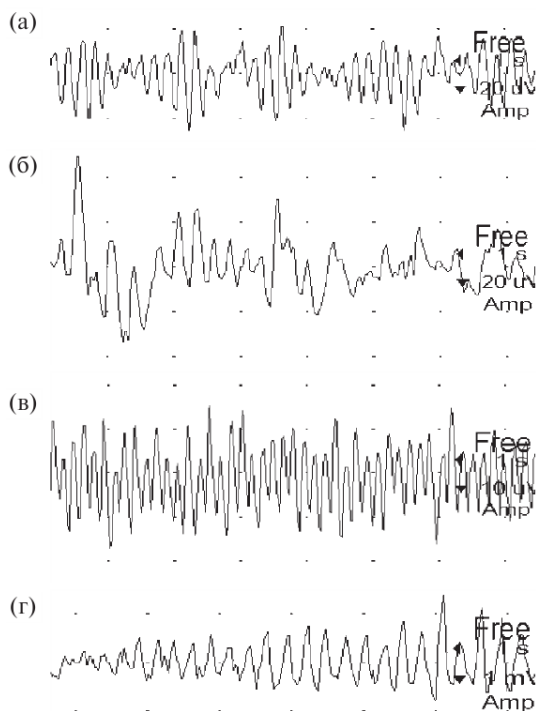


Рис. 2 – Результаты регистрация тремора акселерометрическим методом [2] (а – Паркинсональный тремор покоя; б – дистонический тремор покоя; в – эссенциальный тремор; г – постуральный тремор при гепатолентикулярной дегенерации)

II. АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ДИАГНОСТИКИ ЗАБОЛЕВАНИЙ ЦНС НА ОСНОВЕ АКСЕЛЕРОМЕТРИЧЕСКОГО МЕТОДА

На схеме (рис.3) представлен аппаратно-программный комплекс для исследования параметров тремора, включающий в себя портативное устройство, реализованное на основе микроконтроллера и 3-х осевого акселерометрического датчика.

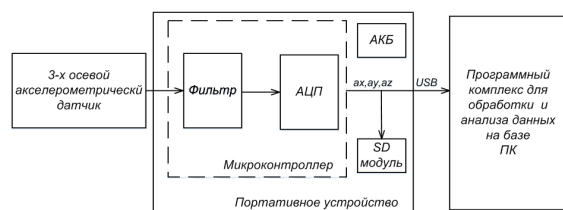


Рис. 3 – Схема аппаратно-программного комплекса для регистрации тремора

Данные с датчика (значения проекций ускорения по трем осям a_x , a_y и a_z) в виде уже уси-

Боброва Татьяна Сергеевна, аспирант кафедры теоретических основ электротехники БГУИР, t.bobrova@bsuir.by

Ярмолик Валерий Иванович, ассистент кафедры информационных технологий автоматизированных систем БГУИР, v.jarmolik@bsuir.by

Научный руководитель: Давыдов Максим Викторович, первый проректор БГУИР, кандидат технических наук, доцент, davydov-mv@bsuir.by

ленного аналогового сигнала считываются портативным устройством. Портативное устройство включает в себя фильтр низких частот, так как полезный сигнал является низкочастотным (чаще всего 3-12 Гц в зависимости от вида тремора), и важно убрать шумы и наводки, а также аналогово-цифровой преобразователь для преобразования аналогового сигнала в цифровой.

Портативное устройство имеет SD модуль для сохранения данных на карту памяти, в случае автономной работы или невозможности передавать данные для обработки в реальном времени. В качестве элемента питания используется аккумуляторная батарея.

С портативного устройства данные через USB-порт передаются на базовое устройство для дальнейшей обработки и анализа сигнала. В роли этого устройства могут выступать персональный компьютер, ноутбук, планшет или смартфон с установленным на него программным обеспечением, с помощью которого проводится частотный спектральный анализ сигнала тремора, вычисляется его амплитуда и мощность.

Наличие автономного режима работы у портативного устройства позволяет проводить мониторинг движения конечностей при ведении пациентом обычного образа жизни, что является важным для диагностики заболеваний центральной нервной системы на ранних стадиях, когда тремор слабо выражен и появляется время от времени. Такой режим позволит выявить и эссенциальный тремор, который появляется при движении конечностей.

III. Выводы

Таким образом, предлагаемое устройство реализует акселерометрический метод и может использоваться для ранней диагностики заболеваний центральной нервной системы, сопровождающихся различными видами тремора, так как позволяет проводить мониторинг движения конечностей в течение длительного времени.

Список литературы

1. Fahn, S. Unified Parkinson's Disease rating scale / S. Fahn, R. Elton, M. Goldstein // Recent Developments in Parkinson's Disease. – N-Y.: Macmillan Healthcare Information, 1987. – Vol.10. – P.153–163.
2. Иванова-Смоленская, И. А. Современные инструментальные методы регистрации тремора / И. А. Иванова-Смоленская, А. В. Карабанов, А. В. Червяков // Новые технологии. – 2011. – № 2. – С. 17–23.

ДОСТАТОЧНОСТЬ 10 ПРИНЦИПОВ OWASP ДЛЯ НАПИСАНИЯ БЕЗОПАСНОГО КОДА

OWASP Top 10 - это список, который публикуется открытым проектом обеспечения безопасности веб-приложений (OWASP). Эта статья сравнивает, насколько слабости, описанные в первой десятке списка на самом деле сообщают об уязвимостях, которые перечислены в Национальной базе уязвимостей (NVD). Таким образом, это позволяет эмпирически показать, является ли OWASP Top 10 список достаточно полным, для ориентирования в слабых местах кода, которые были найдены в последнее десятилетие.

ВВЕДЕНИЕ

Открытый проект безопасности веб-приложений (OWASP) публикует известный список уязвимостей, ранжированных по порядку от одного до десяти. Они стремятся предоставить этот список в качестве предупреждения разработчикам, чтобы избежать создания уязвимостей в написанных ими приложениях. С другой стороны, национальная база уязвимостей данных является хранилищем данных для всех уязвимостей, которые были официально опубликованы и описаны. Каждая уязвимость обычно имеет назначенный идентификационный номер уязвимости и подверженности (CVE). Идентификаторы CVE обычно имеют система классификации ошибок, приводящих к уязвимостям (CWE), которые описывают в целом типы уязвимостей, наблюдаемых в сообщенных CVE. Между CVE и CWE существуют отношения один ко многим соответственно.

I. ИССЛЕДОВАТЕЛЬСКИЕ ВОПРОСЫ

1. Каковы 10 самых частых CWE за последние десять лет в Национальной базе уязвимостей?
2. Какие идентификаторы CWE используются для уязвимостей, описанных OWASP топ 10?
3. Какой самый частый CWE за последние десять лет
4. Сколько уязвимостей с соответствующими CWE в Национальной базе уязвимостей фактически присутствует в OWASP топ 10 CWE?

II. ПОДГОТОВКА ДАННЫХ

Данные из Национальной базы данных уязвимостей (NVD) можно получить в формате нотации объектов JavaScript (JSON) [1]. Одна из проблем в отношении анализа того, как сообщается об уязвимостях является тот факт, что они

сообщаются год за годом, не в одном файле. Кроме того, они не соответствуют структуре таблицы для легкого анализа, так как JSON объектно-ориентированный язык с вложением массивов JSON в JSON объект. Это приводит к необходимости нормализовать данные в виде таблицы для легкого анализа. Скрипт на питоне поможет решить эту проблему, так как все онлайн-конвертеры JSON для файлов CSV занимали слишком много времени. Был создан собственный скрипт на Python, чтобы замапить информацию через фреймы данных Pandas в объекты Series. Общий алгоритм в этого процесса:

1. Нормализация данных

Преобразуются вложенные объекты JSON в массивы и операция повторяется, пока не доходит до самого внутреннего уровня. Это выполняется функцией из пакета «pandas.io.json».

2. Адаптация данных

Преобразуется полученный объем данных в Series. Series позволяет нам проверить данные, а затем выбрать конкретный столбец, который нас интересует. Если есть дальнейшее вложение, происходит возврат к шагу 1 еще раз.

3. Извлечение данных

Теперь извлекается столбец с CWE для связанных идентификаторов по каждой уязвимости. Примечание: может быть более одного CWE номера или идентификатора для любой уязвимости.

4. Преобразование данных

Наконец, данные записываются в файл Excel. Рисунок 1 дает нам точное представление об алгоритме реализации в Python 3.

Приведенная выше реализация алгоритма создает страничные Excel файлы, которые необходимо объединить вручную в Microsoft Excel.

Далее, мы используем сводную таблицу Microsoft Excel чтобы вычислить количество значений для идентификаторов CWE по всем табли-

цам, полученным из Национальной базы уязвимостей.

```
import pandas as pd
from pandas.io.json import
json_normalize
import json
with open('nvdCVE-1.1-2019.json') as
data_file:
    data = json.load(data_file)
df = json_normalize(data,
['CVE_Items'])
cvecol = df['cve']
cvenormalized =
json_normalize(cvecol)
problemtypedata =
cvenormalized['problemtype.problemtyp
e_data']

problemtypelist =
problemtypedata.tolist()
testdf =
pd.DataFrame(problemtypelist)
details = json_normalize(testdf[0],
['description'])
cvevalues = details['value']
cvedf = pd.DataFrame(cvevalues)

cvedf.to_excel(r'cve_counts.xlsx',
sheet_name='2019',
engine='xlsxwriter')
```

Рис. 1 – Адаптация данных и преобразование кода Python

III. РЕЗУЛЬТАТЫ

В целом было несколько интересных наблюдений. Давай попробуем отвечать на них один за другим в соответствии с вопросом исследования предложенные вопросы исследования. Каковы 10 самых частых CWE за последние десять лет в Национальной базе уязвимостей?

В целом, список топ 10 CWE, сгенерированных для статьи:

1. CWE 119 - Переполнение буфера
2. CWE 79 - Межсайтовый скриптинг
3. CWE 20 - Невалидная проверка ввода
4. CWE 200 - Информационная экспозиция
5. CWE 264 - Разрешения, привилегии и доступ управления
6. CWE 84 - Неправильная нейтрализация закодированного URI схемы на веб-странице
7. CWE 310 - Криптографические проблемы
8. CWE 125 - Выход за пределы массива при чтении
9. CWE 399 - Ошибки управления ресурсами
10. CWE 352 - Подделка межсайтовых запросов

Этот список является наиболее значительным вкладом в исследование с полным подсчетом и анализом слабостей, обнаруженных в программном обеспечении с 2010 по 2019 год. Это прежде всего дает нам широкий обзор самых

слабых 10 мест безопасности в реальном мире за последнее десятилетие. Какие идентификаторы CWE используются для уязвимостей, описанных OWASP топ 10?

OWASP топ 10 описывает идентификатор CWE под разными категориями [2]. Мы не будем вдаваться в подробности относительно того, что они для краткости. Они есть:

- A1 Инъекция
- A2 Сбой аутентификации
- A3 Воздействие на конфиденциальные данные
- A4 Внешнее воздействие на XML
- A5 Взломанный доступ
- A6 Неправильная настройка безопасности
- A7 Межсайтовый скриптинг
- A8 небезопасная десериализация
- A9 Использование компонентов с известными уязвимостями
- A10 Недостаточное логирование

В приведенном выше списке «A9» особенно важен. Он говорит об использовании программных компонентов, которые, заранее известно, уязвимы. Самый простой способ устранения уязвимости в безопасности заключается в том, чтобы закрыть дыру с уязвимостью и выпустить новую версию.

Все CWE в приведенном выше списке соответствуют различным категориям с различной степенью тяжести. Но они в сумме представляют уязвимости, о которых OWASP особенно предупреждает разработчиков и просит их избегать. Какой самый частый CWE за последние десять лет в Национальной базе данных уязвимостей и занимает ли он первую строчку в топ 10 OWASP?

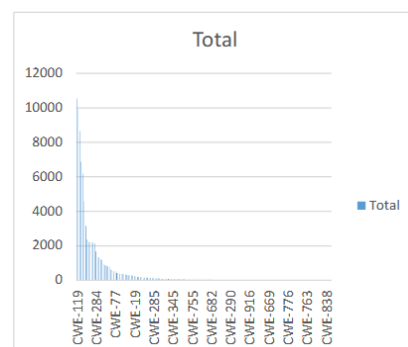


Рис. 2 – Количество идентификаторов CWE в Национальной базе уязвимостей для всех уязвимых мест 2010-2019 гг.

Самый частый из зарегистрированных CWE-119 (переполнение буфера). Это не то же самое, что OWASP Top 10 A1, который говорит

нам об инъекциях. Инъекция сообщается с CWE ID 77,78,88,89,90,81,564,917 и 943 для категории A1 в OWASP топ 10. Это можно визуально наблюдать на рис. 2 приведенном выше.

Еще одно интересное наблюдение из рис. 2 заключается в том, что OWASP действительно фокусируется на безопасности веб-приложений по сравнению с другими архитектурами, такими как настольные и мобильные. Но в зависимости от технологического стека, который используется для манипулирования данными в фоновом режиме, злоумышленники может обойти меры безопасности, предлагаемые браузером по умолчанию, а затем напрямую атаковать базовое приложение через интерфейс программирования (API).

Например, рассмотрим проблему переполнения буфера во внутреннем сервере, который прослушивает HTTP-запросы. Атакующий может специально создать запрос HTTP GET [3] с параметрами, которые могут превышать пределы буфера и таким образом вызвать переполнение буфера с критическими для безопасности последствиями [7]. Они например могут включать произвольное исполнение кода, что может привести к полному отключению основных системы. Сколько уязвимостей с соответствующими CWE в Национальной базе уязвимостей фактически присутствует в OWASP топ 10 CWE? Ответ здесь, довольно ясен: общее количество слабых мест, перечисленных в Национальной базе уязвимостей, являются частью набором уязвимостей, перечисленных в OWASP топ 10. Все перечисленные слабые места в OWASP топ 10 включены в Национальной базе уязвимостей. Но это также означает, что список топ 10 OWASP недостаточно исчерпывающий, и разработчики должны знать о проблемах, которые могут быть не включены внутри в топ.

На рис. 3 представлен общее количество идентификаторов CWE которые наблюдались в течение 2010-2019 годов в Национальной База уязвимостей, а также присутствовали в OWASP топ 10 [2]. В топ 10 OWASP есть межсайтовый скриптинг "A7". Предполагая, что список является рейтингом уязвимостей, мы можем наблюдать, что для уязвимости, которая позиционируется в самой верхней части списка, полученного в этой статье из Результаты подсчета Национальной базы уязвимостей, не совпадает с топовой уязвимостью, описанной OWASP топ 10.

Бойко Мария Владимировна, магистрантка каф. ИТАС, boikomary@gmail.com.

Научный руководитель: Навроцкий Анатолий Александрович, заведующий кафедрой информационных технологий автоматизированных систем БГУИР, доцент, navrotsky@bsuir.by.

Это интересное наблюдение можно объяснить тем, что список OWASP топ 10 имеет другие критерии по сравнению с наблюдаемыми результатами за последнее десятилетие. Возможно, OWASP лучше учитывает потребности разработчиков. Таким образом, составители топ 10 более осведомлены об ошибках, которые совершают другие инженеры.

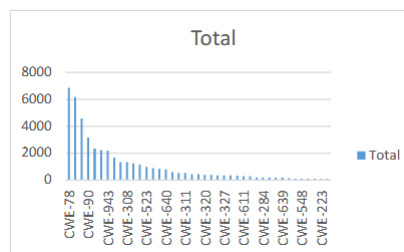


Рис. 3 – Количество идентификаторов CWE в Национальной базе данных уязвимостей для OWASP топ 10 уязвимых мест за 2010-2019 гг.

1. NVD - Data Feeds [Electronic resource] / - Nvd.nist.gov., 2020. - Mode of access: <https://nvd.nist.gov/vuln/data-feeds>. - Date of access: 06.02.2020.
2. CWE - CWE-1026: Weaknesses in OWASP Top Ten (2017)(3.4.1) [Electronic resource] / - Cwe.mitre.org., 2020. - Mode of access: <https://cwe.mitre.org/data/definitions/1026.html>. - Date of access: 08.02.2020.
3. Sridhar, M. FLASH IN THE DARK: ILLUMINATING THE LANDSCAPE OF ACTIONSCRIPT WEB SECURITY TRENDS AND THREATS / M. Sridhar, M. Chirva, B. Ferrell, K. Hamlen, D. Karamchandani. - Utdallas.edu. - Mode of access: <http://www.utdallas.edu/hamlen/sridhar17jissec.pdf>. - Date of access: 08.02.2020.
4. Garg et al, S. Network-based detection of Android malicious apps / S. Garg // International Journal of Information Security. - 2017. - Vol. 16, № 4. - P. 385-400.
5. Jelen B. Microsoft Excel 2019: Pivot Table Data Crunching / B. Jelen, M. Alexander // Pearson. - 2019. - P.512
6. Heydt, M. Learning Pandas: Get to Grips with Pandas - a Versatile and High-Performance Python Library for Data Manipulation, Analysis, and Discovery / M. Heydt // Packt Publishing. - 2015. -P. 504
7. Black, P. E. Defeating Buffer Overflow: A Trivial but Dangerous Bug / P. E. Black, I. Bojanova // IT Professional. - 2016. - Vol. 18, № 6. - P. 58-61.
8. R. N. Peclat et al, R. N. Semantic Analysis for Identifying Security Concerns in Software Procurement Edicts / R. N. Peclat et al // New Generation Computing. - 2018. - Vol. 36, № 1. - P. 21-40.

АЛГОРИТМИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ С РАЗЛИЧНОЙ АРХИТЕКТУРОЙ

Рассматриваются алгоритмы функционирования информационной системы с монолитной и микросервисной архитектурой применительно к задачам организации автоматизированных информационных систем на примере системы продажи проездных документов на железнодорожный транспорт.

ВВЕДЕНИЕ

При создании сложных автоматизированных информационных систем (приобретение проездных документов на различные виды транспорта, а также документов на посадочные места в концертные залы и стадионы и т.п.) важными этапами являются разработка структуры и алгоритмов функционирования таких систем, что влияет на эффективность, производительность, надежность и удобство эксплуатации применяемых аппаратный и программных средств. В настоящей работе исследуется информационная система приобретения проездных документов на железнодорожный транспорт.

В состав системы входит несколько модулей: регистрации пользователей; регистрации пассажиров пользователя; оформления проездных документов на поезда; оформления проездных документов на электрички; оплаты проездных документов; справочной информации.

При построении системы продажи проездных документов необходимо учитывать особенности монолитной и микросервисной архитектуры.

I. МОНОЛИТНАЯ СИСТЕМА

Для системы с монолитной архитектурой характерно нахождение всех модулей системы в одном проекте с общей кодовой базой. Основным достоинством такой системы является удобство в работе: настройка одного модуля, пусть и большого, что проще и быстрее по сравнению с настройкой приложения с микросервисной архитектурой, включающей набор отдельно настраиваемых различных модулей. Основным недостатком в сравнении с микросервисной архитектурой является сложность распределения ресурсов между модулями, так как система с монолитной архитектурой содержит их внутри себя, что не позволяет распределять ресурсы между модулями. В ряде случаев это играет ключевой фактор при построении крупных систем и сильно влияет на отказоустойчивость.

Схема алгоритма функционирования системы продажи документов на железнодорожный транспорт с монолитной архитектурой изображена на рисунке 1.

Когда пользователь отправляет запрос, система через API-шлюз принимает его и обраба-

тывает соответствующими модулями. Если, например, пользователь оформляет проездной документ на поезд, то вся система обрабатывает его запрос, при этом выполняя функции, соответствующие модулю оформления проездных документов на поезда, а остальные модули в это время не работают ввиду отсутствия необходимости, и недоступны извне, так как система обрабатывает конкретный запрос. Конечно же модули взаимодействуют между собой, поэтому после оформления проездного документа, пользователь будет перенаправлен на оплату оформленного заказа, или, к примеру, для оформления заказа система предложит варианты транспорта по некоторым критериям поиска из модуля справочной информации. Здесь стоит отметить преимущество монолитной архитектуры в прямом доступе к остальным модулям. И в дальнейшем при обработке данных по оплате заказа система будет задействована, используя модуль оплаты проездных документов, и в это время остальные модули не используются и не доступны извне.

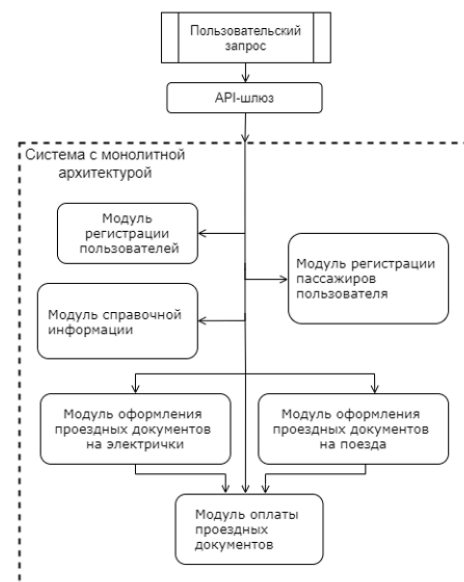


Рис. 1 – Схема алгоритма функционирования системы с микросервисной архитектурой

Управление ресурсами такой системы происходит следующим образом: что бы система могла обрабатывать несколько запросов одновременно, аппаратное обеспечение выполняет обработку данных в нескольких параллельных

процессах. Тогда для создания отдельного процесса для обработки запроса другого пользователя в отдельном потоке запускается копия системы. Если модуль оплаты заказов используется в два раза чаще каждого из модулей оформления проездных документов, потому что оплачивать необходимо оба вида проездных документов, то для поддержания стабильности работы ресурсы необходимо рассчитывать так, чтобы самый нагруженный модуль системы выдерживал нагрузку и обрабатывал стабильно все запросы пользователей.

II. МИКРОСЕРВИСНАЯ СИСТЕМА

Для системы с микросервисной архитектурой характерна самостоятельность каждого из модулей, так как фактически они являются отдельными приложениями. Это позволяет настраивать и распределять ресурсы для каждого модуля в отдельности. Основным и важнейшим преимуществом такой системы является увеличение отказоустойчивости, когда при недоступности или выходе из строя одного модуля все остальные продолжают работать в своем предопределённом настройкой режиме. Для больших систем с множеством различного функционала зачастую этот фактор может являться наиболее значимым. Схема алгоритма функционирования системы с микросервисной архитектурой изображена на рисунке 2.



Рис. 2 – Схема алгоритма функционирования системы с монолитной архитектурой

Система принимает пользовательский запрос и через API-шлюз направляет его на обработку соответствующему модулю. Таким образом, если пользователь направляет запрос на

оформление проездного документа на поезда, то его запрос поступит на соответствующий модуль, а все остальные модули системы будут доступны и могут обрабатывать запросы других пользователей. При этом модули могут взаимодействовать между собой, и не важно на одном ли физическом устройстве они находятся, поскольку взаимодействуют они как независимые приложения посредством сети Интернет.

Достоинством такой системы является возможность настройки каждого модуля. При ситуации, когда, к примеру, высоконагруженный модуль не справляется с потоком запросов, появляется возможность увеличения производительности такого модуля методом создания копии.

Копии могут находиться на различных аппаратных средствах. К примеру, в регионе, где интенсивность потока запросов выше, установить две копии модуля на более мощном сервере, а в регионе с меньшей интенсивностью – две копии на более слабом серверном оборудовании. В общем случае это уже особенности настройки, но показать ее гибкость важно, потому что это одно из ключевых преимуществ системы с микросервисной архитектурой.

III. ВЫВОДЫ

Иногда именно благодаря настройке принципа функционирования можно значительно увеличить эффективность системы, а микросервисную систему можно более гибко настраивать, так как каждый ее модуль – самостоятельный. Появляется возможность создать копию отдельного модуля, а не полностью всей системы. При этом увеличивается отказоустойчивость системы. Это становится важным фактором для крупных систем с большим набором различных функций, которые удобно разнести в отдельные модули.

Окончательные рекомендации по эффективности рассмотренных структур могут быть даны в результате моделирования системы с учетом конкретных исходных данных относительно размерности системы, интенсивности входящих информационных потоков и возможностей используемых ресурсов.

1. Крис Ричардсон. Микросервисы. Паттерны разработки и рефакторинга / К.Ричардсон – Изд-во: Издательский дом "Питер", Санкт-Петербург, 2016. – 544 с.
2. Сэм Ньюмэн. Создание микросервисов / С.Ньюмэн. – Изд-во: Издательский дом "Питер", Санкт-Петербург 2016. – 304 с.
3. Интернет-ресурс <https://habr.com/ru/>

Бортник Артем Владимирович, магистрант кафедры информационных технологий автоматизированных систем, ttemmik97@gmail.com

Научный руководитель: Лукьянец Степан Валерьянович, кандидат технических наук, профессор, Lukyansstvtv@gmail.com

ВИЗУАЛИЗАЦИЯ ДАННЫХ В ВИРУТАЛЬНОЙ РЕАЛЬНОСТИ

В данной работе предлагается использование виртуальной реальности для визуализации различных видов данных.

ВВЕДЕНИЕ

В наше время требуется обрабатывать большие потоки данных. Чем больше данных, тем сложнее находить взаимосвязи между ними. Человеческий мозг не способен уловить закономерности в больших потоках данных, поэтому необходимо находить решения для визуализации информации. Многоуровневые модели, сложные механизмы, статистические зависимости, а также моделирование в реальном времени — если вы хотите что-то показать, используйте для этого иммерсивные технологии. Их возможности вне конкуренции.

I. ИСПОЛЬЗОВАНИЕ ЦИФРОВОЙ СРЕДЫ ДЛЯ DATAVIZ

Фраза «внутри виртуальной реальности» означает нечто совершенно иное на сегодняшний день. Дополненная и виртуальная реальность очень важны масштабному государственному сектору тем, что помогают наглядно визуализировать большое количество данных. К примеру, в начале 2017 года на развитие бизнеса в этой области стартап Virtualitics получил 3 миллиона долларов. Многие эксперты DataViz могут не воспринимать всерьез идею использования DataViz в VR и могут считать это бессмысленной тратой ресурсов. Тем не менее, я думаю, что эта технология имеет большой потенциал в данной области. При работе с появляющейся новой технологией и средой наступает период неопределенности в отношении лучших практик проектирования. Многие люди с самого начала совершают ошибки, но это нормально, потому что это часть процесса развития. Но в итоге формируются стандарты работы.

II. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ВИРУТАЛЬНОЙ РЕАЛЬНОСТИ ДЛЯ DATAVIZ

Вероятно, главные проблемы возникнут из-за неправильного понимания человеческого восприятия. Уже из некоторых проектов DataViz VR я вижу, что они испытывают те же проблемы, что и 3D-диаграммы, несмотря на то, что они находятся в виртуальной реальности и ин-

терактивны. Например, если вы преобразуете 2D гистограмму в 3D и просто поместите ее в среду виртуальной реальности, вы все равно получите искажения, возникающие при просмотре диаграммы под разными углами. Некоторые диаграммы будут хорошо работать, будучи преобразованными в VR, тогда как другие будут бесполезны.

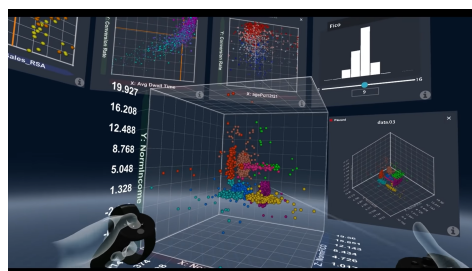


Рис. 1 – Пример не правильного использования VR.

Один из интересных примеров, который я нашел: автор адаптировал временную шкалу в «американские горки». Мне кажется, это интересное решение задачи.

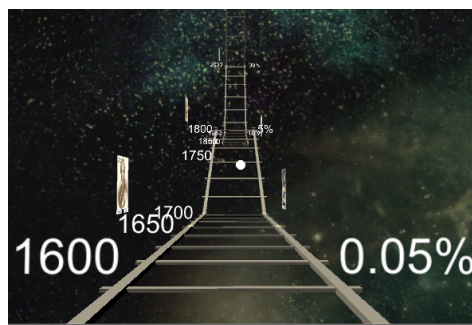


Рис. 2 – Пример работы с данными в VR.

III. ВЫВОДЫ

При правильном использовании цифровой среды для DataViz можно добиться больших успехов во многих направлениях.

1. Луисо Томмазо Де Паолис, Патрик Бурдо, Антонио Монжелли, Дополненная реальность, Виртуальная реальность и компьютерная графика.

Василевский Алексей Николаевич, магистрант 2 курса факультета информационных технологий и управления БГУИР, kinogoblin@gmail.com.

Научный руководитель: Шилин Леонид Юрьевич, профессор, д.т.н.

ВЕБ-ТЕХНОЛОГИИ ДЛЯ РЕШЕНИЯ ЛОГИСТИЧЕСКИХ ЗАДАЧ ЖЕЛЕЗНОДОРОЖНЫХ ГРУЗОПЕРЕВОЗОК

Рассматриваются возможности использования веб-технологий для решения задач транспортного предприятия на примере использования модуля ведения прейскурантов для расчёта грузового тарифа.

В настоящее время, в мире непрерывного потока информации, все большую актуальность приобретают процессы автоматизации работ рутинного характера. В связи с этим разрабатываются системы, призванные решить возникающие проблемы.

Услуга расчета грузового тарифа включает веб-интерфейс ведения электронной базы тарифов на перевозку грузов, дополнительные работы и услуги, в том числе ведение фиксированных ставок и скидков, обеспечивает возможность оперативно вносить изменения в программу расчета минимизировав при этом человеческий фактор. Веб-интерфейс обеспечивает централизацию ведения таблиц с описанием условий, требуемых по документам- прейскурантам. Для выбора прейскуранта необходимо задать Период – даты начала и конца действия документа. Далее обеспечивается возможность создания, копирования, работа с документом и каждым задаваемым параметром.

За платформу разработки был взят фреймворк Angular 2. Для формирования интерфейса была взята библиотека Bootstrap4.

Общий вид интерфейса системы имеет вид страницы с панелью-контейнером. Панель-контейнер состоит из вкладки переключения функциональных подсистем веб-модуля и зоны размещения элементов интерфейса выбранной подсистемы. На рис. 1 располагается общий макет интерфейса веб-модуля.



Рис. 1 – Общий макет интерфейса веб-модуля

Интерфейс ведения прейскурантов включает список документов и страницы задаваемых параметров.

Высоцкий Евгений Сергеевич, Аспирант кафедры информационных технологий автоматизированных систем БГУИР, evgenyvy95@gmail.com.

Научный руководитель: Гуринович Алевтина Борисовна, заместитель декана ФИТиУ, кандидат технических наук, доцент, gurinovich@bsuir.by.

Интерфейс формирования параметров поиска заданного прейскуранта представлен на рисунке рис. 2

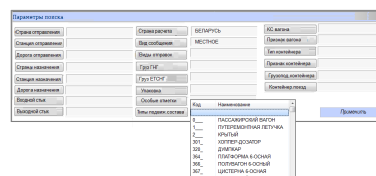


Рис. 2 – Макет страницы параметров поиска прейскуранта

На рисунке 3 приведена компоновка элементов интерфейса страницы детальных ведения прейскуранта. Из рисунка видно, что детальные данные результатов ведения прейскуранта представлены таблицей из колонок наименования и значения параметра.

имя 1	значение 1
имя 2	значение 2
имя 3	значение 3
...	

Назад

Рис. 3 – Макет страницы детальных данных прейскуранта

Вывод

Таким образом, применение данной системы позволяет ускорить бизнес процесс и обеспечить автоматизацию железнодорожных транспортных услуг. При необходимости система может быть расширена, в неё могут быть добавлены другие подсистемы.

РАСПРЕДЕЛЕННЫЙ СЕРВИС СЕРТИФИКАЦИИ НА БЛОКЧЕЙН ПЛАТФОРМЕ ETHEREUM

В работе рассматриваются вопросы, необходимые для создания и управления распределенной сетью для хранения сертификатов, а также возможные алгоритмы решения этой задачи.

ВВЕДЕНИЕ

Ethereum — это распределенная сеть открытых блоков, которая фокусируется на запуске программного кода любого децентрализованного приложения. Проще говоря, это всемирная платформа для обмена информацией, которой нельзя манипулировать и которую нельзя изменить. Реализована как единая децентрализованная виртуальная машина. Был предложен основателем журнала Bitcoin Magazine Виталием Бутериным в конце 2013 года, сеть была запущена 30 июля 2015 года. Технология Ethereum даёт возможность регистрации любых сделок с любыми активами на основе распределённой базы контрактов типа блокчейн, не прибегая к традиционным юридическим процедурам. Эта возможность является конкурентной по отношению к существующей системе регистрации сделок

I. Цели создания системы

Одно из не самых очевидных применений блокчейн-технологий — сертификация товаров и услуг. Особенности технологии распределенного реестра отлично подходят для защиты продукции от фальсификации. Это выгодно и для производителей, и для продавцов, и для конечных потребителей. Разберем применение распределенного реестра и его технологий для сертификации алкогольной продукции — одной из наиболее часто фальсифицируемых категорий товаров. Как сегодня производители ведут борьбу с подделками? Наклеивают специальные этикетки с водяными знаками и голограммами, что не дает практически никакого положительного эффекта. Многие изготовители контрафакта с легкостью подделывают любые знаки. Проблема усугубляется тем, что далеко не все конечные потребители хорошо разбираются в алкоголе. Они не могут отличить настоящую продукцию от поддельной не то, что во время выбора и покупки, но и даже по итогам дегустации напитка. В результате производитель теряет существенную долю прибыли, государство недополучает налоги (фальсификаторы их, естественно, не оплачивают), а торговцы вынуждены терпеть нападки покупателей.



Рис. 1 – Программная модель

Одно из решений проблемы — это оснащение бутылок с алкогольной продукцией так называемыми «умными» этикетками. Каждая такая этикетка содержит уникальный электронный код соответствия оригинальному качеству с указанием времени и места производства продукта. В отличие от бумажных этикеток, штрих-кодов и различных голограмм подделать электронный код не представляется возможным независимо от используемых средств. Вернемся к примеру с бутылкой алкоголя, сертифицированной с применением блокчейна и укомплектованной электронным кодом. Каждую единицу товара уникальный код сопровождает на каждом этапе следования от производителя до конечного потребителя. Для проверки подлинности продукта достаточно вбить код в общую базу, и просмотреть подробные сведения по конкретной бутылке или другой единице товара. Все, что для этого нужно — компьютер или любое мобильное устройство с подключением к Интернету. Проверить оригинальность может не только покупатель, но и продавец после получения партии товара от поставщика. Этим он уберезет себя от реализации некачественных товаров.

II. ОПИСАНИЕ СИСТЕМЫ

На рисунке 2 представлена блок-схема алгоритма системы.

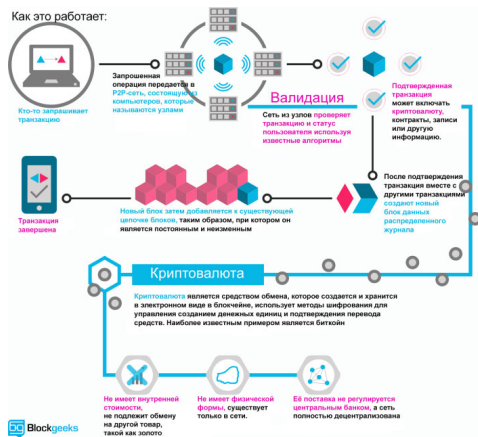


Рис. 2 – Блок-схема алгоритма системы

Блокчейн (цепочка блоков) — это распределенная база данных, у которой устройства хранения данных не подключены к общему серверу. Эта база данных хранит постоянно растущий список упорядоченных записей, называемых блоками. Каждый блок содержит метку времени и ссылку на предыдущий блок. Если объяснять на пальцах, то блокчейн часто сравнивают со стандартным дневником или картотекой, куда последовательно в хронологическом порядке вносятся записи о том, что сделано – поспал, поел, постирал, погулял, взял в долг, заплатил 100 долларов за ужин и т.д. Чтобы никто посторонний не мог внести по своему усмотрению изменения в дневник, вся информация особым образом шифруется, причем шифр продуман качественно. Если дневник в одном экземпляре, с ним всякое может случиться – сгорел дом и он вместе с ним, украли, при огромнейшем желании расшифровали и внесли коррективы. А потому для надежности дневник имеет множество копий, которые хранятся в разных местах. Причем, когда в дневник вносится новая информация, она после провер-

ки обновляется на всех экземплярах. Применение шифрования гарантирует, что пользователи могут изменять только те части цепочки блоков, которыми они «владеют» в том смысле, что у них есть закрытые ключи, без которых запись в файл невозможна. Кроме того, шифрование гарантирует синхронизацию копий распределенной цепочки блоков у всех пользователей. В технологию блокчейн изначально заложена безопасность на уровне базы данных. Концепцию цепочек блоков предложил в 2008 г. Сатоши Накамото (Satoshi Nakamoto). Впервые реализована она была в 2009 г. как компонент цифровой валюты — биткойна, где блокчейн играет роль главного общего реестра для всех операций с биткойнами. Благодаря технологии блокчейна биткойн стал первой цифровой валютой, которая решает проблему двойных расходов (в отличие от физических монет или жетонов, электронные файлы могут дублироваться и тратиться дважды) без использования какого-либо авторитетного органа или центрального сервера.

III. Выводы

В мире существует потребность в системе, которая способна сохранять и отслеживать все данные относящиеся к сертификации, которая сможет облегчить процесс сертификации, сделать его прозрачнее и надежнее.

1. Карпов О. Э., Клименко Г. С., Лебедев Г. С. Создание смарт-контрактов Solidity для блокчейна Ethereum. Практическое руководство. 2016. № 2. – С. 7–23.
2. Alex Popp В. Г. Mastering Ethereum is a book for developers, 2016.–№ 6–7. – С. 32–33.
3. Лебедев Г. С., Мухин Ю. Ю. Блокчейн изнутри // часть 2, 2015, с. 98–105.

Воробьев Егор Александрович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, e.vorob8@gmail.com.

Научный руководитель: Свито Игорь Леонтьевич, доцент кафедры ТОЭ БГУИР, кандидат технических наук, доцент, kaftoe@bsuir.by.

ПОДХОДЫ К ПОСТРОЕНИЮ ХРАНИЛИЩ ДАННЫХ

Рассматриваются два основных подхода к построению хранилища данных. Предлагается комбинированный метод построения хранилища данных для совмещения преимуществ основных подходов.

ВВЕДЕНИЕ

На сегодняшний день существует два основных подхода к построению Хранилища данных. Это хранилище данных с архитектурой шины Ральфа Кимбола и с корпоративной информационной системой Билла Инмона.

I. ПОДХОД БИЛЛА ИНМОНА

Рассматриваемая структура хранилища данных представлена на рисунке 1.

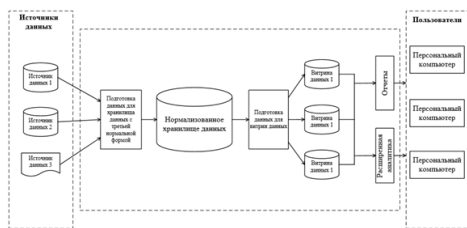


Рис. 1 – Структурная схема хранилища данных с корпоративной информационной системой

Данные согласованно извлекаются из источников в централизованное хранилище с третьей нормальной формой, откуда поступают в витрины данных для дальнейшего анализа. Такая структура предоставляет возможности для манипулирования атомарными данными, но не обеспечивает высокой производительности.

II. ПОДХОД РАЛЬФА КИМБОЛА

Структура рассматриваемого хранилища данных представлена на рисунке 2.

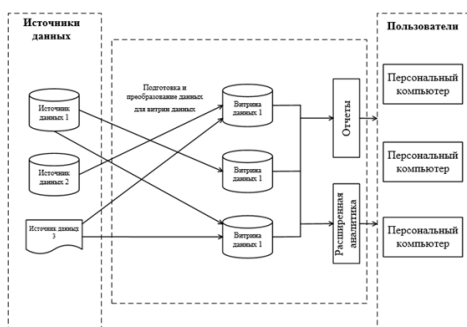


Рис. 2 – Структурная схема хранилища данных с архитектурой шины

Геврасёва Ирина Петровна, магистрант кафедры информационных технологий автоматизированных систем БГУИР, irina_gevraseva@mail.ru.

Научный руководитель: Навроцкий Анатолий Александрович, заведующий кафедрой информационных технологий автоматизированных систем БГУИР, кандидат физико-математических наук, доцент, navrotsky@bsuir.by.

Данные из источников поступают непосредственно в пространственно структурированные витрины данных, позволяет добиться высокой производительности без дополнительного перепрограммирования. С другой стороны, в этой схеме нет необходимой гибкости структуры.

III. КОМБИНИРОВАННЫЙ МЕТОД ДЛЯ ПОСТРОЕНИЯ ХРАНИЛИЩЕ ДАННЫХ

Рассмотрим комбинированное хранилище, структура которого представлена на рисунке 3.

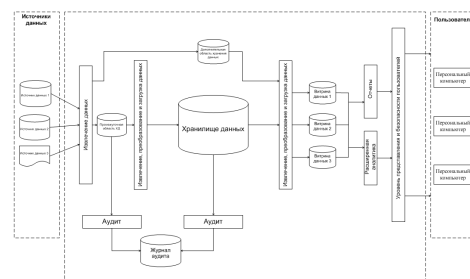


Рис. 3 – Структурная схема комбинированного хранилища данных

Данные извлекаются из источников данных в стейджи данных. Затем данные преобразуются и поступают в пространственно структурированное хранилище, откуда поступают в витрины данных для дальнейшего анализа. Для уменьшения задержки попадания данных к конечному пользователю предлагается использовать дополнительную область хранения данных, заполняемую чаще, чем хранилище данных.

ВЫВОДЫ

Предложенный подход обеспечивает высокую производительность и гибкость работы с атомарными данными. Минусом данной системы является ее сложность и ресурсоемкость.

Список литературы

1. Kimball, R. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition / R. Kimball, M. Ross // Wiley Publishing, Inc. – 2013. – С. 564.
2. Inmon, W. H. Building the Data Warehouse, Fourth Edition / W. H. Inmon // Wiley Publishing, Inc. – 2005. – С. 576.

SAP HANA КАК ПЛАТФОРМА ДЛЯ ПОСТРОЕНИЯ КОРПОРАТИВНОГО ХРАНИЛИЩА ДАННЫХ

В данной работе рассматривается такой подход к решению бизнес-задач, как реализация корпоративного хранилища данных.

ВВЕДЕНИЕ

SAP HANA – реляционная СУБД, разработанная компанией SAP в 2011 году, особенность которой является размещение базы данных в оперативной памяти, благодаря чему достигается высокая скорость выполнения запросов на извлечение данных. Она представляет собой многофункциональную платформу, включающую СУБД, функционал для обработки данных, сервер приложений, несколько языков программирования и множество других инструментов [1].

I. КОРПОРАТИВНОЕ ХРАНИЛИЩЕ ДАННЫХ

Корпоративное хранилище данных (Data Warehouse) - это база данных, специально разработанная и предназначенная для подготовки отчётов и бизнес-анализа с целью поддержки принятия решений в организации. Основные требования к такой базе данных – это быстрота отклика, гибкость в интеграции с информационными системами, устойчивость к нагрузкам, а также обеспечение безопасности корпоративной информации. Характеризуется следующими свойствами: служит для поддержки принятия решений; содержит как атомарные, так и агрегированные данные; содержит как исторические, так и текущие данные; данные не изменяются пользователями; данные загружаются в хранилище с определенной периодичностью; объединяет данные из множества предметных областей. Классическую архитектуру построения корпоративного хранилища данных, предложенная Ральфом Кимболлом, можно увидеть в следующем рисунке (см.рис.1).

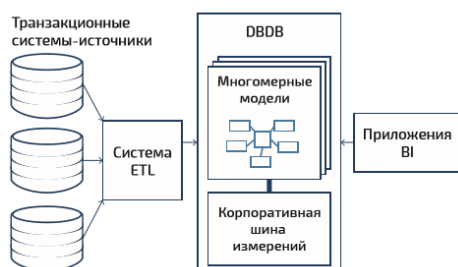


Рис. 1 – Классическая архитектура построения корпоративного хранилища данных

Ядром этой архитектуры является система ETL. Консолидация данных осуществляется за счет использования так называемой корпоративной шины измерений, то есть определенного набора согласованных измерений, используемых многомерными моделями предприятия. Недостатком данной архитектуры является невозможность получить аналитику над ранее полученными данными учетной системы из-за временных задержек, создаваемых системой ETL. При использовании SAP HANA как платформы для построения корпоративного хранилища данных возможно реализовать корпоративную шину измерений с помощью информационных представлений данных (см.рис.2). Многомерные модели данных, которые также реализуются с помощью представлений данных, при такой схеме используют измерения из корпоративной шины.



Рис. 2 – Архитектура корпоративного хранилища данных с использованием SAP HANA

II. ЗАКЛЮЧЕНИЕ

Таким образом, SAP HANA позволяет реализовать корпоративное хранилище данных, с помощью которого возможно получать аналитику над операционными данными в режиме реального времени.

Список литературы

1. Berg B., Silvia P. SAP HANA: An Introduction. – SAP Press, 2012.
2. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition. – John Wiley Sons, Inc., 2013.

Долгая Елена Сергеевна, студентка 3 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники

АЛГОРИТМ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ВРАЧЕБНЫМ ПЕРСОНАЛОМ ДЛЯ УЧРЕЖДЕНИЙ ЗДРАВООХРАНЕНИЯ

В работе рассматриваются вопросы, необходимые для управления учреждениям здравоохранения, а также возможные алгоритмы решения этой задачи.

ВВЕДЕНИЕ

Текущее состояние развития информатизации здравоохранения в Республики Беларусь ставит в ряд актуальных задачу модернизации информационных систем всех медицинских организаций. В настоящей статье всесторонне рассматриваются вопросы модернизации и развития электронной медицинской системы медицинской организации. Основное внимание уделяется автоматизированной системе управления ресурсами медицинской организации. Предложенные подходы могут быть использованы руководителями медицинских организаций, руководителями информационных подразделений для построения системы нормативных документов, организующих работы по модернизации и развитию медицинских информационных систем.

I. ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

Программная модель была разработана на основе исследований и анализа медицинских учреждений в РБ. Эта модель, показанная на рисунке 1, подходит для большинства учреждений здравоохранения, нуждающихся в ведении, хранении и совместном использовании большого количества медицинских карт пациентов. Чтобы преодолеть недостатки в управлении информацией в учреждениях здравоохранения, у каждого пациента будет уникальный идентификатор (PID).

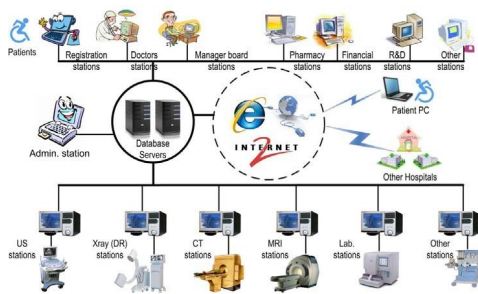


Рис. 1 – Программная модель

Это позволит включать всю информацию о пациентах в базу данных сервера в соответствии с их PID. Как только информация о пациенте передается из клинических станций в базу данных, врач входит в систему, используя уникальный PID, и просматривает всю информацию о

пациенте в базе данных сервера, чтобы помочь в диагностике и лечении. Кроме того, система будет предоставлять удаленные медицинские услуги, такие как онлайн информирование пациентов посредством входа в систему через Интернет, или передачи информации о пациентах в другие учреждения здравоохранения, когда пациентам требуется перевод.

II. ОПИСАНИЕ СИСТЕМЫ

На рисунке 2 представлена блок-схема алгоритма системы.

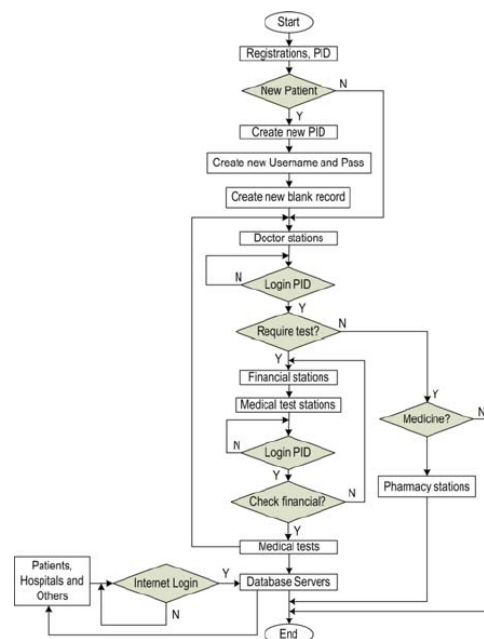


Рис. 2 – Блок-схема алгоритма системы

Во-первых, пациенты будут проходить регистрацию, если это новый пациент. Система будет создавать новый PID в базе данных записей, которые включают себя имя пользователя и пароль, соответствующие этому PID. Затем пациент пойдет в отделение общей врачебной помощи с предоставленным PID. На основании PID, врач войдет в базу данных, чтобы просмотреть основную информацию о пациенте для диагностики состояния здоровья пациента. В случае необходимости дополнительных анализов для диагностики, врач поместит запрос на тестирование в базу данных пациента. Затем пациенту будут назначены анализы, соответствующие запро-

су врача. На испытательной станции медперсонал войдет в базу данных, чтобы проверить необходимую информацию. Если запрос действителен, они проведут тесты и отправят результаты в базу данных. Затем пациент вернется в кабинет врача, чтобы получить результаты медицинского осмотра и лечение. При необходимости пациенты отправляются в аптеку, чтобы купить лекарства, прежде чем покинуть больницу. Дома, в учреждениях здравоохранения или в других местах, где есть доступ в Интернет, пациенты могут войти в систему с помощью имени пользователя и пароля, которые были предоставлены в больнице, и просмотреть свои медицинские записи, диагноз врача и результаты лечения.

III. Выводы

В РБ существует потребность в веб-системе, чтобы сделать медицинские записи доступными для медицинского персонала и пациентов, которая улучшат систему здравоохранения.

1. Карпов О. Э., Клименко Г. С., Лебедев Г. С. Развитие электронной системы здравоохранения // Федерализм. 2016. № 2. – С. 7–23.
2. Кудрина В. Г. Обзор современных отечественных исследований в области информатизации здравоохранения. – Информационные технологии в здравоохранении, 2001.–№ 6–7. – С. 32–33.
3. Лебедев Г. С., Мухин Ю. Ю. Классификация медицинских информационных систем // часть 2, 2012, с. 98–105.

Герилевич Роман Сергеевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, roman.herilovich@gmail.com.

Научный руководитель: Свито Игорь Леонтьевич, доцент кафедры ТОЭ БГУИР, кандидат технических наук, доцент, kaftoe@bsuir.by.

ОБРАБОТКА РЕЗУЛЬТАТОВ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ

В статье исследуются проблемы актуальности применения автоматизированного тестирования и вопросы, которые возникают в ходе использования данного подхода к тестированию программных продуктов. Производится описание системы, предлагающей инструменты для их решения.

ВВЕДЕНИЕ

Автоматизированное тестирование программного обеспечения стало неотъемлемой частью процесса разработки ПО. Это является одним из мировых трендов для компаний, занимающихся ИТ-инженерией. Использование автоматизированного тестирования позволяет оптимизировать процесс разработки программного продукта основные цели данного подхода представлены на рисунке 1.



Рис. 1 – Цели автоматизированного тестирования

Преимущества такого подхода к тестированию программных продуктов бесспорны, однако, существует ряд недостатков. Главная проблема заключается в отсутствии централизованного управления результатами автотестов. Это влечёт за собой невозможность получения информации о состоянии автоматизации в целом, например, менеджерами проекта и заинтересованными лицами.

I. АКТУАЛЬНОСТЬ СИСТЕМЫ

Компании, использующие автоматизированное тестирование в процессе разработки ПО, вынуждены мириться с недостатками данного подхода. Проблемы связаны, в основном, со структуризацией полученных результатов автотестов, приведением результатов тестирования различных частей программного продукта к одному виду и организацией простого доступа к этим данным любому члену команды.

Автоматизация тестирования предполагает написание программного кода, тестирующего

работу другого кода. В общем случае, различные части программного продукта реализуются с использованием разных технологий и языков программирования, что влечёт за собой и многообразие платформ для их тестирования. Таким образом получение актуальных данных о состоянии автоматизации и, следовательно, работоспособности функционала ПО, является задачей, требующей непосредственного контакта управляющего звена с группами тестировщиков-автоматизаторов. Это создаёт дополнительные временные затраты на обсуждение и визуализацию результатов тестирования.

В связи с этим указанные недостатки автоматизированного тестирования дают толчок к созданию инструмента агрегации и визуализации результатов автотестов, а также совместной работы с ними.

II. ПОСТАНОВКА ЗАДАЧИ

Системе обработки результатов автоматизированного тестирования необходима поддержка следующих возможностей:

- Универсальность. Представляет собой возможность системы работать с результатами автотестов независимо от платформы тестирования.
- Визуализация данных. Система включает в себя возможность графического отображения результатов тестирования в виде графиков и диаграмм.
- Совместный доступ. Предполагает собой возможность организации совместной работы с результатами тестирования в системе с поддержкой разграничения прав доступа к данным.
- Гибкость. Включает в себя возможность продукта по интеграции со сторонними сервисами, включая существующие системы отслеживания дефектов и сервера электронной почты.

III. ПРИНЦИП РАБОТЫ

Архитектура системы представляет собой сетевую клиент-серверную архитектуру. Совокупность взаимодействующих между собой по-

ЗАКЛЮЧЕНИЕ

средством сетевых протоколов сервисов - серверная часть. Сторона клиента представлена группой посредников, так называемых «агентов», являющихся связующим звеном между платформами для тестирования и серверной частью приложения. Для визуализации данных и упрощения работы с ними также реализуется сервис пользовательского интерфейса. Взаимодействие между «агентами» и серверной частью системы осуществляется через API (Application programming interface). Структурная схема системы представлена на рисунке 2.

Для реализации серверной части системы используется объектно-ориентированный язык программирования Java совместно со Spring Framework, система управления базами данных PostgreSQL. Все используемые технологии являются программным обеспечением с открытыми исходным кодом. Это даёт доступ к большому количеству инструментов и сообществу, что способствует успешному процессу разработки.

Поддержка необходимой платформы тестирования требует реализации соответствующего «агента», «понимающего» язык платформы и «умеющего» работать с API сервера.

Гончарик И. А., магистрант факультета информационных технологий и управления БГУИР, amsterget@gmail.com

Научный руководитель: Гуринович Алевтина Борисовна, кандидат физ.-мат. наук, доцент, gurinovich@bsuir.by

В работе показано, что автоматизируя тестирование работоспособности на соответствие требованиям запроюктированного функционала программного продукта, можно существенно уменьшить риски получения некачественного продукта и оптимизировать затраты на разработку. При переводе описания тест-кейсов в код, тестировщик-автоматизатор получает задокументированную часть функционала системы и возможность в любой момент проверить это соответствие. Для получения полного представления о готовности программного продукта, необходимо систематизировать результаты тестирования всего приложения и произвести их визуализацию. Система обработки результатов автоматизированного тестирования призвана помочь в решении данных задач.

1. Автоматизированное тестирование программного обеспечения / Э. Дастин, Дж. Рэшка, Д. Пол // Издательство: Лори, 2003. – 580 с.
2. Experiences of Test Automation: Case Studies of Software Test Automation / D. Graham, M. Fewster // Addison-Wesley, 2012. – 607 с.
3. Problems with Test Automation and Modern QA [Electronic resource] / A. Ghahrai. – 2018. – Mode of access: <https://www.testingexcellence.com/problems-test-automation-modern-qa/>.

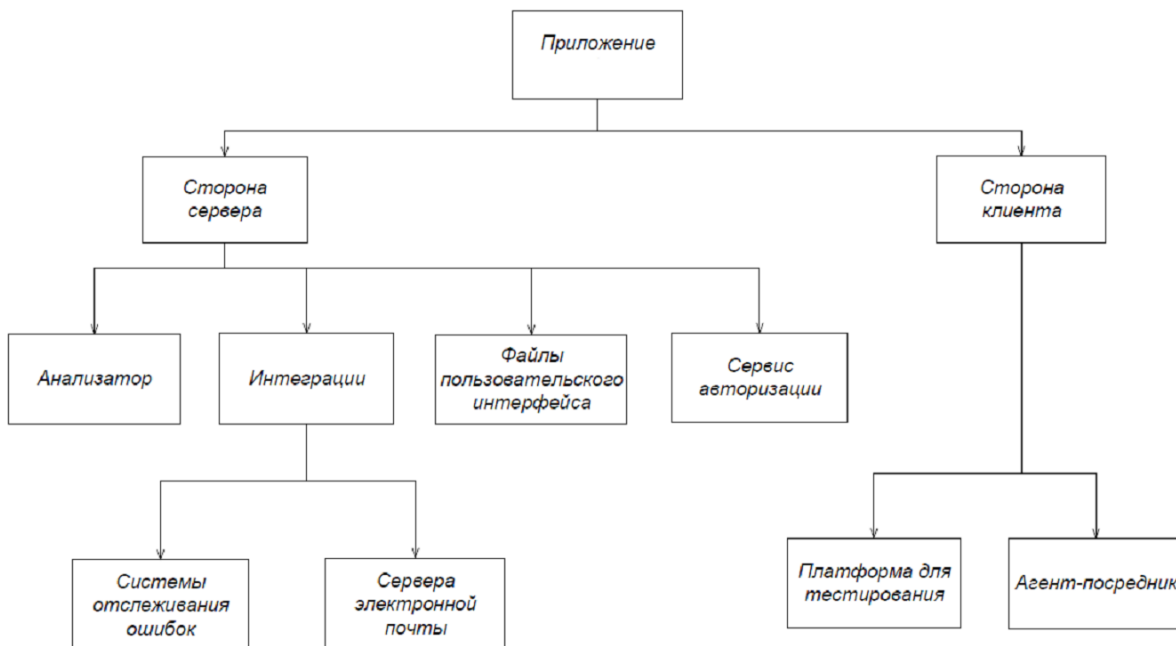


Рис. 2 – Структурная схема системы

СРАВНИТЕЛЬНЫЙ ОБЗОР ТЕХНОЛОГИЙ АНАЛИЗА ДАННЫХ В ПАРАЛЛЕЛЬНЫХ СИСТЕМАХ

ВВЕДЕНИЕ

Несоответствие между чрезмерным ростом объемов данных и тенденциями улучшения скорости обработки и доступа к памяти требует параллельной обработки, применяемой для обработки больших объемов данных. Как объемы данных, так и скорость обработки находятся на экспоненциально растущих траекториях, но первый вырос гораздо быстрее, чем второй. Отсюда следует, что параллельная обработка необходима для преодоления разрыва. В дополнение к обеспечению более высокой способности анализа. В статье представлен сравнительный анализ параллельной СУБД и MapReduce.

I. ПАРАЛЛЕЛЬНАЯ СУБД

Традиционный подход к анализу данных заключается в загрузке данных в базу данных и использование языка запросов для выполнения анализа. Параллельные СУБД были разработаны для повышения производительности систем баз данных. Она может быть определена как СУБД, которая работает на нескольких узлах. Они поддерживают стандартные реляционные таблицы, обычно разделенных на несколько узлов, и используют язык структурированных запросов (SQL).

В параллельной СУБД существует несколько различных типов архитектур:

- совместное использование памяти: несколько процессоров совместно используют пространство памяти и доступ к дискам;
- Общий диск: несколько процессоров имеют общий доступ к одному и тому же дисковому пространству, но каждый имеет свою собственную память;
- Ничего не делиться: каждый узел имеет свою память и дисковое пространство.

Важным моментом является то, что архитектуры без общего доступа перемещают только вопросы и ответы между узлами, в то время как две другие архитектуры будут перемещать данные через сеть присоединения. Основное преимущество нескольких процессоров без разделения ресурсов состоит в том, что они могут быть масштабированы до сотен и, возможно, тысяч процессоров, которые не мешают друг другу. Альтернативой параллельной СУБД, MapReduce, описана в следующем разделе.

II. MAPREDUCE

MapReduce - это метод обработки больших объемов распределенных данных, позволяющий

использовать кластеры без общего доступа. В распределенных архитектурах кластер без общего доступа - это кластер, в котором узлы кластера не разделяют ни общее дисковое пространство, ни общий ЦП, ни общую память [1].

Работа MapReduce основана на разбиении обработки данных на две фазы: фазу отображения (map) и фазу свертки (reduce). Каждая фаза использует в качестве входных и выходных данных пары «ключ-значение». Сперва данные анализируются и некоторые вычисления завершаются. Эти задачи называются «Map» задачами. Затем данные перераспределяются по всем узлам кластера. Далее, второй набор задач выполняется параллельно каждым узлом в разделе данных, которые он получает. Эти задачи называются задачами «Reduce», принимая результаты задач карты и объединяя их вместе.

MapReduce позволяет пользователю сосредоточиться на реализации логики, необходимой для решения их конкретной проблемы. Для этого пользователь должен написать программу, которая интегрируется с библиотекой MapReduce.

При выполнении сложных алгоритмов MapReduce общий подход заключается в добавлении дополнительных циклов MapReduce, а не в попытке решить все элементы за один раз. Точная реализация MapReduce доступна только для Google. Тем не менее, парадигма была реализована и другими компаниями. Самыми популярными в коммерческом плане является Hadoop и Spark.

MapReduce имеет несколько выдающихся преимуществ [2]:

- Отказоустойчивость. Отказы являются общими для компьютерного кластера с тысячами узлов. MapReduce может справляться с мелкими сбоями, минимизировать объем потерянной работы и не требует перезапуска задания с нуля.
- Гибкость: входные данные MapReduce могут иметь любой формат вместо конкретной схемы.
- Независимость: MapReduce также зависит от системы хранения. Поддерживаемые системы хранения включают файлы, хранящиеся в распределенной файловой системе, результаты запросов к базе данных, данные, хранящиеся в Bigtable, и структурированные входные файлы.
- Масштабируемость: MapReduce может масштабироваться до тысяч процессоров.

III. СРАВНЕНИЕ МЕЖДУ СУБД И MAPREDUCE

До MapReduce параллельные СУБД использовались в качестве подходов для крупномасштабного анализа данных. По сути, все задачи MapReduce могут быть записаны как эквивалентные задачи СУБД. Но имея свои достоинства и недостатки, MapReduce дополняет СУБД, а не конкурирующую технологию. Целью СУБД является эффективность, а MapReduce - масштабируемость и отказоустойчивость. Обе системы явно улучшаются за счет привлечения силы партнера. Существенные различия между СУБД и MapReduce перечислены в таблице 1. Сравнение также выявляет некоторые противоречивые недостатки MapReduce. Далее описаны два основных.

Таблица 1 – Сравнение Параллельной СУБД и MapReduce

	Параллельная СУБД	MapReduce
Схема	Есть	Не поддерживается
Индексация	Есть	Не поддерживается
Программирование (Язык)	Декларативный (SQL)	Императивные (C++, Python, ...)
Оптимизация	Компрессия, хранение столбцов	Не поддерживается
Предварительный разбор	Есть	Не поддерживается
Гибкость	Не поддерживается	Есть
Отказоустойчивость	Уровень транзакций	Уровень задач
Обновления	Чтение и запись многократно	Однократная запись, многократное чтение
Интеграция	Высокая	Низкая
Доступ	Интерактивный и пакетный	Пакетный

– Неэффективность: Сравнивая результаты производительности, время загрузки MapReduce быстрее, чем СУБД, но медленнее во времени выполнения задач. Более длительное время выполнения MapReduce частично объясняется некоторыми специфическими проблемами реализации MapReduce, такими как стоимость запуска Hadoop [3]. Есть и некоторые модельные причины. Как показано в Таблице 1, СУБД выполняет синтаксический анализ во время загруз-

ки и может также реорганизовать входные данные для определенных оптимизаций, в то время как MapReduce не будет изменять макет данных и должен выполнять весь анализ во время выполнения.

- Возможность повторного использования: MapReduce не использует схему или индексацию. Необходимо анализировать структуру входных файлов или реализовать индексацию для ускорения в программах «Map» и «Reduce». Кроме того, пользователи должны предоставлять реализации для простых и распространенных операций, таких как выбор и проекция. Однако СУБД имеет встроенную схему и индексацию, которая упрощает работу. СУБД также поддерживает множество операторов с более высоким уровнем абстракций.

IV. ЗАКЛЮЧЕНИЕ

В соответствии с преимуществами MapReduce, ситуации, когда MapReduce предпочтительнее СУБД, являются:

- Наборы данных для однократного анализа. Эти наборы данных не стоят усилий по индексации и реорганизации в СУБД. Для сравнения, СУБД больше подходит для задач, требующих повторного анализа.
- Комплексный анализ. В некоторые ситуациях, когда функции «Map» слишком сложны, чтобы их можно было легко выразить в запросе SQL, такие как извлечение исходящих ссылок из коллекций документов HTML и агрегирование по целевому документу [4].
- Быстрый старт анализа. Реализации MapReduce просты в настройке, программировании и запуске.
- Ограниченная бюджетная задача. Большинство реализаций MapReduce являются открытыми, в то время как существуют редкие параллельные СУБД с открытым исходным кодом.

V. СПИСОК ЛИТЕРАТУРЫ

1. <https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf>
2. Уайт Т. Hadoop: Подробное руководство. — СПб.: Питер, 2013. — 672 с.
3. M. Stonebraker et al., "MapReduce and parallel DBMSs: Friends or foes?" Commun. ACM, vol. 53, no. 1, pp. 64–71, 2010
4. J. Dean and S. Ghemawat, "MapReduce: A flexible data processing tool," Commun. ACM, vol. 53, no. 1, pp. 72–77, 2010

Гобрик О.Д., магистрант ФИТиУ БГУИР, oleg.gobrik@gmail.com.

Научный руководитель: Гуринович Алевтина Борисовна, кандидат физ.-мат. наук, доцент, gurinovich@bsuir.by

ПРИМЕНЕНИЕ ОПТИЧЕСКОГО РАСПОЗНАВАНИЯ СИМВОЛОВ В ДОКУМЕНТООБОРОТЕ

В данной работе рассматривается применение технологии оптического распознавания символов в документообороте, типовые проблемы технологии OCR, а так же один из возможных методов постобработки результатов распознавания.

ВВЕДЕНИЕ

Переход от бумажного документооборота к электронному является одним из этапов развития организации. Ввиду того, что электронный документооборот локализован в рамках одной организации или отдела, то возникают ситуации, когда всё-таки приходится работать с документами на бумажных носителях, обработка которых неавтоматизированными средствами занимает продолжительное время. Автоматизировать получение текстовых данных с бумажного носителя позволяет технология оптического распознавания символов (англ. optical character recognition, OCR).

I. ТИПОВЫЕ ПРОБЛЕМЫ, СВЯЗАННЫЕ С РАСПОЗНАВАНИЕМ СИМВОЛОВ

Внедрение оптического распознавания символов в документооборот безусловно является одним из шагов успеха в оптимизации и автоматизации, но за данным шагом скрывается ряд существенных проблем распознавания. Наиболее важные из них:

- разнообразие форм начертания символов;
- искажение изображений символов;
- вариации размеров и масштаба символов.

Каждый отдельный символ может быть написан различными стандартными шрифтами, например (Times, Gothic, Elite, Courier, Orator), а также - множеством нестандартных шрифтов, используемых в различных предметных областях. При этом различные символы могут обладать сходными очертаниями. Например, «U» и «V», «S» и «5», «Z» и «2», «G» и «6» [1].

Искажения цифровых изображений текстовых символов могут быть вызваны:

- шумами печати, в частности, непропечаткой, «слипанием» соседних символов, пятнами и точками вблизи символов и т. п.;
- смещением символов относительно их ожидаемого положения в строке;
- изменением наклона символов;
- искажением формы символа за счет оцифровки изображения с «грубым» дискретом;
- эффектами освещения при сканировании документа.

Готовко Роман Юрьевич, магистрант 1 курса кафедры информационных технологий автоматизированных систем БГУИР, gotovkoromanhzby@gmail.com.

Научный руководитель: Севернёв Александр Михайлович, доцент, к.т.н.

II. ПОСТОБРАБОТКА РЕЗУЛЬТАТОВ РАСПОЗНАВАНИЯ

В системах OCR качество распознавания, получаемое при распознавании отдельных символов, не считается достаточным. В таких системах необходимо использовать также контекстную информацию. Использование контекстной информации позволяет не только находить ошибки, но и исправлять их. Доказано, что словарные методы являются одними из наиболее эффективных при определении и исправлении ошибок классификации отдельных символов.

После распознавания всех символов некоторого слова словарь просматривается в поисках этого слова, с учетом того, что оно, возможно, содержит ошибку. Если слово в словаре отсутствует, считается, что в слове допущена ошибка распознавания. Для исправления ошибки прибегают к замене такого слова на наиболее похожее слово из словаря. Исправление не производится, если в словаре найдено несколько подходящих кандидатур для замены. В этом случае интерфейс некоторых систем позволяет показать слово пользователю и предложить различные варианты решения, например, исправить ошибку, игнорировать ее и продолжать работу или внести это слово в словарь. Главный недостаток в использовании словаря заключается в том, что операции поиска и сравнения, применяющиеся для исправления ошибок, требуют значительных вычислительных затрат, возрастающих с увеличением объема словаря.

III. ВЫВОДЫ

Технология оптического распознавания символов позволяет сократить временные и трудовые издержки при переносе текста с бумажного носителя на электронный, что является одним из ключевых признаков для внедрения в документооборот организации.

1. Оптическое распознавание символов (OCR) [Электронный ресурс]. – Режим доступа : <http://wiki.technicalvision.ru>

ИССЛЕДОВАНИЕ ЧАСТЫХ ОШИБОК ПРИ ХРАНЕНИИ ДАННЫХ В МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ

Рассматриваются существующие проблемы в области хранения данных в долговременной памяти при разработке мобильных приложений.

ВВЕДЕНИЕ

В современных реалиях, мобильные устройства занимают важную роль в жизни человека. Однако очень редко разработчики мобильных приложений обеспечивают должный уровень защиты данных пользователя. Одной из причин этого явления служит отсутствие стандартизированного подхода к решению проблемы.

I. ХРАНЕНИЕ ДАННЫХ В ОТКРЫТОМ ВИДЕ

Разработчики зачастую склонны сохранять информацию в защищенные системные хранилища и даже в файловую систему без дополнительной защиты, поскольку системные механизмы хорошо сопротивляются взлому. Однако уровень их стойкости падает до минимума в случае, если устройство рутованное. Если злоумышленник физически достигает мобильного устройства, злоумышленник подключает мобильное устройство к компьютеру с помощью свободно доступного программного обеспечения. Эти инструменты позволяют злоумышленнику видеть все сторонние каталоги приложений, которые часто содержат хранимую личную информацию или другие конфиденциальные информационные активы.

Уязвимость также касается хранения секретных данных внутри кода (в статических константных строках, в ресурсах приложения и т.п.). Яркие примеры: хранение соли для пароля (password salt) в константе или макросе, которая применяется по всему коду для шифрования паролей; хранение приватного ключа для асимметричных алгоритмов; хранение паролей и логинов для серверных узлов или баз данных. Легко вскрывается третьей стороной при наличии базовых навыков декомпиляции.

Защита: Пользовательские данные не должны использоваться в приложении без дополнительного шифрования. Как только необходимость в "открытой"; информации отпала — она немедленно должна быть либо зашифрована, либо уничтожена. Любые данные перед выходом за пределы приложения должны быть зашифрованы. Локальные хранилища платформы не явля-

ются областью приложения, они тоже должны получать на вход только зашифрованные данные.

II. ПРИМЕНЕНИЕ АЛГОРИТМОВ С ХРАНЕНИЕМ ПРИВАТНОГО КЛЮЧА

Уязвимость актуальна в случае, когда приватный ключ вынужденно сохраняется в коде или ресурсах мобильного приложения. Легко вскрывается с помощью реверс-инжиниринга.

Защита: В мобильной разработке желательно применять только современные симметричные алгоритмы с генерируемым случайным одноразовым ключом, обладающие высокой стойкостью с взлому методом грубой силы, либо выводить асимметричный приватный ключ за пределы приложения, либо персонализировать этот ключ (приватным ключом может выступать пользовательский код входа, сохраненный в зашифрованном виде или не сохраняемые вообще).

III. ВЫВОДЫ

Часто приложения позиционируют себя как защищенные, но на деле таковыми не являются, так как содержат внутри себя средства для расшифровки персональной информации.

Хорошим примером реализации безопасного хранения данных является запрашивание у пользователя PIN-кода, затем использование PIN-кода как связанных данных (associated data) с использованием AEAD-режима блочного шифрования. Таким образом, возможность или невозможность расшифровки пользовательских данных будут служить индикатором правильности PIN-кода. При этом PIN-код вообще не сохраняется на устройстве, но операции шифрования основаны на PIN-коде.

1. Drake, J. J. Android Hacker's Handbook / J. Drake, P. For, Z. Lanier, C. Mulliner, S. Ridley, G. Wicherski // John Wiley & Sons, Inc., Indianapolis, Indiana. – 2014. – 545 с.
2. Elenkov, N. Android Security Internals / N. Elenkov // No Starch Press, Inc., San Francisco, CA. – 2015. – 433 с.

Глушень Руслан Русланович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, ruslan.hlushen@gmail.com.

Научный руководитель: Матвеевко Владимир Владимирович, кандидат технических наук, доцент кафедры вычислительных методов и программирования, vladimir66@bsuir.by.

АНОНИМНОСТЬ РАДИ БЕЗОПАСНОСТИ ПРИ ИСПОЛЬЗОВАНИИ АНОНИМАЙЗЕРОВ

Рассматриваются анонимайзеры, особенности передачи информации и уязвимости браузеров.

ВВЕДЕНИЕ

Помимо провайдеров, обладающих информацией о всех действиях пользователей их сети, информацию посещений также сохраняют открытые пользователем страницы во время каждой интернет-сессии. Для того чтобы загрузить ресурс необходимо передавать информацию по протоколам передачи информации, обязательное требование которых - знать адреса отправителя и получателя. Таким образом, чтобы скрыть адрес пользователя, который запрашивает или отправляет данные, нужно найти способ подмены настоящего адреса. Но сначала предстоит убедиться, что браузер не сохраняет лишние данные, которые могут раскрыть настоящий адрес.

I. ОТПЕЧАТКИ БРАУЗЕРА

Для обеспечения быстрого и удобного доступа к интернет-ресурсам, разработчики пишут код, который опирается на сохранённые в браузере ранее данные сайта или пользователя. Все HTTP-запросы сначала направляются браузером в его кеш, чтобы проверить наличие действительного сохранённого ответа. Если совпадение найдено, ответ считывается из кеша, уменьшая время загрузки сайта и объем скачиваемых данных. Например для того, чтобы определить, надо ли загружать статичный контент, в общении браузера с HTTP-сервером используется ETag. Это контрольная сумма, которая должна меняться вместе с изменением файла. При первой загрузке файла браузер получает ETag. Далее браузер передаёт ETag серверу, и тот проверяет, не поменялась ли контрольная сумма, и следовательно — файл. Таким образом, вместо использования по назначению, веб-сервер может использовать этот ETag как идентификатор пользователя. Чтобы остановить кеширование необходимо в настройках браузера отключить соответствующий флажок и очистить кеш браузера. Также особенно часто используются - Cookies, текстовые файлы с какими-либо значениями, хранимые браузером для разных задач, например, аутентификации. К примеру, если пользователь сначала посетил ресурс из открытого сеанса, браузер сохранил cookies, а потом пользователь соединился из ано-

нимного сеанса, тогда сервер может сопоставить cookies и вычислить пользователя. Также существуют 3rd-party cookies, которые сохраняются после просмотра рекламного баннера с другого сайта (3rd-party). И сайт-владелец этого баннера способен отслеживать пользователя на всех ресурсах, где размещены его баннеры. Cookies также можно запретить в настройках браузера. На сайтах могут встретиться такие плагины, как Flash, Java, Adobe. Эти плагины являются отдельными приложениями, которые запускаются от имени пользователя. Они могут обходить настройки прокси, хранить свои отдельные долгоживущие cookies. Для каждого запуска такого плагина нужно подтверждение пользователя, поэтому следует запрещать запуск в появившемся окне. Также, браузер предоставляет серверу user agent (полная информация о приложении и устройстве). Необходимо изменить в настройках браузера user agent на другое значение.

II. ПОДМЕНА АДРЕСА С ПОМОЩЬЮ АНОНИМАЙЗЕРОВ

По причине отсутствия шифрования в прокси-серверах, не стоит рассматривать их как анонимайзеры. Рассмотрим VPN. При подключении к VPN пользователь подключается к другой сети, а она обеспечивает зашифрованное соединение с нужным сервером напрямую. Внутри этой сети существуют свои серверы и выходной узел. Но владельцы VPN могут хранить или продавать данные о пользователях. Рассмотрим TOR. TOR — это система маршрутизаторов, в которой пользователь соединяется с Интернетом через цепочку узлов. Цепочка состоит из трех узлов, каждому из них неизвестны адреса клиента и ресурса одновременно. TOR шифрует сообщения отдельно для каждого узла, а открытый трафик виден только выходному роутеру. Но уже существуют способы перехвата данных направленных к цепочке TOR. Для обеспечения анонимности самым лучшим решением будет включение VPN, а затем TOR. Сторонним лицам виден лишь двойной слой шифрования данных (один от TOR, другой от VPN), а дальше трафик проходит по зашифрованной цепочке TOR до адреса назначения. В такой системе трафик нельзя перехватить.

Гурман Артём Витальевич, студент кафедры информационных технологий автоматизированных систем БГУИР, goodjman@outlook.com.

Научный руководитель: Ярмолик Валерий Иванович, ассистент кафедры информационных технологий автоматизированных систем БГУИР, v.jarmolik@bsuir.by.

ИССЛЕДОВАНИЕ ПРОБЛЕМ И МЕТОДОВ УПРАВЛЕНИЯ ПРОЦЕССАМИ ЛЕСОВОССТАНОВЛЕНИЯ

Рассматриваются проблемы и методы управления процессами лесовосстановления.

ВВЕДЕНИЕ

Лесовосстановление - область лесного хозяйства, в которой информационные технологии наиболее востребованы. Это связано с большими объемами информации, которые используются в работе. Применение современных информационных технологий позволяет уменьшить объем выполняемых работ по управлению процессами лесовосстановления, улучшить качество и сократить время их выполнения.

I. СУЩЕСТВУЮЩИЕ МЕТОДЫ УПРАВЛЕНИЯ

Основой принятия решений в области управления лесовосстановлением, более эффективного учета лесных культур является информация о состоянии участков лесовосстановления, получаемая путем выполнения взаимосвязанной группы технологических процессов: определение участков для обязательного восстановления, оценка защитных свойств лесов, их продуктивности и устойчивости, определение необходимых операций по уходу и др. В настоящее время в лесном хозяйстве РБ вся информация о результатах выполнения вышеперечисленных технологических процессов хранится в базах данных различных автоматизированных систем, таких как:

- АРМ "Лесопользование";
- геоинформационная система "Гис.Лес";
- автоматизированная система электронного учёта древесины (АСЭУД) и др.

Для эффективного управления также необходимо оценить экономические затраты, спрогнозировать временной промежуток восстановления лесных участков и процент приживаемости лесных культур.

При таком подходе, лицу, принимающему решения в области управления процессами лесовосстановления, следует собрать необходимую информацию из разных источников, проанализировать и систематизировать показатели вручную. Такой способ значительно замедляет и

усложняет управление лесовосстановительными процессами.

II. АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ ПРОЦЕССАМИ ЛЕСОВОССТАНОВЛЕНИЯ

Данные, полученные в результате выполнения вышеперечисленных технологических процессов лесовосстановления, предлагается хранить в единой информационной системе управления.

Для прогнозирования восстановления лесных участков использовать регрессионный анализ. Этот метод предполагает исследование влияния различных факторов (параметры окружающей среды, качество посадочного материала, процент приживаемости на похожих участках и др.) на показатели восстановления лесных культур.

Экономический анализ осуществлять с помощью балансового метода. Данный метод предполагает разработку системы показателей, в которой одна часть, характеризующая лесные ресурсы, равна другой части, показывающей использование ресурсов по всем направлениям.

III. ВЫВОДЫ

Изучив и проанализировав существующие методы и проблемы управления процессами, предлагается использование единой информационной системы управления, реализующей данные методы. Такая система должна хранить все показатели необходимые для управления лесовосстановлением, систематизировать данные, синхронизироваться с другими программными средствами, осуществлять экономический анализ и прогнозирование восстановления участков.

Список литературы

1. Черных, В. Л. Проблемы внедрения новых информационных технологий в лесную отрасль / В. Л. Черных // Лесн. хоз-во. – 2001. – 159 с.
2. Мухин, В. И. Исследование систем управления / В. И. Мухин // Издательство: "Экзамен 2003. – 384 с.

Дробышев Сергей Владимирович, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, drobyshvsergei96@gmail.com.

Научный руководитель: Кукин Дмитрий Петрович, заведующий кафедры вычислительных методов и программирования Белорусского государственного университета, кандидат технических наук, kukin@bsuir.by.

АЛГОРИТМ АВТОМАТИЗАЦИИ СОСТАВЛЕНИЯ ЗАКАЗА

Рассматривается реализация алгоритма автоматизации составления заказа. Используется алгоритм «укладки рюкзака» как решение задачи.

ВВЕДЕНИЕ

Для реализации алгоритма автоматизации составления заказа используется алгоритм «укладки рюкзака». Задача автоматизации заказа заключалась в том, что необходимо наиболее оптимально подобрать блюда к введенным критериям. Для чего полностью подходит алгоритм «укладки рюкзака».

I. АЛГОРИТМ «УКЛАДКИ РЮКЗАКА»

Алгоритм «укладки рюкзака» своё название получил от конечной цели: уложить как можно большее число ценных вещей в рюкзак при условии, что вместимость рюкзака ограничена.

Само собой разумеется, что к данной классической формулировке задачи могут сводиться многие другие задачи разных размерностей. В качестве цены и веса могут использоваться совершенно различные характеристики и даже их комбинации.

Сама задача является NP-полной задачей, то есть такой, время работы алгоритма для решения которой существенно зависит от размера входных данных, при этом если предоставить алгоритму некоторые дополнительные сведения, то он сможет за время, не превосходящее некоторого многочлена от размера входных данных решить задачу.

В свою очередь это означает, что задачу можно решить при помощи динамического программирования.

Итак, формулировка задачи о ранце с возможностью неограниченного выбора звучит так. По заданному набору из n предметов со стоимостями v_1, v_2, \dots, v_n и весами w_1, w_2, \dots, w_n необходимо найти такой поднабор, что его стоимость будет максимальна среди всех поднаборов с общим весом не более W .

Необходимо учесть тот факт, что каждый предмет имеется в единственном экземпляре, что

является наиболее подходящим для решения поставленной задачи в проекте, так как блюда должны быть подобраны из разных категорий по одному разу.

Тогда $K[i, v]$ – максимальная стоимость предметов, полученных из первых i имеющихся предметов, с суммарным весом не превышающим w .

Рекуррентные соотношения:

$$K[0, i] = 0, 0 \leq i \leq n \quad (1)$$

$$K[w, 0] = 0, 0 \leq w \leq W \quad (2)$$

$$K[w, i] = \max(K[w, i-1], K[w-w_i, i-1] + v_i), \\ w_i \leq w, 0 \leq w \leq W \quad (3)$$

$$K[w, i] = K[w, i-1], w_i > w \quad (4)$$

II. АДАПТАЦИЯ АЛГОРИТМА

Для адаптации задачи для автоматизированного составления заказа, изменим вес на вводимые параметры: жиры, белки, углеводы. Стоимость остается такая же. Тогда в формулах 1 – 4 w будет являться параметром, по которому идет отбор блюд, W – вводимый параметр.

Для составления заказа без учета цены или с наименьшей ценой данный алгоритм модифицируется, так как выше приведена только классическая формулировка задачи.

III. ВЫВОДЫ

Алгоритм «укладки рюкзака» решает задачу автоматизации составления заказа. Данный алгоритм может модифицироваться для разных случаев задачи.

1. Беллман, Р. Прикладные задачи динамического программирования / Р. Беллман, С. Дрейфус // Наука. – 1965. – С. 460.

Научный руководитель: Курулев Александр Петрович, профессор, кандидат технических наук.

ТЕХНОЛОГИЯ ГЕНЕРАТИВНЫХ НЕЙРОННЫХ СЕТЕЙ В ПРЕОБРАЗОВАНИИ ИЗОБРАЖЕНИЙ

В работе приводится описание технологии генеративных нейронных сетей и их применение в преобразовании изображений.

ВВЕДЕНИЕ

Потенциал генеративных нейронных сетей огромен, поскольку они имитируют любое распределение данных. Генеративные нейронные сети обучают создавать структуры, устрашающе похожие на существа из нашего мира в области изображений, музыки, речи, прозы. Автоматическое редактирование – этот подход уже используется в современных смартфонах и некоторых программах. Благодаря работе генеративных нейронных сетей возникает синтез данных, на которых потом будут обучаться другие системы.

I. ТЕХНОЛОГИЯ ГЕНЕРАТИВНЫХ НЕЙРОННЫХ СЕТЕЙ В ПРЕОБРАЗОВАНИИ ИЗОБРАЖЕНИЙ

Генеративно-состязательная нейронная сеть (англ. Generative adversarial network, GAN) – один из алгоритмов классического машинного обучения, обучения без учителя. Суть идеи в комбинации двух нейронных сетей, при которой одновременно работают два алгоритма: «генератор» и «дискриминатор».

Дискриминатор пытается распознать созданный образ. Дискриминационные алгоритмы пытаются классифицировать входные данные. Учитывая особенности полученных данных, они стараются определить категорию, к которой они относятся. Для распознавания используются сверточные нейронные сети (CNN). Для того, чтобы нейронная сеть научилась что-то распознавать, ей нужно обработать большое количе-

ство изображений, где содержатся искомые образы.

Задача генератора – генерировать образы заданной категории. Генеративные алгоритмы заняты обратным. Вместо того, чтобы предсказывать категорию по имеющимся образам, они пытаются подобрать образы к данной категории. Формирование изображений начинается с генерации произвольного шума, на котором постепенно начинают проступать фрагменты искомого изображения. Архитектура GAN представлена на рисунке 1.

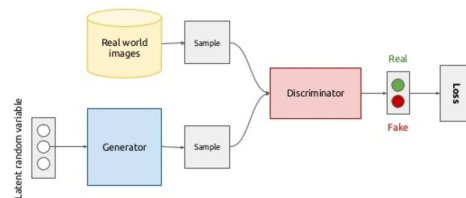


Рис. 1 – Архитектура GAN

ЗАКЛЮЧЕНИЕ

Разработанная технология генеративной нейронной сети позволяет трансформировать один вид изображений в другой.

Список литературы

1. Николенко С., Глубокое обучение / А. Кадурич, Е. Архангельская. – Питер, 2020. – 480 с.
2. Гудфеллоу Я., Глубокое обучение / И. Бенджио, А. Курвилль. – ДМК, 2017. – 652 с.
3. Патанаяк С., Глубокое обучение и TensorFlow для профессионалов. – Вильямс, 2019. – 480 с.

Завистович Игорь Геннадьевич, магистрант 1 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, igor.zavistovich.98@mail.ru

Чернаштан Дмитрий Николаевич, магистрант 1 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, dchernashtan@mail.ru

Научный руководитель: Герман Олег Витольдович, доцент кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, кандидат физико-математических наук, ovgerman@tut.by

ВЕБ-ПРИЛОЖЕНИЕ ВИДЕОМОНИТОРИНГА

В настоящее время системы видеонаблюдения установлены во многих общественных местах. Отличие видеомониторинга от системы видеонаблюдения заключается в том, что он позволяет следить за интересующим объектом в режиме реального времени из любой точки мира. Возможность просматривать онлайн-поток с помощью любого браузера через компьютер, мобильный телефон или планшет.

ВВЕДЕНИЕ

Данная статья посвящена реализации получения видеопотока с камеры в режиме реального времени. Было предложена следующая схема (см.рис.1.).

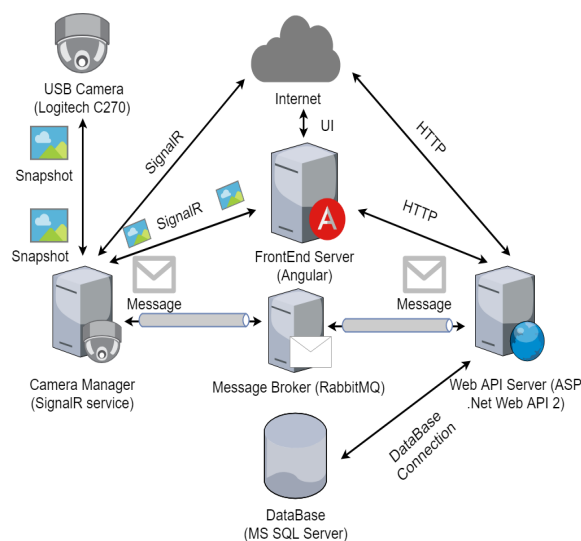


Рис. 1 – Схема веб-приложения видеомониторинга

I. ФУНКЦИОНИРОВАНИЕ СИСТЕМЫ

Последовательность работы системы состоит из следующих шагов:

1. Клиент создает модель камеры обращаясь к Web API Server через Angular Frontend Server по HTTP протоколу [1];
2. Модель сохраняется в Data Base;
3. При создании камеры публикуется Message в Even Bus о создании камеры;
4. Web Socket Server получает Message из Event Bus и подключается к USB Camera;
5. При подключении к USB Camera он подписывается на получение кадров изображений (snapshot) для этой камеры;
6. После получения кадра изображения. Кадр кодируется в base64. И передается с помощью SignalR для клиентов, которые подписались на получение этого сообщения [2];

Зенович-Лешкевич-Ольпинский Александр Юрьевич, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, alexander.zenovich@gmail.com.

Научный руководитель: Никольшин Борис Викторович, заведующий кафедры электронных вычислительных машин Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, доцент, nik@bsuir.by.

7. Клиент переходит на детальную страницу для USB Camera. И подписывается на получение кадров изображений для определенной камеры;
8. После подписки на определенную камеру. Сервер начинает слать кадры для клиента.

II. ПРЕИМУЩЕСТВА ПОДХОДА РАЗРАБОТКИ

Основные преимущества данного подхода к разработке:

- Возможность поддержки одной и более веб-камеры;
- Доступность из глобальной сети;
- Адаптивный интерфейс под различные разрешения. Поддержка мобильных устройств, компьютеров и планшетов;
- Расширяемость архитектуры;
- Отделение взаимодействия сервера с соединениям камер (Camera Manager) и сервера с хранением и управлением всех камер (Web API Server);
- Автоматизация процессов с камерой. Операции удаления, добавления, просмотр изображения и редактирования камеры.

III. ВЫВОДЫ

При старте проекта были решены две главные проблемы. Первая проблема – возможность просмотра потока видеоизображений в режиме реального времени. И вторая проблема – автоматизация процессов с видеокamerой. В данном проекте был разработан модуль видеомониторинга на базе платформы .Net Framework и Angular. Данное программное средство используется в сфере телекоммуникаций для визуального контроля за объектом.

Список литературы

1. Сетевое программирование: протокол HTTP [Электронный ресурс]. – Режим доступа: https://professorweb.ru/my/csharp/web/level7/7_1.php. – Дата доступа: 02.02.2020
2. Руководство по SignalR [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/signalr>. – Дата доступа: 03.02.2020.

МЕТОДЫ И АЛГОРИТМЫ ФОРМИРОВАНИЯ ПЕРСОНАЛА КОМПАНИИ

Рассматривается реализация программного средства, которое позволяет автоматизировать управление подбором персонала компании

ВВЕДЕНИЕ

В настоящее время человеческим ресурсам уделяется все большее внимание. Если ранее служба персонала была представлена отделом кадров, то в настоящее время кадровая работа направлена на формирование работоспособного и эффективно функционирующего персонала. Система автоматизации позволяет сократить временные затраты, связанные с процессом поиска и найма сотрудников.

I. ПРОЦЕСС ПОДБОРА ПЕРСОНАЛА

Первый этап – формирование заявки руководителем, заполняется форма самой заявки и отправляется рекрутерам. На втором этапе формируется вакансия из заявки, рекрутеры создают вакансию. Далее она отправляется на согласование. Перед отправкой на согласование рекрутеру требуется сформировать маршрут согласования вакансии. Процесс из четырех основных шагов. На первом этапе согласования выборе этого шага вакансия отправляется на согласование автору вакансии. Автор может согласовать или не согласовать вакансию или отправить вакансию на доработку. На втором этапе согласования вакансия отправляется на рассмотрение параллельно всем сотрудникам из списка. Каждый из участвующих в рассмотрении может должен свое мнение о вакансии в комментарии и поставить резолюцию «Рассмотрел». На третьем этапе указывается список согласующих вакансию. Согласование на этом шаге происходит последовательно. На четвертом этапе вакансия отправляется на ознакомление параллельно каждому пользователю из списка. Изменение статуса вакансии на каждом шагу фиксируется и заносится в базу в историю согласования вакансии. После этапа формирования вакансии следует этап анализа и подбора кандидатов. Сначала данные о вакансии переходят на шаг анализа предложенных системой кандидатов, где на основе компетенций система подбирает наиболее подходящих кандидатов. Далее рекрутеры сравнивают отобранных системой кандидатов и формируют шортлист.

Зинченко Владислав Андреевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, zinchenko.vladislav.1995@gmail.com.

Научный руководитель: Никульшин Борис Викторович, заведующий кафедрой ЭВМ доцент, кандидат технических наук, nik@bsuir.by

II. РАЗРАБОТКА СИСТЕМЫ

Исходя из спецификации требований, разработано программное средство. Программное средство состоит из четырех модулей: «Заявки», «Вакансии», «Кандидаты» и «Календарь интервью», каждый из которых представляет собой портлет, который является отдельным визуальным элементом портала.

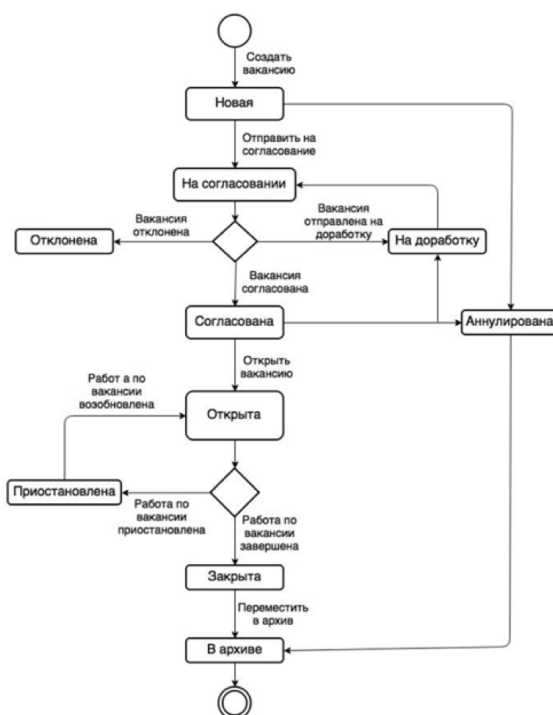


Рис. 1 – Процесс согласования вакансии

III. ВЫВОДЫ

Предлагаемая мною система с методами и алгоритмами формирования персонала позволяет создавать заявки и подбирать кандидатов на вакансии.

1. Базаров, Т. Ю. Управление персоналом // М: ЮНИТИ. – 2005. – С.342.

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

В работе рассматривается подход к моделированию оптимизационных задач линейного программирования. Для данной цели авторами статьи применено программное обеспечение, использующее программную библиотеку для построения графиков.

ВВЕДЕНИЕ

В процессе исследования и создания сложных систем важное место занимает компьютерное моделирование. Компьютерное моделирование заключается в создании компьютерной программы (пакета программ), описывающей поведение элементов исследуемой системы в процессе ее функционирования, учитывающей их взаимодействие между собой и внешней средой, и проведении на ЭВМ серии вычислительных экспериментов. Это делается с целью изучения природы и поведения объекта, его оптимизации и структурного развития, прогнозирования новых явлений. Реализация моделирования осуществлена с использованием платформы для разработки веб-приложений Angular и программной библиотеки для построения графиков Highcharts.

К числу задач линейного программирования можно отнести задачи управления производственными запасами, рационального использования сырья и материалов, планирование производства и т.п.

1. РЕАЛИЗАЦИЯ АЛГОРИТМА

Главной задачей компьютерного моделирования является выбор и построение информационной модели объекта. Существует несколько типов моделей: 1) математические модели; 2) имитационные модели; 3) графические модели.

Рассматривается графическая динамическая модель оптимизационных задач линейного программирования, где показатель эффективности W зависит только от двух групп параметров: заданных условий a и элементов решений x , т.е. $W=W(a, x)$.

Характерно для них то, что: а) показатель эффективности (целевая функция) W линейно зависит от элементов решения x_1, x_2, \dots, x_n и б) ограничения, налагаемые на элементы решения, имеют вид линейных равенств или неравенств относительно x_1, x_2, \dots, x_n . Возможности удобного и наглядного графического метода решения задач линейного программирования ограничены случаем двух переменных x_1 и x_2 . [1]

Чтобы найти оптимальное решение, следует перемещать прямую, характеризующую показатель эффективности, в направлении возрастания или убывания целевой функции до тех пор, пока она не сместится в область недопустимых решений. Результаты, которые были получены при занесении различных коэффициентов в математическую модель, наглядно показывают на графике (рис.1), что оптимальному значению всегда может быть подставлена в соответствие одна из допустимых угловых точек пространства решений.

После нахождения оптимального решения, логично выяснить, как отразятся на оптимальном решении изменение запасов ресурса: на сколько можно снизить запас некоторого ресурса при сохранении полученного оптимального значения целевой функции и на сколько можно увеличить запас ресурса для улучшения оптимального значения целевой функции.[2]

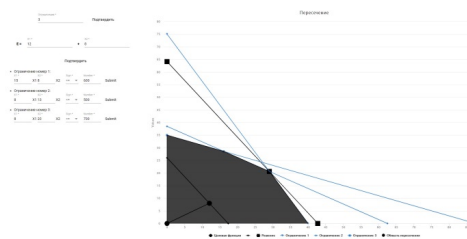


Рис. 1 – Пример оптимизационной задачи

ЗАКЛЮЧЕНИЕ

Предлагаемое программное средство поможет наглядно понимать процессы в оптимизационных задачах линейного программирования. Предлагается использовать в помощь студентам и преподавателям при изучении дисциплин, связанных решением оптимизационных задач.

1. Вентцель Е.С. Исследование операций: задачи, принципы, методология. – 2-е изд., стер. – М.: Наука. Гл.ред.физ.-мат.лит., 1988.-208 с.
2. Таха Х. Введение в исследование операций: В 2-х книгах. Кн. 1. Пер. с англ. – М.: Мир, 1985. – 479 с.

Ивановский Сергей Николаевич, студент 2 курса ФИТиУ БГУИР, serheoddd@yandex.by

Климов Глеб Сергеевич, студент 2 курса ФИТиУ. klimovgleb@mail.ru

Научный руководитель: Протченко Екатерина Владимировна, старший преподаватель каф. ИТАС, protchenko@bsuir.by

WEB-ПРИЛОЖЕНИЕ ТУРИСТИЧЕСКОГО АГЕНСТВА

Рассматривается подход к созданию web-приложения, содержащего клиентскую и серверную части. Приводятся наиболее популярные технологии решения проблем реализации приложения.

ВВЕДЕНИЕ

Для создания клиент-серверного приложения требуется понимание целого набора инструментов, языков и систем. Кроме знания HTML/CSS и JavaScript, необходимо освоить языки и инструменты для реализации серверной части, которая осуществляет взаимодействие с базой данных. Сложность создания приложения усиливается постоянным развитием и изменением технологий.

В статье представлен набор средств, которые активно используются разработчиками и являются одними из самых актуальных на данный момент.

I. КЛИЕНТСКАЯ ЧАСТЬ

Для реализации клиентской части можно использовать язык JavaScript, он гибок и прост в использовании. Но по мере усложнения JS-кода возрастает преимущество использования языка TypeScript. TypeScript – это язык-расширение JavaScript, которое добавляет возможность статической типизации. Когда TS-код компилируется в JS, все определения типов опускаются. TypeScript позволяет проще писать комплексные решения, которые легче развивать и тестировать в дальнейшем. Однако важно учитывать тот факт, что разработка на TypeScript займет больше времени.

Существуют два основных подхода к разработке клиента: можно создавать как одностраничные веб-приложения (SPA), так и многостраничные (MPA). SPA-приложения работают в рамках браузера и не требуют перезагрузки страницы или загрузки дополнительных страниц во время использования. Самыми популярными примерами SPA-приложений являются GitHub, Gmail, Google Maps и Facebook. В SPA есть всего одна страница с разметкой, стилями и скриптами. При переходе на другие страницы, приложение подгружает соответствующие модули и контент. Кроме вышеперечисленного, SPA-приложения имеют, как правило, уже готовые удобные решения для организации REST API, управления отображением HTML-кода и множество других полезных при разработке средств. Так, на страничке приложения турагенства отображается список туров, предлагаемых пользователю сайта в виде блоков с информацией о туре. Блок HTML-кода создается только один, а данные в нем зависят от переданных значений. С помощью специальных инструкций, реализующих

циклы, на странице отображается нужное количество блоков с различными данными, соответствующими информации о каждом туре.

При таком архитектурном подходе чаще всего используются технологии React, Angular и Vue. Они являются «тремя китами» в разработке клиентской части приложения.



Рис. 1 – Популярные JavaScript-фреймворки

У каждого фреймворка есть свои плюсы и минусы, а это означает, что при разработке продукта нужно сделать правильный выбор для каждого отдельного случая.

II. СЕРВЕРНАЯ ЧАСТЬ

Сервер организует взаимодействие как с клиентской частью, так и базой данных. Здесь создаются API для фронтенда. Для реализации серверной части можно использовать языки Ruby и популярные фреймворки Ruby on Rails и Sinatra, Python и фреймворки Django и Flask или же языки Java, PHP. Однако, с опытом работы с JavaScript наилучшим решением станет Node.js. Node.js – это просто окружение JS, то есть, при знании JavaScript, нет необходимости в изучении нового языка программирования. Функциональность Node.js расширяется с помощью пакетов. Наиболее часто используется фреймворк Express, все альтернативы в десятки раз менее популярны. Он существенно упрощает разработку веб-приложений на базе Node.js.

III. ВЗАМОДЕЙСТВИЕ КЛИЕНТА И СЕРВЕРА

Итак, определившись с технологиями для создания приложений клиента и сервера, возникает вопрос их взаимодействия. REST API – это «мостик» между двумя этими сторонами. REST – это набор принципов и ограничений взаимодействия клиента и сервера в сети интернет, использующий существующие стандарты (HTTP протокол, стандарт построения URL, форматы данных JSON и XML) в ходе взаимодействия. Так, клиентской части предоставляется некий URL, определенный на серверной части, по которому

клиент получает необходимую информацию из базы данных.

Например, при загрузке страницы туров, с клиента, используя URL, отправляется запрос на сервер для предоставления информации о каждом туре.

IV. БАЗА ДАННЫХ

При выборе базы данных предстоит принять важное решение: остановиться на реляционной (SQL) или нереляционной (NoSQL) структуре БД. MySQL – отличный выбор для любого приложения, которому будет удобно пользоваться ее заранее определенной структурой и готовыми схемами. MongoDB, напротив, подойдет для бизнесов с быстрым ростом или для баз данных, в которых не используются определенные схемы. Для работы с MongoDB используется специальная ODM-библиотека (Object Data Modelling) Mongoose, которая позволяет сопоставлять объекты классов и документы коллекций из базы данных.

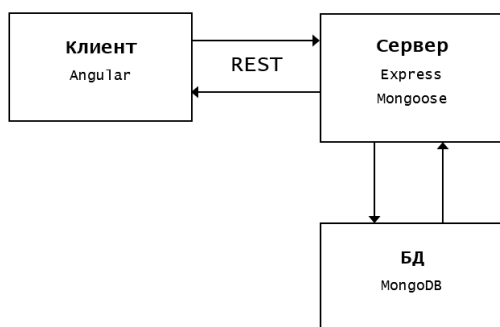


Рис. 2 – Архитектура клиент-серверного приложения

V. АУТЕНТИФИКАЦИЯ И АВТОРИЗАЦИЯ

Когда в системе создаётся новый пользователь, его пароль необходимо хэшировать и сохранять в базе данных. Пароль в базе сохраняют вместе с адресом электронной почты и другими сведениями о пользователе.

Когда же клиент входит в свою учетную запись, он отправляет серверу аутентификационный ключ. Обычно это – пара логин/пароль. Если пользователь существует в базе данных – сервер хэширует отправленный ему пароль и сравнивает то, что получилось, с хэшем пароля, сохранённым в базе данных.

Карчевская Яна Геннадьевна, студентка кафедры информационных технологий автоматизированных систем БГУИР, уапа-03032017@mail.ru.

Научный руководитель: Трофимович Алексей Фёдорович, старший преподаватель БГУИР, зам. декана ФИТиУ, trofimovich_a_f@tut.by.

Использование JSON Web Token (JWT) – JSON объекта, определенного в открытом стандарте RFC 7519 – считается одним из безопасных способов передачи информации между двумя участниками. Сервер будет обеспечивать пользователя токеном, с помощью которого он позднее сможет взаимодействовать с приложением. Когда пользователь сделает запрос к API приложения, к нему добавится полученный ранее JWT для проверки того, что данные были действительно отправлены авторизованным источником.

Так, пользователь, имеющий учетную запись, сможет аутентифицироваться при каждом запросе, например, для получения данных о своих заказах.

VI. ВЫВОДЫ

Архитектура клиент-серверного приложения предполагает разделение системы на клиентов и на серверов. Разделение интерфейсов означает, что, например, клиенты не связаны с хранением данных, которое остается внутри каждого сервера, так что мобильность кода клиента улучшается. Серверы не связаны с интерфейсом пользователя или состоянием, так что серверы могут быть проще и масштабируемы. Серверы и клиенты могут быть заменяемы и разрабатываться независимо, пока интерфейс не изменяется.

Существует множество инструментов для реализации данной архитектуры приложения. Использование технологии SPA для создание клиентской части и Node.js для серверной является актуальным подходом для простой и удобной разработки клиент-серверных приложений.

1. Express: fast, unopinionated, minimalist web framework for Node.js [Electronic resource] / StrongLoop, IBM, and other expressjs.com contributors, 2017. – Mode of access: <https://expressjs.com/>. – Date of access: 14.04.2020.
2. Node.js Docs [Electronic resource] / Joyent, Inc., 2020. – Mode of access: <https://nodejs.org/en/docs/>. – Date of access: 14.04.2020.
3. single page application (SPA) и multi page application (MPA) [Электронный ресурс] / MIRENEAD: fintech и блокчейн разработка, 2019. – Режим доступа: <https://merehead.com/ru/>. – Дата доступа: 14.04.2020.
4. Как подступиться к fullstack-разработке [Электронный ресурс] / Хабр: сообщество IT-специалистов, 2019. – Режим доступа: <https://habr.com/ru/>. – Дата доступа: 14.04.2020.

ИССЛЕДОВАНИЕ МЕТОДОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА МОБИЛЬНЫХ УСТРОЙСТВАХ

Рассматриваются шумы и основные методы фильтрации изображений

ВВЕДЕНИЕ

В процессе передачи и преобразования посредством радиотехнических систем, изображения подвергаются воздействию различных помех, что в ряде случаев приводит к ухудшению визуального качества и потере участков изображений. С широким внедрением цифровых систем связи, увеличивается актуальность решения задач восстановления изображений полученных с помощью камер, с целью фильтрации изображений.

I. ОСНОВНЫЕ ЗАДАЧИ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Задачей обработки изображения может быть как улучшение (восстановление, реставрация) изображения по какому-то определенному критерию, так и специальное преобразование, кардинально меняющее изображение. В последнем случае обработка изображений может быть промежуточным этапом для дальнейшего распознавания изображения (например, для выделения контура объекта). Чаще всего шумоподавление служит для улучшения визуального восприятия. Также шумоподавление играет важную роль при сжатии изображений. При сжатии сильный шум может быть принят за детали изображения, и это может отрицательно повлиять на результирующее качество сжатого изображения. Источники шума могут быть различными: неидеальное оборудование для захвата изображения (видеокамера, сканер и т.п.); плохие условия съемки (сильные шумы, возникающие при ночной съемке); помехи при передаче по аналоговым каналам – наводки от источников электромагнитных полей, собственные шумы активных компонентов (усилителей) линии передачи.

II. Виды шумов

Наиболее адекватными с точки зрения использования в практических задачах являются модели аддитивного Гауссова и импульсного шума. Аддитивный Гауссов шум характеризуется добавлением к каждому пикселю изображения значений из соответствующего нормального рас-

пределения с нулевым средним значением. Такой шум обычно вводится на этапе формирования цифровых изображений. Импульсный шум характеризуется заменой части пикселей на изображении значениями фиксированной или случайной величины. Такая модель шума связана, например, с ошибками при передаче изображений.

III. Методы удаления шумов

Алгоритмы шумоподавления обычно специализируются на подавлении какого-то конкретного вида шума. Не существует пока универсальных фильтров, детектирующих и подавляющих все виды шумов. Однако многие шумы можно довольно хорошо приблизить моделью белого Гауссова шума, поэтому большинство алгоритмов ориентировано на подавление именно этого вида шума. К самым распространенным методам удаления шумов можно отнести: сглаживающие фильтры; фильтры Винера; медианные фильтры; ранжирующие фильтры.

IV. Выводы

Фильтрация изображений помогает решить множество проблем, возникающих при формировании изображения любым цифровым устройством. При фильтрации яркость каждой точки исходного изображения, искаженного помехой, заменяется некоторым другим значением яркости, которое признается в наименьшей степени искаженным помехой. Размытие фото применяется для придания изображению с ограниченным количеством цветов большего количества оттенков. Как правило, эффект размытия применяется к изображениям и с плавным переходом цветов для увеличения диапазона используемых оттенков. Также этот эффект позволяет расфокусировать необходимые участки на снимке.

1. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс // Техносфера. – 2012 – 1104 с.
2. A. Buades A review of image denoising algorithms, with a new one / A. Buades, B. Coll, J.M. Morel, // SIAM Multiscale Modeling and Simulation, vol. 4 – 2005 – 490 с.

Клебан Евгений Геннадьевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, kleban.eugene@gmail.com.

Научный руководитель: Матвеевко Владимир Владимирович, кандидат технических наук, доцент кафедры вычислительных методов и программирования, vladimir66@bsuir.by.

МЕТОДОЛОГИЯ ДВУМЕРНОЙ ВИЗУАЛИЗАЦИИ РАСПИСАНИЯ УЧЕБНОГО ПРОЦЕССА

Визуализация данных – не только инструмент для наглядного представления закономерностей, но и эффективнейшее средство для понимания абстрактных процессов, широко использующееся во многих сферах деятельности – от научных проектов до повседневной инфографики.

I. ОПРЕДЕЛЕНИЕ ДВУМЕРНОЙ ВИЗУАЛИЗАЦИИ ДАННЫХ

Двумерная визуализация данных – это графическое представление каких-либо данных на плоскости, которое обеспечивает наиболее эффективную работу по их изучению. Является самым распространённым по причине того, что обладает наибольшей наглядностью представления разрозненных сложных данных путем графического изображения на плоскости.

II. СПОСОБЫ ДВУМЕРНОЙ ВИЗУАЛИЗАЦИИ РАСПИСАНИЙ

Существует несколько методов к визуализации расписаний, каждый из которых обладает своими положительными и отрицательными моментами, однако всё же выбор зачастую зависит от сферы применения данной визуализации.

1. Диаграммы и графики, – наверное, самые привычные для нас виды визуализации данных в целом. Используются как для презентации данных, так и для анализа. В применении к расписанию обычно используется для отображения какой-либо статистики и зависит от целей визуализации (например, отображение частоты посещений или выбора самых загруженных дней).
2. Инфографики и схемы. Инфографика относится к журналистике данных, где графики и схемы объясняют какие-либо факты по выбранной теме. Обычно инфографика статична и представляет собой длинную «простыню» с картинками и текстом. В визуализации расписания учебного процесса преимущественно используется при составлении планов на учебный период.
3. Списки и таблицы. Данные способы визуализации расписаний имеют наиболее привычный и распространённый для нас вид. Они используются повсеместно: от отображения следования учебного процесса до ви-

зуализации графика работы. Информация отображается в виде перечня занятий за определённый промежуток времени. Каждый элемент обычно начинается с маркера или номера-буквы, что позволяет упорядочить список или таблицу.

4. Картограмма – карта с нанесенной на неё информацией в виде цвета или других способов. Данный способ визуализации используется реже остальных, однако является самым показательным из этих методов. Примером может служить совокупность картограмм университета, на которых отображено, в каких аудиториях в определённый промежуток времени проводятся занятия с указанием как обучающихся, так и преподавателей.

III. ВЫВОДЫ

Сегодня не существует ни одного метода двумерной визуализации, в том числе учебного процесса, способного в полном объеме удовлетворить потребности использующих их людей. Выбор способа происходит исходя из того, какой из методов будет наиболее актуален и информативен в применении. Таким образом, имеется необходимость в знании достоинств и недостатков методов и их связей с конкретными задачами, что в совокупности позволит выбирать адекватные поставленной задаче методы визуализации.

Список литературы

1. Multidimensional Data Visualization: Methods & Examples [Electronic resource] / D. Gloag. – Study.com, 2015. – Mode of access: <https://study.com/academy/lesson/multidimensional-data-visualization-methods-examples.html>. – Date of access: 10.04.2020.
2. Романова, И. К. Современные методы визуализации многомерных данных: анализ, классификация, реализация, приложения в технических системах / И. К. Романова // Наука и образование. МГТУ им. Н. Э. Баумана. Электрон. журн. – 2016. – № 3. – С. 133–167.

Климович Артём Геннадьевич, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, temaklim96@tut.by.

Научный руководитель: Севернёв Александр Михайлович, доцент кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, ответственный за преддипломную практику, severnev@bsuir.by.

АЛГОРИТМЫ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ В БИЛЛИНГОВЫХ СИСТЕМАХ ТЕЛЕКОММУНИКАЦИОННЫХ КОМПАНИЙ

Рассматривается eTOM модель как средство для анализа и управления бизнес-процессами в BSS/OSS системах в телеком-индустрии, также рассматриваются способы оценки эффективности бизнес-процесса.

ВВЕДЕНИЕ

На сегодняшний день телекоммуникации стали неотъемлемой частью жизни общества, без информационного обмена невозможно представить повседневный быт человека. Провайдером телекоммуникационных услуг являются телекоммуникационные компании, от деятельности которых напрямую зависит скорость и качество информационного обмена. Для того чтобы предлагать клиентам высококачественные товары или услуги, компания должна контролировать внутренние процессы их создания. Продуманные и хорошо отлаженные бизнес-процессы обеспечивают высокий уровень качества, а эффективное управление бизнес-процессами позволяет снизить издержки и помогает компании адаптироваться к быстро меняющимся условиям телеком бизнеса.

I. OSS/BSS СИСТЕМЫ В ТЕЛЕКОМЕ

Решения класса OSS/BSS (Operation Support System/Business Support System) отвечают за две стороны работы телекоммуникационной компании: управление инфраструктурой и ресурсами, а также взаимодействие с абонентами. То есть основная функция таких решений, работающих в комплексе, заключается в том, чтобы услуги предоставлялись и учитывались. Эта задача функционально делится на несколько частей. За правильную работу сетевой инфраструктуры и оборудования отвечает OSS. Взаимодействие с абонентами (учет предоставленных услуг по тарифам, контроль состояния счета, выставление счетов и т.д.) происходит во второй части системы – BSS. Основа BSS – биллинговая система, в которой происходят все финансовые взаиморасчеты с абонентами.

II. eTOM МОДЕЛЬ

Для создания оптимального алгоритма управления бизнес-процессом прежде всего необ-

ходимо провести анализ существующего процесса, также необходимо провести оценку эффективности его подпроцессов. В рамках данной статьи был произведен анализ бизнес-процесса предоставления услуги согласно модели eTOM.

eTOM – многоуровневая, эталонная модель бизнес-процессов управления производством, является базой для анализа и проектирования бизнес-процессов в отрасли связи и ориентиром при проектировании и разработке OSS/BSS систем. На концептуальном уровне модель eTOM состоит из трех основных областей: «Стратегия, Инфраструктура, Продукт», «Операционные (производственные) процессы» и «Управление предприятием» [1], каждая из которых имеет многоуровневую иерархическую структуру декомпозиции бизнес-процессов от концептуального (нулевого) до достаточно детализированного (третьего) уровня. В рамках данной статьи производится анализ процесса предоставления услуги в области «Операционные (производственные) процессы» на 2 уровне декомпозиции согласно eTOM. Также производится количественная оценка эффективности подпроцесса обработки заказа, для чего строится модель процесса в среде моделирования Arena Simulation.

III. ВЫВОДЫ

Применение eTOM модели для анализа и проектирования бизнес-процессов телекоммуникационных компаний экономит время и затраты на разработку структуры бизнес-процессов, позволяет эффективно ими управлять.

Список литературы

1. What is the Business Process Framework? [Electronic resource] / TM Forum, 2020. – Mode of access: <https://www.tmforum.org/business-process-framework>. – Date of access: 13.04.2020.

Короткевич Мария Сергеевна, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, korotmarie@gmail.com.

Научный руководитель: Шилин Леонид Юрьевич, декан факультета информационных технологий и управления БГУИР, доктор технических наук, профессор, dekfitu@bsuir.by.

ИСПОЛЬЗОВАНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ОБУЧЕНИЯ ИГРОВОГО ПЕРСОНАЖА

Рассматривается реализация обучения игрового персонажа с помощью нейронной сети на основе экспериментов кибер-биолога Михаила Царькова.

ВВЕДЕНИЕ

Суть задачи по обучению игрового персонажа такова: по полю движется змейка, которая уменьшается, если не находит еду, либо увеличивается, если находит. Если змейка врывается в препятствие, то она погибает. Я буду обучать бота, который управляет змейкой. Если змейка разделится, то бот получит в управление ещё одну змейку, которая в свою очередь тоже может разделиться. За основу взят эксперимент со змейками кибер-биолога Михаила Царькова.

I. ОБУЧЕНИЕ БОТА

Сперва был создан бот-рандомайзер. Так я называю бота, который ходит случайным образом. Он необходим для проверки эффективности нейронной сети. При правильном обучении нейросеть должна легко его обыгрывать. Бот получает на вход поле размером 9x9 клеток, в середине которого находится голова змейки. Соответственно наша нейронная сеть будет иметь 81 вход. Порядок расположения клеток, подаваемых на вход, не имеет значения. Сеть при обучении «сама разберется», где что находится. Для обозначения препятствий и других змеек я использовал значения от -1 до 0 (не включительно). Пустые клетки обозначались значением 0.01, а еда 0.99. На выходе нейросети использовалось 5 нейронов для действий:

двигаться влево по оси X; вправо; вверх по оси Y; вниз; делиться пополам.

Движение бота определял нейрон, который имеет самое большое значение на выходе.

II. А/Б ТЕСТИРОВАНИЕ

Для выполнения этой задачи была создана программа, которая делит змеек на две части в зависимости от конфигурации нейронной сети. На поле выпускалось по 20 змеек каждой конфигурации.

Кобяк Валерий Сергеевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, valerakobyak@icloud.com.

Научный руководитель: Навроцкий Анатолий Александрович, заведующий кафедрой кандидат физико-математических наук, доцент, navrotsky@bsuir.by

Все змейки, находящиеся под управлением одного бота, имели одну и ту же нейронную сеть. Чем больше было змеек в его управлении и чем чаще они сталкивались с разными задачами, тем быстрее проходило обучение. Если, например, одна змейка научилась избегать тупиков или делиться пополам, попав в тупик, то автоматически все змейки данного бота, приобрели эти умения. Меняя конфигурацию нейронной сети можно получить хорошие результаты, но этого не достаточно. Для дальнейшего улучшения алгоритма я решил использовать память на несколько ходов.

Для каждого бота я создал память на 8 ходов. В память было записано состояние поля и ход, который предлагал бот. После этого я проводил корректировку весов для всех восьми состояний, которые предшествовали ходу. Для этого я использовал единый коэффициент корректировки, независимый от глубины хода. Таким образом, каждый ход приводил к корректировке весов не один раз, а восемь. Как и ожидалось, боты с памятью начали быстро обыгрывать ботов, которые обучались без использования памяти.

III. ВЫВОДЫ

Предлагаемая мною модификация нейронной сети позволяет быстрее обучить игрового персонажа. Чем больше подконтрольных элементов в сети и чем чаще они сталкивались с разными задачами, тем быстрее проходило обучение нейронной сети. Но есть добавить использование памяти, то бот будет более "умным чем бот, который существует без использования памяти.

1. Сереган, Т. Programming Collective Intelligence: Building Smart Web 2.0 Applications / Т. Сереган // 2011. – №2. – С. 126-2301.

НЕИНВАЗИВНЫЕ НЕЙРОКОМПЬЮТЕРНЫЕ СИСТЕМЫ КАК ТЕХНОЛОГИЯ РЕАБИЛИТАЦИИ

Рассматривается применение неинвазивных нейрокомпьютерных систем как технологии реабилитации после перенесенного инсульта или травм конечностей.

I. ВВЕДЕНИЕ

Современные реабилитационные методы включают в себя использование нейрокомпьютерных систем (систем мозг-компьютер, нейроинтерфейсов) как на основе только ЭЭГ [1], так и смешанных вариаций – ЭЭГ и ЭМГ [2], ЭЭГ и VR [3], ЭЭГ и визуальные интерфейсы [4] и др.

II. АНАЛИЗ СУЩЕСТВУЮЩИХ СИСТЕМ

Реабилитационные технологии, основанные на использовании нейрокомпьютерных систем часто узкоспециализированы и однозадачны. Такая специфика обусловлена сложностью разработки универсальной реабилитационной технологии, которая требует предельно корректного считывания и обработки импульсов регистрируемой мозговой деятельности.

К примеру, в исследовании Т. Karascony, J.P. Hansen, H.K. Inversen, S. Puthusserypady [3] используется тонкая настройка параметров при помощи классификации, основанной на глубоком обучении, и ответа в игровом VR-интерфейсе, для восстановления моторных образов после инсульта (Рис.1).



Рис.1 – VR игра: руки "парят" в позиции, близкой к естественной, позволяя мозгу заполнить пустое расстояние между руками и торсом. Цель игры - ловить падающие фрукты и отталкивать падающие мячи.

Исследование А. Chiedhury, Н. Raza, А. Dutta, G. Prasad [2] использует более традици-

онный подход, совмещая ЭЭГ и ЭМГ, для корректного контролирования экзоскелета одной из кистей рук. Результаты их работы показали, что использование совмещенной системы ЭЭГ–ЭМГ показывает точность около 90%, в то время как использование только ЭЭГ – около 79%.

Система «НейроЧат», созданная российскими исследователями, помимо ЭЭГ использует модуль визуальной стимульной среды [4] для обеспечения коммуникации реабилитируемых больных. Управление в данной системе осуществляется при помощи фокусировки взгляда на одном из элементов управления в модуле визуальной стимульной среды.

III. ВЫВОДЫ

Согласно статистике ACM Digital Library [5] интерес к технологиям реабилитации на основе нейроинтерфейсов начал расти в 2005–2006 году. На данный момент исследователями предложены различные вариации нейроинтерфейсных систем для реабилитации после инсультов или травм конечностей.

1. Программа XI Международного конгресса «Нейрореабилитация 2019» [Электронный ресурс] – Режим доступа : <https://fnkcr.ru/prm-journal/wp-content/uploads/2019/10/prog.pdf>. – Date of access : 23.01.2020.
2. A. Chiedhury, H. Raza, A. Dutta, G. Prasad EEG-EMG based Hybrid Brain Computer Interface for Triggering Hand Exoskeleton for Neuro-Rehabilitation, AIR '17: Proceedings of the Advances in Robotics June 2017 Article No.: 45 Pages 1–6.
3. T. Karascony, J.P. Hansen, H.K. Inversen, S. Puthusserypady Brain Computer Interface for Neuro-rehabilitation With Deep Learning Classification and Virtual Reality Feedback, AH2019: Proceedings of the 10th Augmented Human International Conference 2019 March 2019 Article No.: 22 Pages 1–8
4. Патент «Нейрокомпьютерная система для выбора команд на основе регистрации мозговой активности» [Электронный ресурс] – Режим доступа : <https://findpatent.ru/patent/262/2627075.html>. – Date of access : 13.02.2020.
5. ACM Digital Library [Electronic resource] – Mode of access : <https://dl.acm.org/>. – Date of access : 15.02.2020.

Кривальцевич Елена Викторовна, аспирант-соискатель, кафедра информационных технологий автоматизированных систем, БГУИР, elena.krivaltsevich@gmail.com.

Научный руководитель: Навроцкий Анатолий Александрович, заведующий кафедрой информационных технологий автоматизированных систем, кандидат физико-математических наук, доцент, navrotsky@bsuir.by.

МЕТОД ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ТЕХНИКИ ИНТЕРПОЛЯЦИИ И КЛАСТЕРИЗАЦИИ

Рассматривается объединённое использование методов кластеризации и интерполяции для принятия решений при оценке стоимости автомобилей, бывших в эксплуатации.

ВВЕДЕНИЕ

Современные методы анализа данных находят все возрастающее применение в различных прикладных сферах, одна из них - оценка стоимости автомобилей, бывших в эксплуатации, требующая принятия во внимание комплекса критериев. Если число критериев не выше трех, то с этой задачей справится человек-эксперт, в ином случае - необходимо привлечение средств автоматизации. Рассмотрим применение для решения данной задачи методов интерполяции и кластеризации.

I. ПРИМЕНЕНИЕ КЛАСТЕРНОГО АНАЛИЗА

Кластерный анализ - многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы [1] (см.рис.1.).



Рис. 1 – Результат кластерного анализа

Кластерный анализ позволит решить задачу проверки гипотезы для определения, действительно ли выделенная группа присутствует в имеющихся данных. Для проведения кластерного анализа выделим следующие этапы:

- отбор выборки для кластеризации;

Кривда Андрей Сергеевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, a.kryuda@gmail.com.

Научный руководитель: Герман Юлия Олеговна, кандидат технических наук, доцент, julgerman@gmail.com.

- определение множества параметров характеристик автомобиля, по которым они будут оцениваться в выборке;
- вычисление меры сходства между автомобилями;
- применение метода кластерного анализа для создания групп сходных по характеристикам авто;
- проверка достоверности кластерного решения.

Также можем воспользоваться иерархической кластеризацией для выделения подгрупп автомобилей, что позволит проводить анализ на более конкретизированных данных.

II. ПРИМЕНЕНИЕ МЕТОДА ИНТЕРПОЛЯЦИИ НА ПОЛУЧЕННЫХ ГРУППАХ

Интерполяция - это метод нахождения неизвестных промежуточных значений некоторой функции по имеющемуся дискретному набору ее известных значений[2].

Функцию, для которой будем применять метод интерполяции, мы получим исходя из оценки значимости характеристик автомобиля. Также важную роль в этом процессе будет играть группа, к которой автомобиль будет относиться после кластеризации.

Применив метод интерполяции на полученных группах после кластеризации, мы сможем достаточно точно оценить стоимость автомобиля, подпадающего под определенную группу.

III. ВЫВОДЫ

Предлагаемая техника выделения кластеров автомобилей с последующим применением на них методов интерполяции позволит в значительной мере ускорить и улучшить существующие процессы для принятия решений при оценке стоимости автомобилей.

1. Мандель, И. Д. Кластерный анализ. Кластерный анализ. — М.: Финансы и статистика, 1988. — 176 с.
2. Информационная площадка аналитической платформы Loginom [Электронный ресурс] – Режим доступа: <https://wiki.loginom.ru>. – Дата доступа: 10.04.2020.

ОБЗОР DEERFAKE ТЕХНОЛОГИЙ И РЕШЕНИЙ В ОБЛАСТИ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Технология Deepfake использует глубокие нейронные сети для убедительной замены на видео одного лица другим. У этой технологии есть очевидный потенциал для злонамеренного использования, и она становится всё более распространённой. Есть все основания полагать, что Deepfake-технология в следующие годы будет становиться только лучше, быстрее и дешевле. Ключевые слова: Deepfake, глубокое обучение, преобразование изображений и видео, Компьютерное зрение – теория и технология создания машин, которые могут производить обнаружение, отслеживание и классификацию объектов.

ВВЕДЕНИЕ

В последние годы в области глубокого обучения и компьютерного зрения произошло множество революций и появилось много новых технологий. У некоторых технологии, таких как Deepfake, есть очевидный потенциал для злонамеренного использования, и по поводу социальных и политических последствий этого тренда уже многое известно. В следующих пунктах представлены более подробные сведения о техническом устройстве Deepfake решений.

I. ВЫЧИСЛИТЕЛЬНЫЕ МОЩНОСТИ И ДАННЫЕ

Видео называемые Deepfake [«глубокими подделками»] получили свое название потому, как создаются с использованием глубоких нейросетей. За последнее десятилетие специалисты по информатике обнаружили, что нейросети становятся всё более мощными при добавлении дополнительных слоёв из нейронов. Но чтобы раскрыть весь потенциал глубоких нейросетей, нужно очень много данных и огромные вычислительные мощности. Под вычислительными мощностями рассматриваются GPU, с большим объемом графической памяти и наибольшим количеством CUDA-ядер. Для создания Deepfake решения также требуется множество видеороликов с исходным объектом на кадрах видеозаписи и такое же множество видеороликов с объектом используемым для замены.

II. ПРИНЦИП РАБОТЫ

В основе ведущих программных пакетов для создания Deepfake – находится автоэнкодер. Это нейросеть, обученная принимать на вход изображение и выдавать идентичное изображение. Само по себе это умение может быть не таким уж полезным, но, как увидим далее, это ключевой строительный блок в процессе создания Deepfake.

Автоэнкодер берёт это компактное представление, известное, как «латентное пространство», и пытается развернуть его, получив изначальное изображение. С одной стороны сети находится энкодер, принимающий изображение и сжимающий его до небольшого числа пере-

менных. С другой стороны нейросети находится декодер. Он берёт это компактное представление, известное, как «латентное пространство», и пытается развернуть его, получив изначальное изображение.

Искусственно ограничение количества данных, передаваемых от энкодера к декодеру, заставляет две этих сети разработать компактное представление человеческого лица. Энкодер – это что-то вроде алгоритма сжатия с потерями, который пытается сохранить как можно больше информации о лице при ограничениях на объём хранилища. Латентное пространство должно каким-то образом извлечь важные детали, например, в какую сторону смотрит субъект, открыты у него глаза или закрыты, улыбается он или хмурится.

Важно, что автоэнкодеру нужно сохранить только те особенности лица, которые меняются во времени. Ему не нужно хранить неизменные вещи тип цвета глаз или формы носа. Если на каждой фотографии человека у него голубые глаза, тогда декодер его сети обучится автоматически рисовать его лицо с голубыми глазами. Нет нужды записывать в тесное латентное пространство информацию, не меняющуюся при переходе от одного изображения к другому.

Каждому алгоритму для обучения нейросети нужен какой-то способ оценить качество работы сети, чтобы его можно было улучшить. Во многих случаях это делается через обучение с учителем, когда человек обеспечивает правильный ответ для каждого элемента из набора обучающих данных. Автоэнкодеры работают по-другому. Поскольку они просто пытаются воспроизвести собственные входные данные, обучающее ПО может судить об их качестве работы автоматически, что принято называть обучением без учителя в терминологии машинного обучения.

Как и любая нейросеть, автоэнкодеры в Deepfake сетях обучаются при помощи обратного распространения. Обучающий алгоритм получает определённое изображение в нейросеть и смотрит, какие пиксели на выходе не соответствуют входу. Затем он подсчитывает, какие из нейронов последнего слоя внесли наибольший вклад в

ошибки и немного подправляет параметры каждого нейрона так, чтобы он выдавал результаты получше.

Затем эти ошибки распространяются обратно, к предыдущему слою, где параметры каждого нейрона подправляются вновь. Ошибки распространяются таким способом всё дальше назад, пока каждый из параметров нейросети – как у энкодера, так и у декодера – не окажутся исправленными.

Затем обучающий алгоритм скармливает ещё одно изображение сети, и весь процесс повторяется снова. Могут понадобиться сотни тысяч таких повторов для того, чтобы получился автоэнкодер, хорошо воспроизводящий собственный вход.

ПО для Deepfake работает, параллельно обучая два автоэнкодера, один для оригинального лица, а второй – для нового. Во время процесса обучения каждому автоэнкодеру выдают изображения только одного человека, и он обучается выдавать изображения, очень похожие на оригинал. Схема работы автоэнкодеров (Рис. 1).

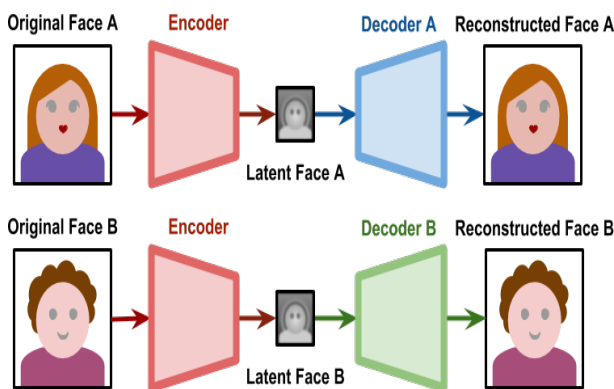


Рис. 1 – Схема работы автоэнкодеров

Есть, однако, некоторые трудности: обе сети используют один и тот же энкодер. Декодеры – нейроны в правой части сети – остаются раздельными, и каждый из них обучается выдавать разное лицо. Но нейроны в левой части сети имеют общие параметры, меняющиеся каждый раз, когда обучается любой из автоэнкодеров. Когда сеть обучается на оригинальном лице, это меняет половину сети, принадлежащую энкодеру и в сети с лицом для замены (fake-лицом).

В итоге у двух автоэнкодеров есть один общий энкодер, способный «считывать» либо настоящее лицо, либо поддельное. Цель энкодера в том, чтобы использовать одинаковое представле-

ние таких вещей, как угол наклона головы или расположение бровей. А это, в свою очередь, означает, что когда вы сжали лицо при помощи энкодера, его можно распаковать при помощи любого декодера.

После обучения таким способом пары автоэнкодеров, остаётся последний шаг для создания Deepfake: изменение декодеров местами (Рис. 2).

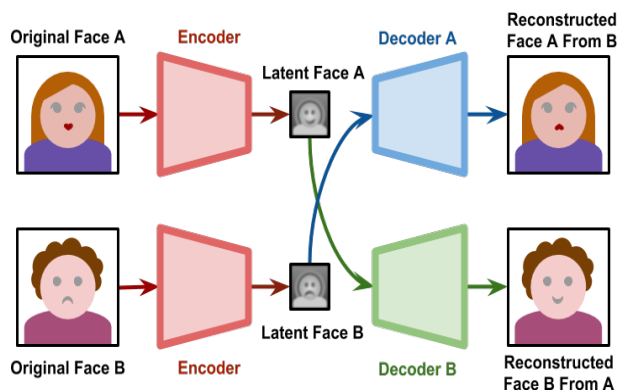


Рис. 2 – Схема изменения декодеров местами

В итоге получается реконструированная fake-фотография – но с тем же положением головы и выражением лица, как и у оригинальной фотографии.

III. ЗАКЛЮЧЕНИЕ

Резонанс Deepfake технологий очевидно вызывает беспокойство. До недавнего времени люди могли с достаточной легкостью принимать видеозапись с человеком за чистую монету. Появление ПО для создания Deepfake и других цифровых инструментов привело к тому, что теперь люди относимся к видеозаписям со скептицизмом. В настоящее время ролик, в котором человек утверждает что-то скандальное – или раздается – стоит рассмотреть возможность того, что некто подделал это видео с целью дискредитации того человека.

IV. СПИСОК ЛИТЕРАТУРЫ

1. Основы современного искусственного интеллекта: как он работает, и уничтожит ли наше общество уже в этом году? <https://habr.com/ru/post/451214/> Часть 1.
2. Эти новые уловки пока ещё способны перехитрить видеоролики от Deepfake <https://habr.com/ru/post/429192/>
3. Я создал свой собственный Deepfake за две недели — <https://habr.com/ru/post/482684/>.

Кулыба В. А., магистрант факультета информационных технологий и управления БГУИР, kulyba.vadim@gmail.com

Научный руководитель: Гуринович Алевтина Борисовна, кандидат физ.-мат. наук, доцент, gurinovich@bsuir.by

СИСТЕМА ОПЕРАТИВНОГО ПЛАНИРОВАНИЯ МЕДИЦИНСКОГО ЦЕНТРА

Рассматривается вопрос оперативного планирования в системе здравоохранения. Предлагается использование двух моделей оптимизации, решающих поставленную задачу.

ВВЕДЕНИЕ

В последние годы системы здравоохранения сталкивались с проблемой оказания высококачественной помощи при ограниченных ресурсах. В этом контексте, учитывая необходимость сдерживания затрат, решающее значение играет разработка действий и процедур, которые улучшат поток пациентов, обеспечат своевременное лечение и будут максимально использовать имеющиеся ресурсы. Процесс, с помощью которого пациенты назначаются для лечения, имеет решающее значение для эффективности потока пациентов и, таким образом, он сильно влияет на общую эффективность систем здравоохранения. Планирование пациентов связано с оптимальным распределением пациентов к медицинским ресурсам. Проблема сосредоточена в процедурах, которые определяют, как назначаются встречи с пациентами, то есть когда и как они назначаются в определенный день и их продолжительность. Правила планирования определены для того, когда можно назначать прием и время между приемами. Основная цель - обеспечить эффективное использование медицинских ресурсов и предоставление правильного лечения пациенту в нужное время.

I. ЗАДАЧА МОДЕЛИ ОПТИМИЗАЦИИ

Планирование пациентов является довольно сложной задачей, потому что преследуются несколько целей и необходимо учитывать большой набор ограничений. Следовательно, эмпирические и эвристические решения не рассматриваются, в то время как разработка математических моделей, количественных методов и многоагентных систем может дать значительно улучшенные результаты.

Задача моделей оптимизации состоит в том, чтобы максимально увеличить число пациентов, подлежащих планированию, с учетом следующих условий:

1. значение приоритета, назначенного пациенту, определяемое на основе «тяжести» болезни пациента;
2. количество сеансов лечения e_j для каждого пациента j ;

3. сеансы лечения для каждого пациента должны проводиться в течение нескольких дней подряд;

4. каждый пациент должен пройти курс лечения в течение определенного количества последовательных недель.

II. БАЗОВАЯ И РАСШИРЕННАЯ МОДЕЛИ

Базовая модель предполагает большинство данных принимать за статические. Одно среднее неизменное время лечения, знание о том, что первый сеанс всегда требует большего количества времени, чем все последующие (диагностика, изучение жалоб и анализов). Целевая функция получается как сумма двух слагаемых. Первый член представляет общее количество запланированных амбулаторных больных (вес является минимальным для пациентов с максимальным значением приоритета). Во втором члене используется фактор, позволяющий избежать генерирования решений, которые являются эквивалентными. Этот фактор позволяет различать пациентов с одинаковым значением приоритета и одинаковым количеством дней лечения на основе доступа к списку.

Расширенная модель продолжает гарантировать, что начавшие лечение пациенты включены в оптимальное решение, но есть возможность изменять график лечения для клиентов и предлагать лечения клиентам из списка ожидания.

III. ВЫВОДЫ

Модели оперативного планирования предлагают медицинскому персоналу возможность обнаруживать свободные слоты в течение всего периода, когда пациенты нуждаются в приеме. Таким образом, пациенты получают полный список назначений на определенный период. В частности, медицинский персонал вводит план лечения в систему и согласовывает первую встречу с пациентами для планирования лечения.

1. Conforti, Guerriero, Guido. "Optimization models for therapy patient scheduling".

Ланин Евгений Михайлович, магистрант факультета информационных технологий и управления БГУИР, yauhenilanin@gmail.com

Научный руководитель: Ревотюк Михаил Павлович, кандидат технических наук, доцент, rmp@bsuir.by

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЛАЗЕРНОЙ РАЗМЕРНОЙ ОБРАБОТКИ

Математическое моделирование температурно-временных зависимостей и распределение тепловых полей на поверхности кремневой пластины при нагреве лазерным источником.

ВВЕДЕНИЕ

3D электронные модули, которые способны интегрировать множество микросхем на основе различных технологий в один единый корпус, позволяют создать устройств малых размеров, широким набором функций и доступных по стоимости. Одно из современных направлений реализации 3D-модулей является технология TSV (Through Silicon Vias – сквозные отверстия в кремнии). Лазерная обработка – один из методов формирования переходных отверстий.

Целью математического моделирования является получение зависимостей распределения тепла на поверхности кремневой пластины подверженной лазерному нагреву. Моделирование включает в себя следующие этапы: задания параметров модели кремниевой пластины и параметров лазерного источника, задание свойств анализа и проведение расчета, просмотр и анализ результатов.

1. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЛАЗЕРНОГО НАГРЕВА

Температура поверхности зависит от мощности, поглощенной единицей площади. Тогда энергия поглощенного излучения [2]:

$$E = P \cdot \tau = q \cdot S \cdot \tau(1)$$

где q – плотность мощности; P – мощность введенная через поверхность; S – площадь облучаемой поверхности; τ – длительность воздействия.

Соотношение между распределением температуры по поверхности $T(x,y,z)$ в зависимости от времени воздействия и плотностью мощности q может быть определено из уравнений теплопроводности типа [2]:

$$\frac{\delta T(x, y, z, t)}{\delta t} - \alpha \cdot \Delta T(x, y, z, t) = \frac{Q(x, y, z, t)}{\rho \cdot c} \quad (2)$$

где Δ – оператор Лапласа, Q – объемная плотность поглощенного светового потока, α – температуропроводность ($\alpha = k/\rho c$, где k – теплопроводность, ρ – плотность, c – теплоемкость), x, y, z, t – аргументы функции.

Объемная плотность поглощенного светового потока с учетом поглощательной способности и направлением по оси z [2]:

$$Q(x, y, z, t) = q_0 \cdot A \cdot e^{-\alpha z} \quad (3)$$

где q_0 – плотность мощности падающего излучения, A – поглощательная способность на волне

излучения; α – линейный коэффициент поглощения на волне излучения, m^{-1} ; z – координата отсчитываемая от поверхности в глубь материала.

Особенностью задач, связанных с нагревом материалов лазерным излучением является то, что распределение излучения и, следовательно, тепловой источник, как правило, обладает резко изменяющимися пространственно-временными характеристиками. Поэтому идеализация свойств тепловых источников, часто допускаемая в расчетных схемах для уменьшения математических трудностей, может приводить к отклонениям расчетных данных от экспериментальных. [2] Таким образом, функция гауссового импульса позволяет описать профиль излучения лазерного источника и задать нормальное распределение плотности мощности падающего лазерного излучения. Распределение тепла по поверхности будем считать равным во всех направлениях от точки (x_0, y_0) нормального падения луча:

$$Q(x, y, z, t) = q_0 \cdot A \cdot e^{-\alpha z} \cdot \frac{\alpha}{\pi r^2} \cdot e^{-\left(\frac{(x-x_0)^2 + (y-y_0)^2}{2r^2}\right)} \quad (4)$$

где $Q(x, y, z, t)$ – функция, характеризующая распределение плотности поглощения светового потока на поверхности материала и изменение его во времени, z – ось перпендикулярная поверхности и отсчитываемая от поверхности вглубь материала, r – радиус пятна излучения.

Теплообменом облучаемой поверхности с окружающей средой можно пренебречь. Задаем действие источника на поверхности ($z = 0$) граничным условием [4]:

$$-k \cdot \frac{\delta T(x, y, z, t)}{\delta z} |_{(z=0)} = Q(x, y, t) \quad (5)$$

Если размеры нагреваемого тела достаточно велики ($l_x, l_y, l_z \gg \sqrt{\alpha \tau}$), то его можно считать полубесконечной средой:

$$T_{x \pm \infty} = T_{y \pm \infty} = T_z = T_H, \quad (6)$$

где T_H – начальная температура ($T_{t=0} = T_H = 293, 15$).

Теплообменом облучаемой поверхности с окружающей средой можно пренебречь. Решив уравнение теплопроводности, получаем распределение температуры по поверхности при использовании источника с гауссовым профилем [3]:

$$T(x, y, z, t) = \frac{2p}{\rho c (4\pi\alpha)^{3/2}} \int_0^t \frac{1}{(\tau + t)\sqrt{t}} \quad (7)$$

где ρ – мощность излучения, введенная в материал через площадку s ; $\tau = \alpha/4r$; r – радиус пятна излучения; a – коэффициент теплопроводности; c – удельная теплоемкость; ρ – плотность материала; t – текущее время.

II. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ В СРЕДЕ COMSOL MULTIPHYSICS 5.3a

В результате моделирования получены температурно-временные зависимости (рис 1) и тепловые поля на поверхности кремневой пластины (рис 2). По температурно-временным зависимостям видно, что профиль температуры имеет мгновенный рост до температуры 2000 после чего, в течении 3 секунд температура достигает своего пика. При продолжении процесса происходит нагрев всего объема кремневой пластины.

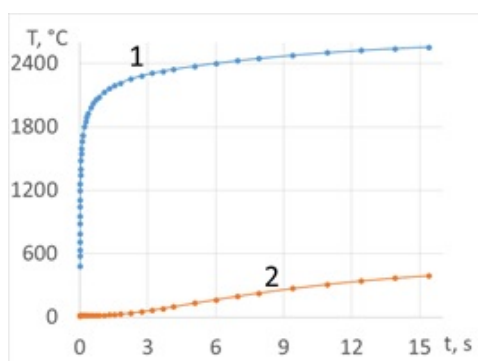


Рис. 1 – Температурно-временные зависимости на поверхности кремневых подложек, 1 – максимальная температура в точке воздействия, 2 – минимальная температура кремневой пластины

В качестве источника лазерного нагрева выбран: Lotis ТП LS-2132UTF с длиной волны 1064мкм, энергия импульса 170мДж, длительность импульса 5-6нс, частота импульса 15Гц, диаметр выходного пучка 5мм, мощность - 750Вт.

Моделирование распределения температуры по поверхности кремневой пластины и зависимостей скорости роста температур, выполнено в программном пакете Comsol Multiphysics 5.3a. Для расчета были заданы исходные и граничные условия, и характеристики лазерного источника. Окружающая среда – воздух в нормальных условиях.

Лаппо Александр Игоревич, аспирант кафедры электронной техники и технологии БГУИР, lappo@bsuir.by

Ярмолик Валерий Иванович, ассистент кафедры информационных технологий автоматизированных систем БГУИР, v.jarmolik@bsuir.by

Наврость Маргарита Сергеевна, студент 4 курса, факультет ФИТУ

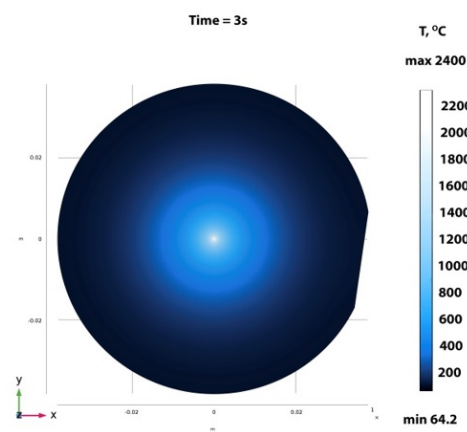


Рис. 2 – Распределение тепловых полей на поверхности кремневой пластины

В качестве модели использовалась кремневая пластина с диаметром 25.4мм, толщина - 200мкм, влучательная способность поверхности пластины - 0.8, плотность - 2329кг/м³, теплоемкость - 700Дж/(кг*К), температура кипения и испарения - 2623К.

ЗАКЛЮЧЕНИЕ

По результатам распределение тепловых полей на поверхности кремневой пластины, видно, что для получения искомой температуры испарения кремния необходимой для формирования переходного отверстия, составляющей 2400°С для лазера потребовалось 3 секунды. Так же выявлено что с ростом продолжительности процесса растет температура всей пластины, что может негативно сказаться на качестве формируемых отверстий из-за формирования наплывов на большей площади от точки воздействия.

1. COMSOL Multiphysics ПО для мультифизического моделирования [Электронный ресурс]. – 2019. – Режим доступа: <https://www.comsol.ru/>
2. Либенсон М.Н., Яковлев Е.Б., Шандыбина Г.Д. Взаимодействие лазерного излучения с веществом (силовая оптика). Часть II. Лазерный нагрев и разрушение материалов. Учебное пособие. Под общей редакцией В.П. Вейко – СПб: НИУ ИТМО, 2014. –181с.
3. Вейко В.П., Шахно Е.А. Сборник задач по лазерным технологиям. Изд. 3-е, испр. и дополн. – СПб: СПбГУ ИТМО, 2007. – 67 с.

МОНИТОРИНГ ЖИВОТНЫХ

Рассматриваются способы спутникового мониторинга животных.

ВВЕДЕНИЕ

В современных реалиях необходимость наблюдения за объектами стала крайне актуальна. Каждый день люди сталкиваются с этим. Транспортному бизнесу важно знать где находится автомобиль, количество топлива потраченного за поездку, людям, которые пользуются общественным транспортом, важно знать как скоро появится нужный автобус на остановке, владельцам животных очень важно знать где его питомец, и в случае потери быстро отыскать его, в сельском хозяйстве важна статистика по животным. Все эти задачи решает спутниковый мониторинг. На сегодня создано не очень много систем контроля животных, а это направления является наиболее перспективным.

I. ПРИНЦИП РАБОТЫ СИСТЕМЫ СПУТНИКОВОГО МОНИТОРИНГА

Чтобы начать отслеживать объект в системе мониторинга, на нем устанавливается трекер – ГЛОНАСС/GPS-терминал, с определенной частотой принимающий сигналы от спутников глобальных навигационных систем. По географическим координатам точки приема определяется положение объекта, по изменению положения за время получения или передачи сигнала [1].

Данные о координатах по каналам сотовой связи передаются в центр обработки данных, попадают с соответствующими СУБД офисного центра программы. Сохранение и обработка собираемой информации осуществляется посредством таких баз как Interbase, MS SQL, ORACLE, что дает возможность ввода пользовательских запросов для визуализации местонахождения запрашиваемого объекта, вывода статистических данных в табличном виде, математического подсчета переменных состояний (простой, преодоленный путь и т.п.), а также их учет.

II. СПОСОБЫ МОНИТОРИНГА ЖИВОТНЫХ

Существует множество различных животных и какого-то одного универсального метода для мониторинга всех не существует. Но есть общее у всех - нужен GPS-трекер. Их существует огромное количество. Но есть один большой нюанс в зарядке. Чтобы трекеру отослать любого рода сообщение на сервер, необходимо затратить

какую-то энергию и чем чаще трекер отсылает сообщения тем быстрее он разряжается.

Домашним животные довольно активны, и хозяева чаще всего хотят знать о них всю информацию и состояния их здоровья. В этом случае нужно получать данные с трекера довольно часто. Естественно он довольно быстро будет разряжаться, но это не будет большой проблемой. Сейчас мы подзаряжаем все: телефон, часы, наушники. Подзарядка трекера не занимает много времени, а многие из них могут держаться до двух дней без подзарядки, ну и если уменьшить количество отправляемых сообщений до 1 в час, можно увеличить время работы трекера. Все что нам нужно в этом случае это любой трекер предназначенный для животных, и передаваемый нужные параметры и приложение на смартфоне, которое может подключиться к нему и собирать эти данные в статистику и понятный вид. Не обязательно чтоб трекер и приложение были одной фирмы, на данный момент существует огромное количество различных трекеров, к которым можно подключиться по WiFi или Bluetooth.

Сельскохозяйственные животные не так активны, и в большинстве случаев, в этом случае важно отслеживать местоположение животного и знать как оно передвигалось. Если брать колхозы, где таких животных может быть довольно много, трекером, который работает только два дня не обойтись, так как это очень сложно для персонала. Подзарядить тысячу трекеров крайне не просто и затратно. В этом случае нужен трекер, который будет работать довольно долго, что сейчас возможно.

III. ВЫВОДЫ

Предложенные способы лишь небольшая часть тех вариантов, которые существуют. Как можно видеть на данный момент самая большая проблема заключается в подзарядке GPS-трекеров. Когда она будет решена, люди будут в состоянии следить за животными с постоянно обновляющимися данными.

1. Рассел Джесси Спутниковый мониторинг транспорта / Джесси Рассел // Книга по Требованию. – 2012. – С. 10-30.

Логвинов Денис Олегович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, logvinovdenharo@gmail.com.

Научный руководитель: Давыдов Максим Викторович, первый проректор БГУИР, доцент, кандидат технических наук davydov-mv@bsuir.by.

ОПТИМИЗАЦИЯ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ПРОЛОЖЕНИЯ

Рассматриваются подходы к балансированию недостатков и преимуществ микросервисной архитектуры. Предлагаются подходы и средства для мониторинга, развертывания и разработки программ с микросервисной архитектурой.

ВВЕДЕНИЕ

За последнее десятилетие микросервисная архитектура (МСА) получила широкое распространение в связи с развитием гибких подходов к разработке ПО. В частности МСА предлагает простоту тестирования отдельных частей приложения, независимость развертывания, гибкость в отказоустойчивости и масштабировании как всей системы так и ее отдельных частей. Однако при декомпозиции и разделении на отдельные сервисы возникает вопрос во взаимодействии и интеграции всех частей.

I. ОРГАНИЗАЦИЯ СЕРВИСА ВОКРУГ ЕГО ЦЕЛЕЙ

Определение границ сервиса — самый важный шаг. От этого будет зависеть весь жизненный цикл микросервиса. Основным принципом определения зоны ответственности микросервиса — сформировать её вокруг некоторой бизнес-задачи. И чем она компактнее, чем формализованней её взаимоотношения с другими областями, тем проще создать новый микросервис. На этом основывается создание любых других компонентов. Однако на начальном этапе разработки бывает достаточно сложно определить четкие границы или в середине разработки границы бизнес-задачи сместились, что потребует изменения функциональности в других микросервисах.

Для минимизации ошибок при определении границ оправданным является подход Monolith First, когда вначале систему развивают в традиционной парадигме, а когда появляются устоявшиеся области, их выделяют в микросервисы. А при изменении границ главное, чтобы выигрыш от разбиения превышал сложности пересмотра этих границ.

II. ВЗАИМОДЕЙСТВИЕ И ИНТЕГРАЦИЯ

Одним из самых сложных и важных вопросов проектирования МСА является взаимодействие между сервисами. Цепочка взаимодействия различных сервисов может быть различна

и поэтому стоит использовать асинхронное взаимодействие (Kafka, asynchronous REST и т.д.). Это позволит более гибко настраивать время ожидания от какого-нибудь медленного сервиса или организовать политику retry в случаях ошибки в сети.

Увеличение количество запросов между сервисами вынуждает уделить особое внимание мониторингу как всей системы так и отдельных ее частей (Prometheus, Grafana, Moira и т.п.). Система мониторинга для приложений на основе микросервисов должна отображать постоянное изменение ресурсов, иметь возможность собирать данные мониторинга в центральном местоположении и отображать информацию, отражающую часто меняющийся характер приложений на микросервисах.

Рост количества различных сервисов вынуждает автоматизировать процессы сборки и развертывания. Однако хорошо выстроенный автоматизированный процесс развертывания сервисов по отдельности позволяет осуществлять релизы часто и с нулевым временем простоя, что облегчает процессы тестирования и ускоряет время доставки ПО конечным пользователям.

III. ВЫВОДЫ

Написание независимого сервиса дает свободу в выборе средств и языков программирования наиболее подходящих для решения конкретной задачи. Для построения эффективной микросервисной инфраструктуры требуется инвестировать время в автоматизацию процессов разработки. Это позволит сделать процесс развертывания системы из многих составляющих (микросервисы) прозрачным и единообразным.

1. Martin L. Abbott. R. The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise., Third Edition // Addison-Wesley Professional., — 2015. — С. 624.
2. Martin Fowler, Microservices. - <https://martinfowler.com/articles/microservices.html>
3. Vikram Murugesan, Microservices Deployment Cookbook // Packt Publishing., — 2017. — С. 280

Любанец Андрей Евгеньевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, andr0sparrow@gmail.com.

Научный руководитель: Матвеевко Владимир Владимирович, кандидат физико-математических наук, доцент, vladimir66@bsuir.by.

МЕТОДИКА И СРЕДСТВА АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ИНТЕРФЕЙСА ПРОГРАММНОГО ПРИЛОЖЕНИЯ

Рассматриваются методика и средства автоматизированного тестирования интерфейса программного приложения.

ВВЕДЕНИЕ

Одно из самых ранних упоминаний автоматизированного тестирования встречается во всем нам известной книге Ф. Брукса «Мифический человек-месяц». Ф. Брукс считает, что на тестирование следует выделять 1/4 от времени, выделенного на весь проект.

I. АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ

Итак, что же такое тестирование и автоматизация тестирования. Г. Майерс даёт следующее довольно широкое и объемное определение понятию «тестирование»: «Тестирование ПО – это процесс выполнения программы с целью обнаружения ошибок». Тогда автоматизация тестирования – это процесс выполнения программы с целью обнаружения ошибок автоматически с помощью специализированных инструментов. При регулярном тестировании на этапе разработки программного обеспечения у разработчика будет гарантия того, что приложение работает именно так, как он рассчитывает, особенно при применении практики continuous delivery.

Таким образом, можно сделать вывод, что основными целями автоматизации тестирования есть: снижение и оптимизация затрат на тестирование, сокращение времени на разработку и выпуск релизов, при этом не теряя в качестве, уменьшение количества ошибок, минимизация рисков и улучшение архитектуры. Преимущества автоматизированного тестирования: снижение временных затрат без потери качества, исключение «человеческого фактора», снижение затрат на поддержку скриптов, автоматизация рассылки и сохранения отчётов, возможность распределения нагрузки на ресурсы. Однако, также стоит учитывать и недостатки автоматизации: повторяемость тестов, затраты на поддержку при изменении ПО, затраты на разработку автоматизированных тестов, стоимость инструментов для автоматизации, трудоемкость автоматизации. Отсюда можно сделать вывод, что автоматизация оправдана и необходима при работе над долгосрочными проектами с частыми релизами.

Мехедко Алексей Олегович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, maol19777@gmail.com.

Научный руководитель: Ломако Александр Викторович, кандидат технических наук, доцент, lavlot@bsuir.by.

II. ОСНОВНЫЕ ПРИНЦИПЫ

Автоматизацию тестирования условно разделяют на две основных категории: тестирование на уровне кода (unit testing) и тестирование пользовательского интерфейса (GUI-тестирование). Также разделяют направления автоматизации тестирования: веб-сервисов, мобильных приложений, десктопных приложений, баз данных, веб-приложений. В самом начале разработки важно четко определить функциональные и нефункциональные требования к ПО и иметь эталоны для сравнения текущего функционирования системы с эталонным. Автоматизацию тестирования часто представляют в виде пирамиды, в основании которой лежит модульное тестирование, на следующем уровне – интеграционное тестирование, уровнем выше – приемочное тестирование, выше – тестирование пользовательского интерфейса и на вершине пирамиды – мануальное тестирование. При автоматизации тестирования необходимо использовать стандарт IEEE 829- 1983, который описывает планирование, спецификацию тестов и отчёты и пр. Безусловно, для автоматизации тестирования при разработке ПО инструменты

III. ВЫВОДЫ

На ранних этапах развития компьютерных систем программирование и тестирование проводилось без помощи специальных инструментов. Однако, увеличение

функциональности, сложности и масштабируемости проектов сделало ручное тестирование менее эффективным и более сложным и дорогим. Следующим логичным шагом в тестировании при разработке программного обеспечения и становится автоматизация.

1. Брукс Ф. Мифический человек-месяц или как создаются программные системы /
2. IEEE-SA / The Institute of Electrical and Electronics Engineers Standards Association / Mode of access : <https://standards.ieee.org/standard/829-1983.html> - Date of access : 14.03.2020

АЛГОРИТМ УЛУЧШЕНИЯ КАЧЕСТВА МЕДИАФАЙЛОВ

ВВЕДЕНИЕ

При передаче и размещении в интернете медиафайлов требуется уменьшение их размеров, для чего используются различные алгоритмы сжатия. Большинство эффективных алгоритмов осуществляют сжатие с потерей части информации. При многократном сохранении искажения становятся значительными. Поэтому актуальным является поиск новых и улучшение известных алгоритмов улучшения качества изображений.

I. ОСНОВНЫЕ ПОДХОДЫ

Известно несколько основных подходов по улучшению изображения: предсказательные модели, краевые методы, статистические методы, методы основанные на паттернах, метод преобразования гистограмм, подход на основе сверточной нейронной сети.

1) Предсказательное моделирование основывается на построении, управлении и расчете моделей при помощи техник аппроксимации.

2) Статистический метод – метод, основанный на анализе большой выборки данных (что и является статистикой);

3) Метод преобразования гистограмм. В данном методе пытаются достичь равномерности распределения яркостей обработанного изображения.

4) Сверточная нейронная сеть - это сеть, использующая множество идентичных копий одного и того же нейрона. Сеть позволяет иметь ограниченное число параметров при вычислении больших моделей. Сверточная нейронная сеть, однажды обучив нейрон, использует его много раз, что облегчает обучение модели и минимизирует ошибки.

Наилучшее качество дают метод преобразования гистограмм и подход на основе сверточной нейронной сети.

II. ОБЗОР ПОДХОДА НА ОСНОВЕ СВЕРТЧНОЙ НЕЙРОННОЙ СЕТИ

Пусть у нас есть изображение Y , которое мы хотим восстановить ($F(Y)$), и оно должно быть максимально похожее на основное изображение высокого разрешения X . Данный подход

состоит из трех операций: 1) Извлечение и представление патчей: эта операция извлекает (перекрывающиеся) патчи из изображения Y с низким разрешением и представляет каждый патч как вектор большой размерности. 2) Нелинейное отображение: эта операция нелинейно отображает каждый многомерный вектор на другой многомерный вектор. Каждый сопоставимый вектор концептуально представляет патч высокого разрешения. 3) Реконструкция: эта операция объединяет вышеприведенные патч-представления высокого разрешения для генерации окончательного изображения высокого разрешения. Первая операция выражается как:

$$F_1(Y) = \max(0, W_1 * Y + B_1), \quad (1)$$

Где W_1 и B_1 представляют фильтры и смещения соответственно, а «*» обозначает операцию свертки. На выходе после первой операции мы получим для каждого патча n_1 -мерный массив. Во второй операции происходит преобразование этого массива в n_2 -мерный, увеличивая размер первого массива для повышения точности. Каждый из выходных n_2 -мерных векторов концептуально представляет собой патч высокого разрешения, который будет использоваться для реконструкции:

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2). \quad (2)$$

В традиционных методах предсказанные перекрывающиеся участки с высоким разрешением часто усредняются для получения окончательного полного изображения. Основываясь на этом, мы определяем сверточный слой для получения окончательного изображения с высоким разрешением:

$$F(Y) = W_3 * F_2(Y) + B_3. \quad (3)$$

III. ВЫВОД

В результате проделанной работы было установлено, что на основе сверточной нейронной сети можно реализовать высокоэффективный алгоритм улучшения качества медиафайлов.

Минько Дмитрий Александрович, магистрант факультета информационных технологий и управления, 98_miniko@mail.ru.

Научный руководитель: Навроцкий Анатолий Александрович, заведующий кафедрой информационных технологий автоматизированных систем БГУИР, кандидат физико-математических наук, доцент, navrotsky@bsuir.by.

ОПРЕДЕЛЕНИЕ СОВОКУПНОСТИ МЕТРИК ДЛЯ РАСЧЕТА ЭФФЕКТИВНОСТИ РАБОТЫ ПРОГРАММНЫХ ПРОДУКТОВ

Цель исследования - определить последовательность действий при выборе оцениваемых параметров программного продукта, степени его важности и метрик расчета выбранных показателей.

ВВЕДЕНИЕ

Современные программные продукты (ПП) представляют собой сложные решения, характеризующиеся большим разнообразием выполняемых действий и характеристик. Зачастую эти характеристики слабо связаны, либо не связаны вовсе между собой, однако каждая из них в той или иной степени влияет на работоспособность и эффективность ПП. Для отражения общего состояния программного продукта оценка эффективности должна включать в себя анализ сразу по нескольким критериям. Это приводит к проблеме разнородности данных, которые необходимо привести к общему значению. Также очевидно, что критерии оценки, их совокупность, а также «вес» каждого из факторов отличается для каждого конкретного программного продукта. Использование единых методов оценки для разных типов программных средств может привести к неправильным или неполным результатам, что приводит к необходимости описания последовательности действий для определения параметров оценки программных средств.

I. ОПРЕДЕЛЕНИЕ СПИСКА КРИТЕРИЕВ И ИХ ЗНАЧИМОСТИ

В большинстве случаев невозможно объективно выявить факторы, отражающие общее состояние приложения. Это вызвано сложностью программных продуктов, разнородностью выполняемых задач, различиями в требованиях к работе. Для решения этой задачи разработано большое количество методов, значительно упрощающих и формализующих процесс выбора. Одним из способов принятия решения о выборе конкретных характеристик системы, а также способов расчета выбранных показателей, является метод экспертных оценок. Он используется в случаях, когда объект полностью или частично не поддается предметному описанию или математической формализации [1]. Суть данного метода заключается в получении оценки на основе мнения одного или нескольких специалистов (экспертов) с целью последующего принятия ре-

шения. Область применения методов экспертных оценок значительно шире рассмотренной, в процессе оценки программного обеспечения методы экспертных оценок также могут быть применены для поиска возможных решений проблем, прогнозирования, путей развития и т.д.

II. ДАЛЬНЕЙШИЕ ДЕЙСТВИЯ С ПОЛУЧЕННЫМИ РЕЗУЛЬТАТАМИ

После выбора и расчета характеристик программного продукта, возможен расчет оценки эффективности его работы. Для этого могут использоваться модифицированные методы сравнительного анализа, основанные на вычислении обобщенной оценки и последующем сравнении по ее результатам. Примером таких методов являются метод комплексной оценки или сравнение с использованием функций полезности. Различие с оригинальными методиками заключается в том, что для сравнения используется эталонное состояние объекта, а не его альтернативы. Также возможен поиск наилучшей стратегии развития программного продукта с применением методов однокритериальной или многокритериальной оптимизации.

III. ВЫВОДЫ

Несмотря на то, что невозможно полностью формализовать процесс выбора критериев оценки эффективности программного средства, имеется возможность описать базовую последовательность действий для их поиска в зависимости от типа ПП. По мере накопления данных о различных ПП, результатов их работы и проведенных расчетов, возможен анализ результатов и формирование более точных и подходящих инструкций по проведению исследований других программных средств, близких по назначению и функциональности.

1. Анохин, А. Н. Методы экспертных оценок / А. Н. Анохин // Редакционно-издательский отдел ИАТЭ, 1996. – 148 с.

Осецкая Анна Валентиновна, магистрант кафедры информационных технологий автоматизированных систем БГУИР, annaos4211@gmail.com.

Научный руководитель: Муха Владимир Степанович, доктор технических наук, профессор, mukha@bsuir.by.

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ГОЛОСОВОГО ВЗАИМОДЕЙСТВИЯ С КЛИЕНТАМИ В КОНТАКТ-ЦЕНТРАХ

ВВЕДЕНИЕ

Контакт-центры являются одним из наиболее важных способов реагирования и удовлетворения потребностей клиентов. Тем не менее, спрос на качественное обслуживание клиентов сопровождается собственным набором операционных проблем. Многие компании начинают использовать облачные технологии и искусственный интеллект, чтобы лучше и эффективнее работать с клиентами. Рассмотрим некоторые методы взаимодействия конечного пользователя с системами искусственного интеллекта для контакт-центров.

I. ВИРТУАЛЬНЫЙ ОПЕРАТОР

Это автоматизация живого человека - людям отвечает не предзаписанный IVR, а голосовой помощник вместо живого оператора. Используя распознавание речи, виртуальный оператор понимает проблему и намерение позвонившего клиента, выраженную в свободной или полусвободной форме. Это означает, что конечному пользователю не потребуется тратить лишнее время на выслушивание предзаписанных сообщений и в итоге он быстрее получит помощь.

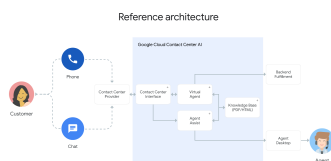


Рис. 1 – Архитектура взаимодействия виртуального оператора с клиентом и агентом

Виртуальный оператор также автоматизирует обработку базовых запросов и переводит разговор на живого оператора в более сложных случаях – все это благодаря распознаванию речи в реальном времени.

II. АССИСТЕНТ «ЖИВОГО» ОПЕРАТОРА

Бот будет «слушать» разговор двух людей и помогать оператору, выдавая советы и ссылки по ходу беседы. Оператор может использовать эти материалы по своему усмотрению: подглядывать в статьи из базы знаний самому либо де-

литься цитатами или ссылками с позвонившим клиентом и так далее. Планируется, что такая инфоподдержка поддержит оператора в любой ситуации.

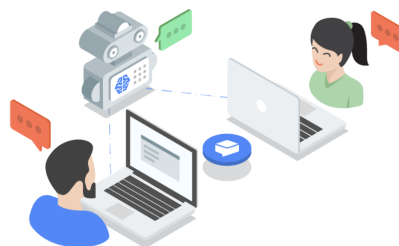


Рис. 2 – Схема использования ассистента "живого" оператора

У ассистента есть несколько режимов работы: бот рекомендует ссылки на статьи, бот рекомендует несколько вариантов ответа на один вопрос - варианты генерируются на основе предиктивного анализа, бот рекомендует ответы на часто задаваемые вопросы, подсказки в духе «как бы бот ответил сам» и бот дает наиболее релевантные подсказки всех типов

III. ВЫВОДЫ

Подобные решения обеспечивают персональную поддержку, немедленное обслуживание и быстрое решение проблем, превращая отдельные транзакции в постоянную лояльность клиентов. Улучшаются показатели отклонения вызовов, сокращается время обработки и затраты на обучение операторов, одновременно делая общие операции более быстрыми и эффективными. Агенты поддержки получают больше времени, чтобы позаботиться о более сложных и специализированных звонках. Автоматизируя более обыденные задачи и предоставляя агентам информацию в реальном времени, рабочие процессы и пошаговые инструкции, искусственный интеллект позволяет им предоставлять более специализированное обслуживание клиентов.

1. Люгер, Дж. О. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж.О. Люгер. // Информатика. – 2016. – №2. – С. 734-741.

Пархамович Александр Сергеевич, магистрант каф. ИТАС, alex.parkhamovich@gmail.com.

Научный руководитель: Никольшин Борис Викторович, заведующий кафедрой ЭВМ БГУИР, кандидат технических наук, доцент, nik@bsuir.by.

АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ ИГР

ВВЕДЕНИЕ

Ручное тестирование является дорогим как по затрачиваемому времени, так и по финансовым ресурсам. Самые продвинутые команды разработки приложений создают тысячи строк тестов для своих приложений: строка за строкой набирая одно и тоже «нажмите сюда» и «проверьте то». Такой подход имеет много недостатков, в том числе в плане финансирования команды тестирования. Создание таких тестов отвлекает внимание разработчиков от основной цели, над которой они работают – самого продукта. На смену ручному тестированию приходит процесс автоматизации тестирования.

I. Достоинства и недостатки АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ

Автоматизация тестирования — часть процесса тестирования с использованием программных средств для выполнения тестов и проверки результатов. Главная цель автоматизации — ускорить процесс тестирования без потери качества, а также избавить ручного тестировщика от постоянного прохождения регрессионных сценариев, позволив сфокусироваться на проверке бизнес-логики приложения с точки зрения конечного пользователя.

Преимущества автоматизации тестирования:

- исключен человеческий фактор;
- высокая скорость выполнения;
- генерируемые отчеты о результатах тестирования;
- выполнение тестов в удобное время.

Недостатки автоматизированного тестирования:

- сложность в написании тестов;
- нестабильная работа многих инструментов;
- множество отличных друг от друга компонентов интерфейса.

II. ОБЗОР ПОДХОДОВ РАБОТЫ С АВТОМАТИЗИРОВАННЫМ ТЕСТИРОВАНИЕМ

При автоматизации тестирования мобильных приложений обычно используется один из следующих подходов:

1. Record and Play

Потапчик Валерий Сергеевич, магистрант кафедры информационных технологий автоматизированных систем, potapchik97@gmail.com.

Научный руководитель: Сердюков Роман Евгеньевич, кандидат технических наук, rserdyukov@gmail.com.

Подход тестирования сводится к записи всех действий тестировщика в приложении. После записи действий инструмент генерирует свой специфический код и на основании него создает автотесты.

Достоинства:

- простая и быстрая реализация;
 - не требуются знания программирования.
- Недостатки:

- большинство изменений в приложении приводят к необходимости создания нового автотеста.

2. Screen Object

Подход, предназначенный для организации архитектуры автотестов в виде взаимодействия экранов приложения. Screen Object моделирует экраны тестируемого приложения в качестве объектов в коде. В результате получается набор классов, каждый из которых отвечает за работу со своим экраном приложения.

Достоинства:

- возможность переиспользовать код;
 - надежность кода, низкая чувствительность к изменениям в структуре приложения;
- Недостатки:

- требует знания языков программирования;
- низкая скорость разработки.

Каждый из этих подходов обладает достоинствами и недостатками, описанными выше, однако следует отметить, что наибольшую популярность приобрел подход Record And Play, в связи с низким порогом вхождения.

III. ВЫВОДЫ

Внедрение автоматизированного тестирования позволяет снять с команды мануального тестирования достаточно объемную часть регрессионных прогонов. Следует помнить, что ни полная автоматизация, ни стопроцентное ручное тестирование не дадут идеального результата. Оптимальным подходом в данном случае является разумный баланс. Использование связки автоматизированного и ручного тестирования и тесное взаимодействие с командой разработчиков от начала процесса работы над проектом до его реализации повысит качество конечного продукта.

АЛГОРИТМЫ РАСПОЗНАВАНИЯ ОБРАЗОВ НА ОСНОВЕ НЕЙРОННОЙ СЕТИ

Рассматриваются алгоритмы применения сверточных нейронных сетей для распознавания изображений

ВВЕДЕНИЕ

Данная работа посвящена распознаванию изображений коллекции CIFAR-10, с помощью искусственных нейронных сетей, с использованием гистограммы ориентированных градиентов и полутоновой матрицы смежности в качестве дополнительных источников признаков, а также тестированию и сравнению методов распознавания. Предложен подход к предобучению сверточных нейронных сетей. Указанная тема выбрана в связи с актуальностью задач распознавания изображений и задач компьютерного зрения в целом.

I. СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ

Сверточной нейронной сетью называют нейросеть со следующей архитектурой, удобно визуализируемой в трехмерном пространстве (см.рис.1.):

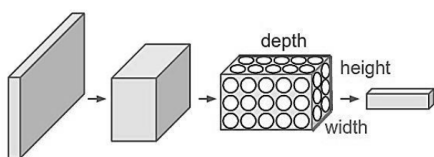


Рис. 1 – Архитектура Сверточной нейронной сети

В качестве входа каждый слой сети принимает трехмерный тензор, на практике чаще всего являющийся изображением.

Выходом сверточной сети является большой набор преобразованных данных, содержащих свойства входного изображения, но не конечный результат задачи аппроксимации. Таким образом, CNN является вспомогательным инструментом. Часто после CNN ставится MLP (многослойный перцептрон), который производит оценку извлеченных свойств.

II. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ

Сравнению были подвергнуты следующие модели:

1) MLP с тремя скрытыми слоями по 300, 200 и 50 нейронов соответственно.

Родионов Вадим Сергеевич, магистрант кафедры вычислительных методов и программирования БГУИР, v.rodionov997@gmail.com.

Научный руководитель: Матвеевко Владимир Владимирович, кандидат физ.-мат. наук, доцент, vladimir66@bsuir.by.

2) Сверточная нейронная сеть с тремя слоями свертки и одним полносвязным слоем из 200 нейронов. Параметры слоев свертки: 100 фильтров (3x3), 100 фильтров(2x2), 150 фильтров(2x2).

3) Модифицированная версия сети из п.2, использующая следующий алгоритм построения и обучения: обучается CNN из п.2. стандартным образом, к выходу 3го слоя свертки подключается полносвязный слой, принимающий также HOG и GLCM, полученная модель применяется к исходной тестовой выборке, формируя новую, производится предобучение MLP с тремя скрытыми слоями по 300, 100 и 100 нейронов на новой выборке. MLP подключается к выходу CNN.

4) Версия сети из п.3, предобучающая только первый скрытый слой.

Таблица 1 – Сравнение моделей

№	Ошибка на обучающей выборке	Ошибка на валидационной выборке	Точность на валидационной выборке	Точность на тестовой выборке
1	0.0412	0.0535	0.6635	0.6521
2	0.0313	0.0383	0.7315	0.7234
3	0.235	0.313	0.1034	0.0967
4	0.0023	0.0421	0.7438	0.7456

III. ВЫВОДЫ

Наибольшую эффективность показала сеть №4, наихудшую – сеть №3. Результаты, достигнутые моделями №4 и №2, отличаются незначительно, что может говорить о невысоком содержании дополнительных данных в HOG и GLCM. Предобучение для сети №3 оказалось неэффективным и только ухудшило результат. Возможно, это связано с тем, что данная задача требует более специфического подхода к предобучению.

1. Stanford CS class “CS231n: Convolutional Neural Networks for Visual Recognition” [Electronic resource] / Fei-Fei Li, A. Karpathy. – Mode of access: <http://cs231n.stanford.edu/>.

МОДИФИКАЦИИ АЛГОРИТМА RAFT В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Рассматривается алгоритм нахождения консенсуса в распределенных системах (Raft), а также его возможные способы улучшения для применения в реальных распределенных системах.

ВВЕДЕНИЕ

Алгоритм Raft – это алгоритм достижения консенсуса в распределенных системах, является результатом научной работы Диего Онгаро (Ph.D. студента Стэнфордского института), целью которой было построение легкого для понимания алгоритма: алгоритм должен не просто работать, но должно быть очевидно, как именно он работает.

В данной работе основное внимание уделяется репликации самих событий и тесно связанному механизму выбора лидера или определения кворума. С другой стороны, пренебрегают важными окружающими темами, такими как действия по обслуживанию кластера или «дела об окружающей среде», которые в реальном мире являются не менее важными компонентами при создании полного решения для высокодоступной системы.

I. МОДИФИКАЦИИ АЛГОРИТМА

Возможны и желательны три модификации алгоритма, которые должны быть реализованы в реальных распределенных системах:

1. Введение универсального уникального идентификатора сервера: чтобы определить, является ли слепок данных или журнал на одном сервере каким-либо состоянием конечного автомата того же конечного автомата на других серверах кластера, или является ли это снимком какого-то совершенно другого состояния машины, которая не имеет ничего общего с этим кластером.
2. Алгоритм преждевременного голосования.
3. Решение проблемы конкурирующих лидеров: основываясь на алгоритме преждевременного голосования, модифицируется Raft, чтобы отклонять сервера, которые пытаются выбрать себя в качестве лидера, если текущий лидер здоров для остальной части кластера.

В работе [1] была представлена идея алгоритма преждевременного голосования, без подробного объяснения деталей такого алгоритма.

Полезность алгоритма преждевременного голосования заключается в решении проблемы гонки выборов за лидерство при голосовании.

Алгоритм Raft обладает сильным свойством, которое заставляет его участников всегда принимать самый большой term, который они получили от другого сервера. Это свойство является ключом к надежности алгоритма: из-за этого выборы становятся детерминированными, и от этого также зависит свойство сопоставления журнала. Недостаток в реалиях реального мира состоит в том, что это легко приводит к ненужной «переизбытку терма». Предполагая, что сервера будут использовать 64-битную целочисленную систему, вряд ли у них закончатся числа в течение срока службы кластера, но кластера действительно будут подвержены поведению, при котором неисправный сервер будет начинать выборы при присоединении к кластеру, даже если остальные сервера были здоровы и продолжают иметь полноценного лидера.

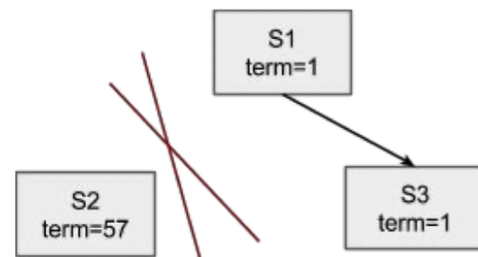


Рис. 1 – Схема Raft кластера состоящая из 3 серверов. Сервер 2 временно отключен от остальных.

- 1) Raft заставляет отключенный сервер начать выборы один раз в каждом тайм-ауте, что приводит к увеличению текущего term. Поскольку он не может подключиться к другим серверам, он потеряет свой выбор и производит повторную попытку.
- 2) Как только он сможет восстановить соединение с остальной частью кластера, его более высокий term будет распространяться на S1 и S3. Это приведет к тому, что S3 прекратит принимать новые записи журнала от S1, и заставит S1 уйти в отставку в качестве лидера.
- 3) Новые выборы на term = 58 в конечном счете инициированы и будут выиграны тем сервером, который успеет первым.

Решение состоит в том, чтобы ввести алгоритм предварительного голосования, который

выполняется перед переходом в статус кандидата. Сервер может переключиться на кандидата только в том случае, если превентивное задание выполнено успешно, в противном случае он должен дождаться следующего тайм-аута выборов.

Реализация алгоритма предварительного голосования проста: получатели RPC PreVote должны отвечать с тем же результатом, что и, если бы это был фактический голос. Тем не менее, важно подчеркнуть, что предварительный ответ не является обязательным для сервера. Хотя каждый сервер должен голосовать только за одного кандидата на определенный срок, это не относится к этапу предварительного голосования. Несколько кандидатов могут получить положительные показания на этапе предварительного голосования, однако, как только они приступят к реальным выборам, только один из них получит реальное голос. Такое поведение яв-

ляется ключом к тому, чтобы избежать гоночных условий, которые могут привести к неудачным выборам. Например, сервер может преуспеть в получении потенциального большинства голосов на этапе предварительного голосования, но затем сам отключится, прежде чем сможет приступить к фактическим выборам. В этом случае было бы не целесообразно тратить драгоценное времени на отказ при голосовании за другого кандидата, который все еще может победить на выборах.

1. Ongaro Diego; Ousterhout John "In Search of an Understandable Consensus Algorithm". –2014
2. Ongaro, Diego. "Consensus: Bridging Theory and Practice". –2014
3. Brian M. Oki; Barbara H. Liskov "A New Primary Copy Method to Support Highly Available Distributed Systems". –1998

Романович Евгений Александрович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, yauheni.ramanovich@gmail.com.

Научный руководитель: Матвеевко Владимир Владимирович, доцент, кандидат физ.-мат. наук vladimir66@bsuir.by.

АЛГОРИТМЫ ДЛЯ АНАЛИЗА ДАННЫХ ДНК

Рассматриваются алгоритмы анализа данных, применяемых к строковым представлениям ДНК для определения их свойств и функций.

ВВЕДЕНИЕ

В настоящее время было создано достаточно много алгоритмов, направленных на изучение свойств и структуры последовательностей ДНК. Большинство этих алгоритмов применяют стандартную технику теорий вероятности и математической статистики для исследования свойств в строении ДНК. Основная задача, которая встает перед исследователем, получившим новую последовательность – определить, что это за последовательность, каковы ее функции и свойства. Наиболее простым способом является сравнение новой последовательности с уже известными последовательностями.

I. СТАТИСТИЧЕСКИЙ АЛГОРИТМ ПОИСКА ПОДОБИЙ МЕЖДУ ПОСЛЕДОВАТЕЛЬНОСТЯМИ

Для последовательностей S_1 и S_2 длины которых соответственно равны N_1 и N_2 строится прямоугольная матрица размером $N_1 \times N_2$. В верхней строке и левом столбце которой записываются сами последовательности. В (i, j) позиции матрицы ставится точка, если i и j символы последовательностей S_1 и S_2 совпадают. Затем выделяются группы точек, расположенных на линии, параллельной диагонали. Полученные таким образом отрезки определяют схожие участки двух последовательностей, причем число точек на отрезке равно длине фрагментов. Этот метод очень прост и нагляден, однако его существенным недостатком является слишком большая избыточность, что затрудняет анализ полученной картины и увеличивает время обработки.

II. АЛГОРИТМ СКОЛЬЗЯЩЕЙ РАМКИ

На каждом шаге алгоритма последовательность S_1 сдвигается относительно S_2 на 1 основание(символ). Общая часть этих последовательностей сканируется окном длиной W и определяется число совпадающих символов. Если это число больше некоторого K , считается, что совпадение найдено. Число операций, выполняемых алгоритмом – $N_1 \times N_2 \times W$ достаточно велико. Поэтому на практике применяют его модификацию, позволяющую сократить трудоемкость в W раз. При этом учитывается тот факт, что при сдвиге на одно основание количество совпадающих букв

изменяется только за счет граничных символов. Таким образом, полный пересчет совпадений на каждой итерации не осуществляется.

III. АЛГОРИТМ НАХОЖДЕНИЯ ОПТИМАЛЬНОГО ВЫРАВНИВАНИЯ

Существенным недостатком статистических алгоритмов является то, что они не учитывают наличие погрешностей в сравниваемых последовательностях. Поэтому их точность в некоторых случаях весьма невысока.

Для учета погрешностей применяется алгоритм нахождения оптимального выравнивания. Определяется функция сходства F , которая учитывала не только число совпадений, но также и ошибки преобразования ДНК в строковую последовательность. В качестве параметров функции вводятся веса, увеличивающие функцию F при обнаружении совпадения и штрафы за погрешности, такие как замены и вставки.

$$F = K_m * V_m - K_d * V_d - K_c * V_c,$$

где K_m, K_d, K_c - количество совпадений, вставок и замен, V_m, V_d, V_c - параметры, характеризующие веса совпадений, вставок и замен соответственно.

Поиск оптимального значения функции сходства осуществляется в два этапа. На первом - для двух сравниваемых последовательностей строится точечная матрица совпадений. Движение по матрице вдоль диагонали через пустые клетки соответствует заменам, через заполненные - совпадениям, движения вверх или вправо - вставкам. Выравнивание представляет из себя путь из правого верхнего угла точечной матрицы совпадений в левый нижний. При этом необходимо, чтобы этот путь соответствовал максимуму функции сходства. Задача о поиске оптимального пути решается методами динамического программирования. Для построения оптимального пути заполняется матрица наилучших значений функции, по которой затем восстанавливается весь путь. Наиболее удачными реализациями алгоритма оптимального выравнивания являются программные реализации FASTA и BLAST.

1. Франк-Коменецкий, М. Д. Компьютерный анализ генетических текстов // Москва, Наука. – 1990.

Савик Олег Владимирович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, oleg.savik1996@gmail.com.

Научный руководитель: Кургулев Александр Петрович, профессор, кандидат технических наук, kaftoe@bsuir.by.

МЕТОДЫ ОЦЕНКИ ГИБКОСТИ АРХИТЕКТУРЫ ПРИЛОЖЕНИЙ

Рассматриваются некоторые принципы построения архитектуры приложения и исследуется их влияние на программный код, а также методы, которыми можно измерить полезность этого влияния.

ВВЕДЕНИЕ

Существуют различные принципы для построения архитектуры приложения. В основе них лежит опыт ведущих специалистов в области разработки программного обеспечения. Исследованы примеры, с целью найти методы оценки этих принципов.

I. ПРИНЦИП ПОДСТАНОВКИ БАРБАРЫ ЛИСКОВ

Пусть $q(x)$ является свойством, верным относительно объектов x некоторого типа T . Тогда $q(y)$ также должно быть верным для объектов y типа S , где S является подтипом типа T .

В приложении есть обработка запросов на доставку еды из магазина. Строка запроса: "store=(smth)/menu=(smth)/address=(smth)". Есть обработчик, настроенный на приём запроса такого формата. Назовём формат базовым типом. Запрос для любого магазина должен наследовать тип базового. Однако, один из разработчиков при добавлении нового магазина изменил в строке запроса *menu* на *food*. и намерен использовать тот же обработчик. Однако теперь этот запрос не может быть подтипом базового запроса, т.к. он нарушает принцип подстановки (см.рис.1). Теперь для корректной работы программы нам необходимо добавить условие *if(store = 'other')*. Вывод: при невыполнении данного принципа нам необходимо добавить по одному лишнему условию для каждого неправильного подтипа. Также есть нарушение другого принципа - открытости/закрытости. Обработчик, использующий иерархию классов с нарушениями принципа Лисков, помимо оперирования ссылкой на базовый класс, также вынужден знать и о подклассе.

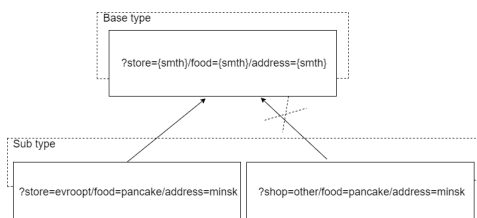


Рис. 1 – Иерархия наследования

Слука Роман Янушевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, roma.sluka.97@mail.ru.

Научный руководитель: Таранчук Валерий Борисович, профессор кафедры компьютерных технологий и систем ФПМИ БГУ, доктор физико-математических наук, taranchuk@bsu.by.

II. ПРИНЦИП УСТОЙЧИВЫХ ЗАВИСИМОСТЕЙ

Зависимости должны быть направлены в сторону устойчивости [1].

Компоненты в системе должны быть разной метрики устойчивости. Иначе, если все компоненты будут устойчивы, такую систему тяжело будет изменить. Устойчивость можно измерить по количеству входящих и выходящих из компонента зависимостей по формуле $I = \frac{out}{in+out}$. Где *out* - число выходящих зависимостей, *in* - входящих (см.рис.2.).

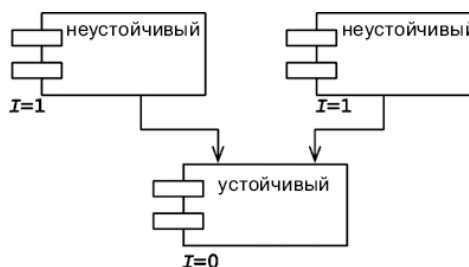


Рис. 2 – Система с тремя компонентами

При нарушении принципа компоненты ниже, будут иметь метрику устойчивости больше чем у верхних. Тогда изменение нижних будет усложнено и должно будет согласовываться со всеми верхними компонентами.

III. ВЫВОДЫ

Были найдены методы оценки принципов, как они влияют на изменение программного кода. На основе исследований можно создать статический анализатор кода, контролирующий соблюдение принципов.

1. Чистая архитектура. Искусство разработки программного обеспечения./ Р. Мартин. – СПб.: Питер, 2019. – 352 с.

КРОСС-ВАЛИДАЦИЯ ФИНАНСОВЫХ МОДЕЛЕЙ

Описаны основные задачи кросс-валидации, проблемы её применения для задач финансов и адаптированные методы, решающие эти проблемы

ВВЕДЕНИЕ

У подавляющего большинства практикующих аналитиков рынков и авторов торговых стратегий, основным показателем эффективности торговых моделей является демонстрация её результатов на исторической симуляции. Однако, высокая эффективность на тестовой исторической выборке легко достигается за счёт тестирования альтернативных конфигураций торговой стратегии. Переобученная модель разочарует при использовании, ведь доходы окажутся меньше ожидаемых, сравнивая с тестовой выборкой. По этой причине, переобучение можно считать одной из основных причин провала отдельных торговых стратегий и целых фондов.[1] Чтобы избежать переобучения, специалисты по машинному обучению применяют кросс-валидацию. Простейшая кросс-валидация - это дробление наблюдений на два подмножества: тренировочное и тестовое. Каждое наблюдение в полной совокупности данных принадлежит одному и только одному подмножеству[2]. Это сделано для того, чтобы избегать утечки из одного подмножества в другое. Существует ряд альтернативных проверок, из которых одной из самых популярных является k-блочная кросс-валидация[3]. На Рисунке 1 изображён принцип 4-блочной кросс-валидации.



Рис. 1 – 4-блочная кросс-валидация

На Рисунке 2 изображён пример как выглядит 4-блочная кросс-валидация для ценового ряда.

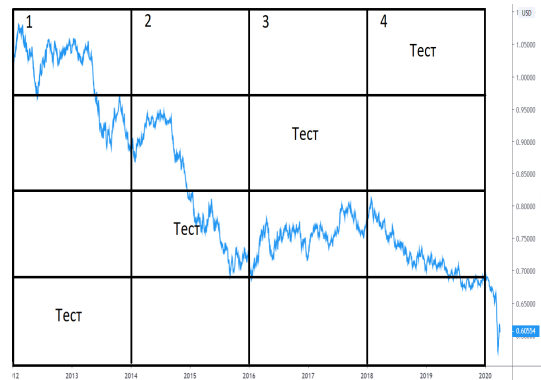


Рис. 2 – 4-блочная кросс-валидация на примере ценового ряда

I. ПРОБЛЕМЫ КРОСС-ВАЛИДАЦИИ В ФИНАНСАХ

Применение k-блочной перекрёстной проверки оказывается безуспешной в финансах из-за двух особенностей.

Первая проблема: наблюдения не берутся из взаимно независимого случайного процесса, т.е. в один момент времени может быть открыто несколько торговых позиций, а т.к. решения по этим торговым позициям зависят от одних и тех же участков ценового ряда, эти наблюдения взаимозависимы. Таким образом возникает систематическая утечка данных. Проблема утечки или заглядывания в будущее тоже является распространённой и очень опасной проблемой при тестировании торговых стратегий.

Вторая проблема: постоянное использование одних и тех же тестовых подмножеств в процессе работы приводит к систематическому переобучению. Поэтому возникли специфичные для финансовой сферы методы кросс-валидации.

II. ПРОЧИЩЕННАЯ КРОСС-ВАЛИДАЦИЯ

Чтобы справиться с проблемой взаимно зависимых наблюдений, можно применить метод очищенной k-блочной перекрёстной проверки.[4] Метод заключается в том, чтобы удалить из тренировочного подмножества те наблюдения, временные метки которых накладываются по времени на метки наблюдений из тестового подмножества. Удалённые области помечены на Рисунке 3 жёлтым цветом. Также исключаем из тренировочного подмножества наблюдения, которые хронологически следуют сразу за наблюдениями тестового подмножества, чтобы уменьшить влияние внутрирядовой корреляции. Об-

ласть с высокой внутрирядовой корреляцией показана на Рисунке 3 красным цветом.

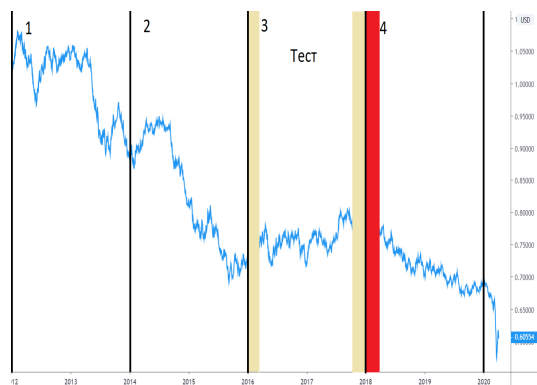


Рис. 3 – Пример прочищенной кросс-валидации

III. КОМБИНАТОРНАЯ КРОСС-ВАЛИДАЦИЯ

Уменьшить влияние второй проблемы можно, если увеличить количество траекторий, по которым можно проводить кросс-валидацию. Метод комбинаторной кросс-валидации позволяет получить больше траекторий из тех же данных. [5] Разделим наблюдение на N групп, указав, что k из них будут тестовыми группами. Тогда количество возможных сочетаний подмножеств равно.

$$\binom{N}{N-k} = \frac{\prod_{i=0}^{k-1} (N-i)}{k!} \quad (1)$$

Тогда количество уникальных траекторий на этих сочетаниях равно.

$$\phi[N, k] = \frac{k}{N} \binom{N}{N-k} = \frac{\prod_{i=1}^{k-1} (N-i)}{(k-1)!} \quad (2)$$

Таким образом, поделив данные на 10 групп, 2 из которых будут тестовыми, получим $\frac{2}{10} \binom{10}{8} = 9$ траекторий. Ещё одной полезной особенностью

Сочинко Андрей Викторович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, andreysoch@gmail.com.

Научный руководитель: Шилин Леонид Юрьевич, декан факультета информационных технологий и управления БГУИР, доктор технических наук, профессор, dekfitu@bsuir.by.

этого решения является возможность подсчитать вероятность, что проверяемая модель переобучена.

IV. ВЫВОДЫ

Используя решения проблем кросс-валидации в финансах, которые предложены в описанных методах, можно адаптировать k -блочную и менее популярные методы кросс-валидации под финансовые задачи. Методы гибкие: их можно применять как по отдельности, так и в связке. Эти решение отлично дополняются тестированием на синтетических, сгенерированных данных, которые соответствуют разным потенциальным сценариям движения цен. Используя комбинаторную кросс-валидацию появляется возможность подсчитывать вероятность переобучения, а значит можно отбирать модели с низким значением этой вероятности.

1. The 10 Reasons Most Machine Learning Funds Fail / Marcos Lopez de Prado // Journal of Portfolio Management, Forthcoming. – 2018.
2. Python Machine Learning / Sebastian Raschka // Packt Publishing (Birmingham-Mumbai). – 2015, – С. 173-176.
3. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Second Edition) / Trevor Hastie, Robert Tibshirani, Jerome Friedman // Springer Series in Statistics. – 2008, – С. 247.
4. The Probability of Backtest Overfitting / David H. Bailey, Jonathan Borwein, Marcos Lopez de Prado, Qiji Jim Zhu // Journal of Computational Finance (Risk Journals). – 2015, Forthcoming – С. 1-34.
5. Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance / David H. Bailey, Jonathan Borwein, Marcos Lopez de Prado, Qiji Jim Zhu // Notices of the American Mathematical Society, 61(5), 2014, С. 458-471.

ПРИНЯТИЕ РЕШЕНИЙ В ЗАДАЧАХ ЛОГИЧЕСКОГО ВЫВОДА

Рассматривается использование методов принятия решений в задачах оптимизации. Предлагается использование генетического алгоритма и метода Саати.

ВВЕДЕНИЕ

Не редки ситуации, когда достижение некоторого результата может быть осуществлено не единственным способом. В таком случае встает вопрос о поиске наилучшего решения. На практике оказывается, что в большинстве случаев понятие «наилучший» может быть выражено количественными критериями, поэтому возможна постановка математических задач поиска оптимального результата. Такие задачи называются задачами оптимизации.

I. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искоемых параметров с использованием механизмов, аналогичных естественному отбору в природе [1]. Идея генетических алгоритмов заимствована у живой природы и состоит в организации эволюционного процесса, конечной целью которого является получение оптимального решения в сложной комбинаторной задаче. Разработчик генетических алгоритмов выступает в данном случае как «создатель», который должен правильно установить законы эволюции, чтобы достичь желаемой цели как можно быстрее. На этапе отбора встает вопрос «отсеивания» наиболее приспособленных особей. В предлагаемой нами модификации генетического алгоритма для этого используется метод анализа иерархий Саати.

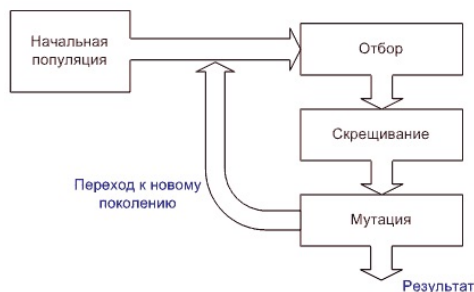


Рис. 1 – Алгоритм работы классического ГА

II. МЕТОД АНАЛИЗА ИЕРАРХИЙ

Метод Анализа Иерархий – математический инструмент системного подхода к сложным проблемам принятия решений. Он не предписывает лицу, принимающему решение, какого-либо «правильного» решения, а позволяет ему в интерактивном режиме найти такой вариант, который наилучшим образом согласуется с его пониманием сути проблемы и требованиями к ее решению. Главным образом это достигается путем определения критериев и их «веса». Продолжая параллель с природой, на этапе отбора особей мы могли бы определить такие критерии как «размер», «скрытность» или даже «умение добывать пищу» и выставить им субъективный вес. Пусть целью является достижение максимальной вероятности выживания. Очевидно, умение добывать пищу получит наибольший вес, в то время как, например, размер особи вряд ли имеет большое значение для выживания. Исходя из поставленной цели и критериев мы можем отобрать наиболее приспособленные особи и продолжить выполнение алгоритма, пока не достигнем оптимальной вероятности выживания.

III. ВЫВОДЫ

Предлагаемая модификация генетического алгоритма совмещает отчасти хаотичный генетический алгоритм с гибким, но достаточно строгим методом Саати, что позволяет принимающему решению лицу четко определить, какой именно результат требуется от полученной модели: акцентировать ли внимание на определенном критерии или найти решение, которое максимально приближенно к оптимальному результату целевой функции, пусть и с некоторыми упущениями по другим параметрам.

1. Генетические Алгоритмы / Л. А. Гладков. – СПб.: Физматлит, 2010. – 320 с.

Ставер Дмитрий Николаевич, магистрант факультета информационных технологий и управления БГУИР, dnstaver@gmail.com.

Научный руководитель: Герман Олег Витольдович, кандидат технических наук, доцент, ovgerman@tut.by.

COMPARATIVE ANALYSIS OF MACHINE AND DEEP LEARNING ALGORITHMS IN SEMANTIC ANALYSIS

The article compare different approaches as machine learning and deep learning in semantic analysis of phone reviews in Russian.

INTRODUCTION

Nowadays, when you can get or lose millions on the market just in one second, it's extremely important to receive feedback for all your actions and products. But you can have no time to read every message about your new phone model. That is a time when sentiment analysis come to the spotlight and help you to see general report of (in our case) feedbacks.

As Indicator of correct work of algorithm has been chosen late submission for Kaggle competition [1] with accuracy score as a result. Since vocabulary of example reviews is connected with phones and provided in Russian, reviews from site 4pda.ru has been chosen as initial dataset for training of our models.

I. DATA QUALITY ASSESSMENT

The main challenge of data cleaning in our case is that reviews are marked from 1 to 10 while Kaggle competition divides them only as positive or negative. The answer to the question "what mark is the highest negative?" is one of hyperparameters of the algorithm. After checking the balance of marks in the dataset (Fig. 1), exploring text of the reviews and testing with different values, it was found, that people usually write negative reviews with marks below 8. This value was chosen as a division into positive and negative reviews.

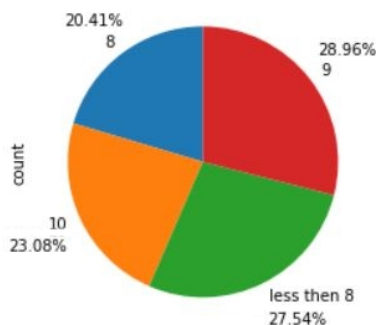


Рис. 1 – The balance of marks in dataset

The next challenge was to clean the reviews from punctuation and other special symbols. After

cleaning the data the accuracy on train data of some algorithms has raised from 0.4 to 0.7.

II. MODELS AND ALGORITHMS

In sentiment analysis there is at least two different approaches: classic Machine Learning and Neural Network. We've tried both. As ML algorithm has been chosen implemented in python library sklearn tf-idf due to its best performance on our train data and on the Sentiment field in total, with LinearSVC for the same reason. After greed search has been chosen default parameters. Sklearn shows good performance and that is easy to build ML algorithms with that wide choice of already implemented functions in it.

As NN algorithm has been chosen LSTM Neural Network with 2 layers and size of hidden dimension equal to 512 implemented on PyTorch due to ability of it to remember context of current word. PyTorch is a powerful Deep Learning library for python that allows us to build networks in easy way, but still have an opportunity to tune a lot of parameters and structure of the model.

III. FEATURE ENGINEERING

Since tf-idf requires only sequence of words as input feature, the only necessary thing with train and test dataset was cleaning data from extra symbols.

For LSTM NN due it was necessary to transform the reviews to arrays of encoded words with the same length of sentence. For that we transform reviews to array of words and after building vocabulary, encode each word with ID.

IV. PERFORMANCE AND CONCLUSION

For Machine Learning Algorithm has been achieved 0.94 accuracy on test dataset, when for Deep Learning Algorithm – 0.64 accuracy.

1. Kaggle Competition [Electronic resource] – Mode of access : <https://www.kaggle.com/c/product-reviews-sentiment-analysis/> – Date of access: 29.02.2020

Стародубец Андрей Сергеевич, студент 3-го курса ФИТиУ БГУИР, astarodubets@mail.ru
 Научный руководитель: Трофимович Алексей Фёдорович, старший преподаватель кафедры ИТАС, trofimaf@bsuir.by

ДИНАМИЧЕСКОЕ ПЕРЕРАСПРЕДЕЛЕНИЕ ЗАДАЧ В СИСТЕМАХ ВЗАИМОДЕЙСТВУЮЩИХ СЕРВИСОВ

Рассматриваются задачи оптимизации управления системами взаимодействующих сервисов. Предполагаемый метод их решения базируется на использовании проекции линейных задач о назначении или задач нескольких странствующих коммивояжеров на рекуррентные сети.

ВВЕДЕНИЕ

Задачи оптимизации управления системами взаимодействующих сервисов, являющиеся предметом активного исследования и обсуждения в последние годы, обычно формулируются в терминах задач о динамическом назначении. Практически всегда такие задачи сводятся к известным задачам дискретной оптимизации, таким как линейная задача о назначении или задача нескольких странствующих коммивояжеров. В общем случае возможно назначение сервису двух или более задач одновременно, но такой вариант формально легко свести к задаче назначения векторного ресурса одной сложной задаче и/или скаляризации критериев оптимизации.

I. ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

Цель исследования – разработка моделей и эффективных по быстродействию алгоритмов оперативного распределения потоков задач в системах взаимодействующих сервисов с централизованным и коллективным управлением.

Для достижения указанной цели необходимо решить следующие задачи [1]:

- 1) разработать модели и алгоритмы спецификации процессов управления взаимодействующими процессами, основанные на рекуррентных сетях;
- 2) исследовать методы оценки устойчивости решения задач об оптимальном назначении координируемых сервисов;
- 3) разработать методы самоорганизации процедур оптимизации управления на сетевых моделях с накоплением информации;
- 4) рассмотреть вопросы эффективной реализации алгоритмов управления взаимодействующими сервисами на основе распределенных вычислений.

II. МЕТОДЫ ПЕРЕРАСПРЕДЕЛЕНИЯ ЗАДАЧ

Реализация перераспределения задач требует наличия динамической модели процесса обслуживания, позволяющей выявлять и прогнозировать будущие события хотя бы на один шаг.

Тараскевич Максим Дмитриевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, maksim.taraskevich@mail.ru.

Научный руководитель: Ревотюк Михаил Павлович, кандидат технических наук, доцент, rmp@bsuir.by.

Известными примерами таких моделей являются диаграммы Ганта, модели потоков работ, сети Петри и их расширения, конечные автоматы, системы векторного сложения и многие другие. Формально такие модели эквивалентны системам рекуррентных уравнений разного порядка. Потребность отражения свойств параллельности и асинхронности процессов обслуживания сервисами определяет причину использования для решения задач координации сервисов понятия рекуррентной сети.

Основная концепция решения поставленных задач – конструктивное объединение методологии имитационного моделирования и оптимального управления для развития базирующихся на сетевых моделях методов построения систем управления дискретными процессами в распределенных системах с императивным характером поведения. Результат объединения – рекуррентная сеть второго порядка с поиском и сохранением альтернатив назначения.

III. ВЫВОДЫ

Рекуррентные сетевые модели второго порядка и алгоритмы динамического перераспределения задач рекомендуются для построения систем координации взаимодействующих сервисов как средство повышения эффективности назначения путем учета актуальной информации о фактическом состоянии системы в наиболее поздний момент. В любой момент времени предполагается наличие оценок интервалов устойчивости назначения задач сервисам. Определение таких оценок имеет полиномиальную вычислительную сложность.

Таким образом, оценки устойчивости назначения позволяют определить границы области изменения локальных критериев эффективности для быстрой классификации полезности новых альтернатив решения задач.

1. Ревотюк, М.П. Динамическое перераспределение мест размещения сервисов/М.П. Ревотюк, М.К. Кароли, Р.В. Кругликов//XIV Белорусско-российская научно-техн. конф. «Технические средства защиты информации» (25 мая 2016 г., Минск). – Минск: БГУИР, 2016. – С. 39

АНАЛИЗ МЕТОДОВ НАХОЖДЕНИЯ СКРЫТОГО ПРЕДСТАВЛЕНИЯ МЕДИАФАЙЛОВ

Рассматриваются основные методы нахождения скрытого представления медиафайлов, а также типы задач, для которых эти методы применяются.

ВВЕДЕНИЕ

В настоящее время актуально множество задач классификации и анализа медиафайлов. Для решения многих задач машинного обучения требуется большой тренировочный набор размеченных данных, на котором нужно обучать модель, потратив значительное время и вычислительные ресурсы.

Чтобы избежать повторного обучения моделей для каждой отдельной задачи, существуют способы нахождения скрытого представления медиафайлов. Они позволяют кодировать данные в универсальном виде, который можно использовать для решения, например, не одной конкретной задачи классификации, а нескольких без переобучения всей модели.

I. АРХИТЕКТУРЫ НЕЙРОННЫХ СЕТЕЙ В РЕШЕНИИ ЗАДАЧ МАШИННОГО ОБУЧЕНИЯ

Одним из способов получения скрытого представления является построение автокодировщиков на различных архитектурах нейронных сетей.

Выделяются виды нейронных сетей:

- сверточные;
- рекуррентные;
- трансформеры;
- wavenet.

Разные реализации используются для работы в основном с определенными типами данных. Одна из главных областей применения рекуррентных нейронных сетей – работа с языковыми моделями, в частности – анализ контекста и общей связи слов в тексте. В последнее время эти сети начали занимать нишу машинного перевода. Одним из крупнейших примеров является использование компанией Google рекуррентных нейронных сетей типа LSTM в системе Google Translate [1]. Однако они также применяются для кодирования и последующей генерации звука [2]. Сверточные сети широко применяются для всех видов сигналов. Wavenet используются в основном для синтеза речи [3]. А трансформеры были изначально разработаны для машинного перевода и постепенно заменили рекур-

рентные нейронные сети в основных задачах нейролингвистического программирования [4]. Примеров применения данных нейронных сетей для обработки звука не так много (можно выделить Magenta – исследовательский проект Google AI – как один из наиболее значимых проектов, связанных с созданием музыки).

На сегодняшний день чаще всего применяются сверточные, рекуррентные и wavenet нейронные сети. В перспективе трансформеры могут позволить добиться более высокого качества обработки звука, хотя сейчас они остаются менее эффективными с точки зрения скорости вычислений.

II. ВЫВОДЫ

Построение автокодировщиков с различными архитектурами и определение результатов их работы при условии одинакового количества операций с плавающей запятой в секунду может ответить на вопрос о наиболее эффективном методе работы со звуковым сигналом при решении ряда задач машинного обучения.

1. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv preprint:1609.08144v2 [cs.CL], 2016.
2. Zhichao Zhang, Shugong Xu, Tianhao Qiao, Shunqing Zhang, Shan Cao, Attention based Convolutional Recurrent Neural Network for Environmental Sound Classification, arXiv preprint:1907.02230v1 [cs.SD], 2019.
3. Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, arXiv preprint:1609.03499v2 [cs.SD], 2016.
4. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, arXiv preprint:1706.03762v5 [cs.CL], 2017.

Царева Вероника Викторовна, магистрант кафедры информационных технологий и управления БГУИР, veronika.tsariova@gmail.com.

Научный руководитель: Ревотюк Михаил Павлович, кандидат технических наук, доцент, rmp@bsuir.by.

ПРОБЛЕМЫ СОВРЕМЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ НА МОБИЛЬНЫХ УСТРОЙСТВАХ

Рассматриваются основные проблемы современных веб-приложений на мобильных платформах. Предлагается использование веб-приложения PWA. Progressive Web Application — это современный способ создания быстрых веб-приложений с автономной поддержкой.

ВВЕДЕНИЕ

Одной из важных вех развития современных веб-технологий является факт того, что современные сайты превратились из коллекции связанных документов в полноценные приложения с широким набором функций и содержательных данных. Популярность доступа к сети Интернет с мобильных или планшетных устройств растет. Согласно статистике «StatCounter» (2019) [1], доля мобильного рынка превысила долю рынка настольных компьютеров и в настоящее время составляет 51,74% по всему миру. В Беларуси эта цифра несколько отличается: 25,59% мобильного трафика [2]. Кроме того, за последние несколько лет, благодаря технологическому прогрессу и снижению цен на аппаратное обеспечение, улучшилась скорость подключения к сети и производительность мобильных устройств.

I. МОДИФИЦИРОВАННЫЙ PUF ТИПА АРБИТР

Несмотря на эти достижения, во многих случаях ощущение скорости при просмотре веб-страниц на мобильных устройствах не возросло такими же темпами. Исследование, проведенное компанией «DoubleClick» (2016) [3] в сетях 3G, показало, что 77% мобильных сайтов требуют более 10 секунд для загрузки и что среднее время загрузки веб-страниц составляет 19 секунд. Данное эффект зачастую вызван постоянно растущей сложностью современных веб-приложений. Расширение возможностей пользователя, увеличение количества и качества визуального контента, такого как изображения и видео, и динамической функциональности с использованием JavaScript привело к росту размеров современных веб-страниц. Согласно данным HTTP Archive(2018a) [4], средний размер веб-страниц увеличился более чем на 260% с ноября 2010 года. Размеры JavaScript также увеличились более чем на 300%, а время загрузки страниц на мобильных устройствах почти на 160% с сентября 2011 года, согласно данным HTTP Archive(2018b). Как показано на рисунке 1, на мобильных устройствах время разбора и интерпретации могут занимать в 2–5 раз больше времени, чем на компьютере.

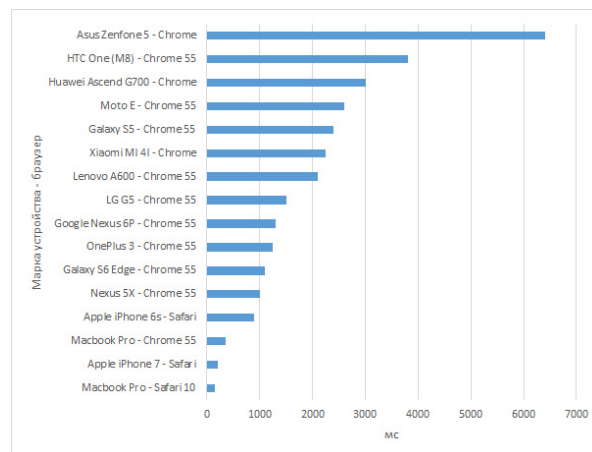


Рис. 1 – Время разбора 1 Мб JavaScript на различных устройствах

Чувство замедленности может выражаться в нескольких меньших или больших задержках перед отображением фактического контента на экране. Эти задержки, как правило, накапливаются и замедляют весь процесс для конечного пользователя. Плохой пользовательский опыт увеличивает процент отказов и снижает вовлеченность пользователей, что, в свою очередь, снижает доходы от рекламы и впечатления от продукта. Более того, скорость сетевого соединения не всегда стабильна. Это может быть вызвано изменениями состояния сетевого подключения, вызванными роумингом между ячейками сети или переходом от беспроводного соединения к мобильному соединению для передачи данных. Кроме того, физические факторы могут вызывать колебания скорости, например, вход в лифт или метро. В случае более медленных или ненадежных соединений важно расставлять приоритеты и минимизировать количество байт, передаваемых с сервера на устройство, для оптимизации просмотра веб-страниц. Самым важным фактором для создания ощущения быстрого просмотра веб-страниц является максимально быстрая визуализация содержимого. Это означает, что в первую очередь приоритет отдается изначально видимым частям веб-страницы. Остальное содержимое может быть загружено асинхронно после завершения начальной загрузки и рендеринга наиболее важного содержимого. Кроме того, не следует загружать на устройство контент или ресурсы, которые не используются или не видны пользователю.

II. МЕТРИКИ ПРОИЗВОДИТЕЛЬНОСТИ

Важность хорошей производительности на мобильных устройствах возрастает в связи с увеличением использования мобильных устройств вместо настольных компьютеров для доступа к веб-приложениям. Проблемы с производительностью веб-приложений можно разделить на две категории: передача данных и runtime. Под передачей данных подразумевается загрузка любых ресурсов, необходимых для работы приложения. Под runtime — работа приложения, рендеринг и обработка пользовательского ввода. Загрузка страницы:

- TTFB — время до получения первого байта (Time to First Byte).
- FCP — время первой отрисовки контента (First Contentful Paint).
- FMP — время первой значимой отрисовки (First Meaningful Paint).
- TTI — время до момента, когда страница может реагировать на пользовательский ввод (Time to interactive).

Runtime

- Время отклика на пользовательский ввод.
- Время перерисовки интерфейса.

III. PROGRESSIVE WEB APPLICATION

Многие компании уделяют больше внимания мобильной производительности для того, чтобы охватить большую аудиторию и обеспечить более быстрый доступ к веб-приложениям на мобильных устройствах. Одним из возможных решений для построения современных веб-приложений, которые хорошо работают даже в медленных соединениях и низкопроизводительных устройствах, является создание прогрессивного веб-приложения (Progressive Web Application - PWA). Веб-приложение PWA — это современный способ создания быстрых веб-приложений с автономной поддержкой. В будущем все больше и больше веб-приложений и веб-сайтов будут поддерживать автономное использование. Пользователи мобильных приложений

уже привыкли к тому, что могут хотя бы частично пользоваться приложениями без подключения к Интернету. Однако разница между мобильными приложениями и веб-приложениями на мобильных устройствах уменьшается в связи с растущей популярностью PWA. С развитием концепции PWA, веб-разработчики могут быстро создавать и развертывать приложения, которые выглядят и работают одинаково хорошо на всех платформах, в то время как пользователи приложений получают мгновенный доступ к необходимым веб-ресурсам в виде мобильных приложений.

IV. ВЫВОДЫ

Предлагаемая концепция разработки веб-приложения предоставляет возможность продолжать работу с приложением при сбоях в интернет-подключении, увеличенная скорость загрузки с помощью PWA обезопасит от увеличения количества отказов со стороны пользователя.

1. Статистика использования веб-приложений на различных устройствах в мире [Электронный ресурс]. – Режим доступа: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>. – Дата доступа: 20.03.2020.
2. Статистика использования веб-приложений на различных устройствах в Беларуси [Электронный ресурс]. – Режим доступа: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/belarus>. – Дата доступа: 20.03.2020.
3. Analyzing and Improving the Loading Performance of Large-scale Websites on Mobile Devices [Electronic resource]. – Mode of access: <https://www.semanticscholar.org/paper/Analyzing-and-Improving-the-Loading-Performance-of-Naapala/c2dc42131c55fa357c06062378cba0899e073ffa>. – Date of access: 20.01.2020.
4. Статистика веб-страниц [Электронный ресурс]. – Режим доступа: <https://httparchive.org/reports/state-of-the-web>. – Дата доступа: 02.02.2020
5. Хабр [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/tinkoff/blog/489230/>. – Дата доступа: 10.02.2020.

Циманович Анастасия Игоревна, магистрант кафедры информационных технологий автоматизированных систем БГУИР, nastya.cimanovich.97@gmail.com.

Научный руководитель: Захарьев Вадим Анатольевич, кандидат технических наук, доцент, zahariev@bsuir.by.

СОВРЕМЕННОЕ СОСТОЯНИЕ МЕТОДОВ ОБНАРУЖЕНИЯ И РАСПОЗНАВАНИЯ ЛИЦ

Рассматриваются алгоритмы обнаружения и распознавания лиц на изображениях.

ВВЕДЕНИЕ

Различные области науки, техники и производства в большой степени ориентируются на системы, информация в которых представлена изображениями. При обработке такой информации возникает большое количество научных и технологических проблем. Задача обработки и распознавания изображений является одной из самых сложных. Большую популярность набирают системы, использующие биометрическую идентификацию человека.

На сегодняшний день существует множество методов и алгоритмов классификации объектов на изображениях, которые находятся в свободном доступе, обеспечивающих хорошие результаты работы.

I. ОСНОВНОЙ АЛГОРИТМ РАБОТЫ

Для построения системы распознавания человека по фотографии лица необходимо выполнить несколько шагов. Сначала лица должны быть обнаружены на снимке. Затем они предварительно обрабатываются и используются в качестве входных данных для алгоритма распознавания. В итоге, полученная система может использовать изображение, чтобы идентифицировать человека.[1]

II. ОБНАРУЖЕНИЕ ЛИЦ

Прежде чем распознавать лицо, сначала необходимо обнаружить и извлечь лица из исходных изображений. Любой другой элемент изображения, который не является частью лица, ухудшает распознавание. Существует несколько популярных алгоритмов для обнаружения лиц.

Признаки Хаара (Haar-cascade) — это метод, который используется в машинном обучении для обнаружения объектов на картинке. Каскад Хаара, относящийся к Хаар-подобным функциям, является слабым классификатором. Данный классификатор необходимо обучать используя различные изображения как содержащие лица, так и не содержащие их.

Гистограмма направленных градиентов (Histogram of oriented gradients, HOG) — еще

один метод обнаружения объектов, который также может использоваться для обнаружения лиц. Для этого метода требуется изображение в оттенках серого. Каждый пиксель изображения представлен целым числом из-за оттенков серого. Метод HOG сравнивает каждый пиксель со своими соседями.

III. ОБРАБОТКА ИЗОБРАЖЕНИЙ

Следующим шагом является предварительная обработка этих лиц, чтобы упростить фазу обучения и улучшить вероятность правильного распознавания человека. Данные, предназначенные для обучения, будут стандартизированы.

IV. РАСПОЗНАВАНИЕ ЛИЦ

Eigenfaces — метод для распознавания лиц на основе статистического подхода. Целью этого метода является извлечение основных компонентов, которые влияют на наибольшую вариацию изображений.

Fisherfaces также использует статистический подход. Этот алгоритм является модификацией Eigenfaces, поэтому в нем также используется метод главных компонент. Основная модификация заключается в том, что Fisherfaces учитывает классы. Fisherfaces использует метод линейного дискриминантного анализа, чтобы сделать разницу между двумя картинками из другого класса.

Использование сверточной нейронной сети (CNN) — еще один способ выполнения распознавания лиц. CNN имеет архитектуру, которая позволяет использовать 2d-изображения в качестве входных данных. CNN состоит из нескольких слоев, называемых скрытыми слоями. Слои состоят из нескольких нейронов. Нейрон имеет определенный вес и получает вход. После применения его логики к входу, он будет выдавать результат. Входом первого слоя является изображение лица. Результатом последнего слоя является соответствующий класс.

1. Ричарт, В. Построение систем машинного обучения на языке Python / В. Ричарт, Л. П. Коэльо — Изд. 2-е. — М.: ДМК Пресс, 2016. — 302 с.

Чопик Андрей Викторович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, chopik.andrew@gmail.com

Научный руководитель: Матвеевко Владимир Владимирович, кандидат физ.-мат. наук, доцент, vladimir66@bsuir.by.

СИСТЕМНЫЙ АНАЛИЗ СЕРВЕРОВ ДЛЯ ВЫЯВЛЕНИЯ АНОМАЛЬНЫХ И АВАРИЙНЫХ СОСТОЯНИЙ

Рассматривается реализация на примере инфраструктуры Автокодировщик

ВВЕДЕНИЕ

Разработка программных систем требует внимания к отказоустойчивости и быстрого реагирования на отказы и сбои оборудования. Мониторинг системы помогает реагировать на отказы и сбои эффективнее и быстрее, но недостаточно, так как очень сложно уследить за большим количеством серверов. Поэтому часто в системы мониторинга внедряют искусственный интеллект с различными способами анализа данных и принятия решений. Чем больше и сложнее система тем больше метрик необходимо для ее мониторинга. Для сокращения объема анализируемых данных в системах мониторинга есть инструменты для автоматического выявления проблем, например триггеры. Однако именно предсказание возможного сбоя до сих пор остается сложной задачей.

I. ПРЕДИКТИВНОЕ ОБСЛУЖИВАНИЕ И ПОИСК АНОМАЛИЙ

В последнее время в мире популярно предиктивное обслуживание программных систем. Суть данного подхода заключается в нахождении неполадок, ведущих к деградации системы на ранних этапах, до её отказа с использованием искусственного интеллекта. Основным инструментом реализации предиктивного обслуживания является задача поиска аномалий во временных рядах, так как при возникновении аномалии в данных велика вероятность того, что через некоторое время возникнет сбой или отказ. Задача поиска аномалий для программных систем имеет свою специфику. Для каждой программной системы необходима разработка или доработка имеющихся методов, так как поиск аномалий очень зависит от данных, в которых он производится.



Рис. 1 – Поиск аномалий при помощи нейросети

II. АЛГОРИТМ ПОИСКА АНОМАЛИЙ

Задача поиска ошибок решается при помощи обучения с учителем, которым является ад-

министратор. Во время обнаружения аномалии система задает вопрос является ли значение аномальным. К вопросу так же прилагается график, по которому администратор может определиться с ответом. Таким образом, система задает много вопросов и после этого адаптируется к изменениям (см.рис.2.)



Рис. 2 – Поиск аномалий при помощи нейросети

Только те значения, которые коррелируют с другими имеют значимость для обнаружения аномалий и предиктивного обслуживания. Так как все метрики показывают данные одной физической машины, с одним приложением программной системы и каждая метрика по отдельности представляет только численные процентные или временные показатели, то все метрики должны иметь корреляцию хотя бы с одной из других. Если метрика не имеет корреляции в интересующий период, то следить за ней нет смысла.

Выводы

В работе рассматривался алгоритм поиска аномалий системы предиктивного обслуживания для прогнозирования отказов. В основе модели обнаружения аномалий используется синхронный автокодировщик и методы машинного обучения с учителем для улучшения результатов. Данный алгоритм позволяет обучить систему и обеспечить сокращение количества отказов программной системы, а также сокращение времени на поиск неполадок приводящих к отказу или деградации системы.

Шаюнов Евгений Михайлович, магистрант кафедры информационных технологий автоматизированных систем, e.shayunov@gmail.com.

Научный руководитель: Шавель Александр Николаевич, Кандидат физико-математических наук, доцент, an.shavel111@gmail.com.

ИНСТРУМЕНТАЛЬНЫЕ МЕТОДЫ АНАЛИЗА КЛИМАТИЧЕСКИХ ИЗМЕНЕНИЙ

Оценка предстоящих изменений климата с учетом 60-летнего цикла

ВВЕДЕНИЕ

Основным методом обнаружения изменений климата является статистический анализ всех накопленных за исторический период данных наблюдений. Ответить же на вопросы – о том, каким причинам следует приписать обнаруженные изменения и какие изменения климата предстоят в будущем, – может помочь только исследование климатических процессов с помощью физико-математического моделирования глобального климата.

I. ОСНОВНЫЕ ПРИЧИНЫ ИЗМЕНЕНИЙ И ИЗМЕНЧИВОСТИ ПРИЗЕМНОЙ ТЕМПЕРАТУРЫ

В качестве основных физических причин климатической изменчивости приземной температуры ниже рассматриваются:

- изменение концентрации парниковых газов в атмосфере (в данном случае использован временной ряд концентрации CO₂);
- естественная долгопериодная изменчивость (в частности, квази-60-летнее колебание во временных рядах температуры);
- вариации солнечной активности (чаще всего процессы на Земле связываются с числом солнечных пятен и числами Вольфа);
- Низкочастотные составляющие глобальной температуры с периодами отсечения около 60 лет, характеризующие естественную долгопериодную изменчивость.

II. ОЦЕНКА ОЖИДАЕМЫХ ИЗМЕНЕНИЙ СРЕДНЕГОДОВОЙ ГЛОБАЛЬНОЙ ТЕМПЕРАТУРЫ У ПОВЕРХНОСТИ ЗЕМЛИ

Относительный вклад роста концентрации CO₂ в изменчивость температуры во всех рассмотренных регионах значительно больше, чем влияние 60-летнего цикла. На континентах СП (в июле) и в Арктике (в январе) вклады этих двух факторов сопоставимы. Во всех случаях вклад солнечной активности (sunsports) в изменчивость глобальной температуры много ниже (менее 2

Экстремумы квази-60-летнего цикла обнаруживаются в рядах температуры для всех рас-

смотренных регионов и во все сезоны, но не воспроизводятся гидродинамическими моделями. Это позволяет предположить, что цикл связан с внеземными влияниями. Например, это может быть цикл изменения расстояния между Солнцем и Землей вследствие влияния положения крупных планет.

Причиной остаточной межгодовой изменчивости

Снижение глобальной температуры вследствие 60-летнего колебания в первые 30 лет после точки максимума может составить до 0,20–0,25 °С.

С учетом влияния обоих факторов (концентрация CO₂ и квази-60-летнее колебание) можно ожидать через 30 лет, от 1981–2010 к 2011–2040 гг., повышения глобальной температуры на 0,58 0,17 °С, в том числе повышения на 0,72 за счет роста CO₂ и снижения на 0,14 °С за счет 60-летнего колебания. По отношению к базовому 30-летию 1961–1990 гг. это потепление составит в среднем 0,84 °С.

III. ВЫВОДЫ

Исследования климата и ожидаемых изменений климата ведутся во многих странах мира, и координация этих исследований осуществляется на уровне ООН и Всемирной метеорологической организации. Межправительственная Группа экспертов по изменению климата (МГЭИК, англ. – IPCC), в задачи которой входит наиболее полная и объективная оценка наблюдаемых и ожидаемых изменений климата и роли антропогенных факторов, регулярно публикует доклады, обобщающие результаты исследований климата учеными всего мира.

1. Монин А.С. Прогноз погоды как задача физики – «Наука», 1969: 184 с
2. Третье Национальное сообщение Российской Федерации о деятельности по Конвенции ООН об изменении климата. М., 2003.
3. Изменения климата. Обобщенный доклад. 2001 г., ВМО, ЮНЕП, МГЭИК. 2003

Шакунов Дмитрий Александрович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, dmitriy21069@mail.ru

Научный руководитель: Муха Владимир Степанович, доктор технических наук, профессор, mukha@bsuir.by.

ПРИМЕНЕНИЕ МЕТОДА РОЯ ЧАСТИЦ ДЛЯ КАЛИБРОВКИ ВЫСОКОСКОРОСТНЫХ АЦП

Исследована возможность устранения побочных спектральных составляющих, возникающих при работе нескольких АЦП в режиме чередования, используя систему компенсации разброса параметров аналоговых трактов АЦП и процедуру калибровки на основе метода роя частиц.

ВВЕДЕНИЕ

Основными параметрами АЦП являются частота дискретизации и разрядность, причем чем выше частота дискретизации, тем меньше можно рассчитывать на высокую точность преобразования. Аналого-цифровое преобразование с временным чередованием нескольких АЦП позволяет увеличить частоту дискретизации в несколько раз, сохранив динамический диапазон. Однако неточность установки фаз тактовых сигналов и различия в коэффициенте передачи аналоговых трактов вызывают появление паразитных составляющих в оцифрованном сигнале. Для их устранения необходимо использовать линии задержки в цепях тактовых сигналов, и автоматическую процедуру калибровки.

I. МЕТОД РОЯ ЧАСТИЦ

Процесс калибровки можно рассматривать как процесс оптимизации функции нескольких переменных, где аргументами функции являются коэффициенты подстройки фазы и амплитуды, а значением функции - отношение сигнал/шум. В качестве алгоритма оптимизации был исследован метод роя частиц [1]. Принцип работы метода заключается в следующем: создается набор векторов (частиц), координатами которых являются коэффициенты компенсации. Начальные координаты задаются случайным образом так, чтобы частицы были равномерно распределены по области предполагаемых значений отклонения параметров системы. Затем измеряется отношение сигнал/шум в координатах каждой частицы и рассчитывается скорость и новые координаты частиц по формуле 1 и 2 соответственно.

$$v_i^{n+1} = \omega v_i^n + c_1 r_1 (p_i - x_i^n) + c_2 r_2 (g_i - x_i^n) \quad (1)$$

$$x_i^{n+1} = x_i^n + v_i^n \quad (2)$$

где x_i^n - i -я координата частицы при n -ой итерации алгоритма; v_i^n - i -я компонента скорости;

p_i - i -я координата лучшего решения, найденного частицей; g_i - i -я координата лучшего решения, найденного всеми частицами; r_1, r_2 - случайные числа в диапазоне от 0 до 1; ω, c_1, c_2 - инерционные коэффициенты. Процедура повторяется до тех пор, пока одна из частиц не достигнет требуемого отношения сигнал/шум.

II. ЭФФЕКТИВНОСТЬ РАБОТЫ АЛГОРИТМА

В качестве меры эффективности работы алгоритма выбрано количество производимых измерений отношения сигнал/шум. Было проведено моделирование работы алгоритма для калибровки четырех АЦП. На рисунке 1 приведен результат многократного запуска симуляции со случайными отклонениями фазы и амплитуды.

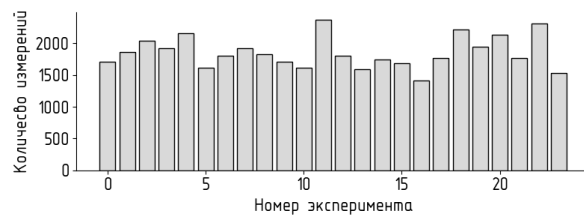


Рис. 1 – Результаты моделирования

III. ВЫВОДЫ

Из результатов моделирования можно сделать вывод о том, что при временных затратах на одно измерение до 1 мс. длительность процесса калибровки не превысит нескольких секунд. Это доказывает возможность использования алгоритма роя частиц для калибровки высокоскоростных АЦП, работающих в режиме чередования.

1. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications / Y. Zhang, S. Wang, G. Ji // Mathematical Problems in Engineering. – 2015. – P. 1–38.

Шекунов Владислав Сергеевич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, shekunov@bsuir.by.

Научный руководитель: Давыдов Максим Викторович, первый проректор БГУИР, кандидат технических наук, доцент, davydov-mv@bsuir.by.

АНАЛИЗ ЭФФЕКТИВНОСТИ ИНВЕСТИЦИОННОЙ ДЕЯТЕЛЬНОСТИ

В данной работе рассматривается теоритическое применение системы, позволяющей оценить эффективность инвестиционной деятельности.

ВВЕДЕНИЕ

В последнее время активно обсуждается проблема повышения конкурентоспособности. Она тесно связана с интенсификацией производственных процессов, протекающих как в народном хозяйстве в целом, так и на уровне отдельных его субъектов.

От эффективности инвестиционной деятельности зависит эффективность операционной и финансовой деятельности. Поэтому повышение эффективности инвестиционной деятельности является неизменно важной задачей для предприятия.

I. МЕТОДИКА РАЗРАБОТКИ ИНВЕСТИЦИОННОГО ПЛАНА

Особенностью инвестиционной деятельности предприятий является ее проектный характер, в связи с чем эта деятельность в целом представляет собой поток инвестиционных проектов, каждый из которых имеет свои особенности. Поэтому существующие методики управления инвестициями строятся на основе управления каждым проектом в отдельности, анализируется эффективность каждого проекта и выдвигается требование эффективности каждого проекта. При этом в качестве критериев эффективности выступают показатели, определяемые по таблице движения денежных средств при использовании в качестве норматива эффективности цены капитала [1].

В целом методика разработки инвестиционного плана может быть представлена следующим образом: анализ и оценка текущего финансово-экономического состояния предприятия с учетом реализуемых проектов; выявление альтернатив развития предприятия – совокупности проектов, готовящихся к реализации в будущих периодах; отбор приемлемых для реализации проектов по критериям, принятым для оценки отдельных проектов; составление нескольких альтернативных инвестиционных планов на несколько лет вперед; прямой имитационный анализ и оценка отобранных инвестиционных

планов; выбор одного инвестиционного плана по рейтинговому критерию.

II. ОЦЕНКА ЭФФЕКТИВНОСТИ ОТДЕЛЬНЫХ ПРОЕКТОВ

Эффективность проекта оценивают в течение расчетного периода, включающего временной интервал от начала проекта до его прекращения. Начало расчетного периода рекомендуют определять на дату вложения денежных средств в проектно-изыскательские работы. Прекращение реализации проекта может быть следствием: прекращения производства в связи с изменением требований (норм и стандартов) к выпускаемой продукции, технологии производства или условиям труда на данном предприятии; отсутствия потребности рынка в данной продукции в связи с ее моральным устареванием или потерей конкурентоспособности; износа активной части основных производственных фондов [2].

При необходимости в конце расчетного периода предусматривают ликвидацию объекта. Расчетный период разбивают на шаги-отрезки, в пределах которых осуществляется агрегирование данных, используемых для оценки финансовых показателей.

Отрезки времени, в которых прогнозируют высокие темпы инфляции (свыше 10%), целесообразно разбивать на более мелкие шаги.

Денежный поток проекта — это зависимость от времени денежных поступлений и платежей при реализации порождающего его проекта, определяемая для всего расчетного периода.

III. ВЫВОДЫ

Предлагаемая нами система позволяет определить наиболее эффективный план и эффективность инвестиционной деятельности.

1. Виленский П. Л., Лившиц В. Н., Смоляк С. А. Оценка эффективности инвестиционных проектов: Теория и практика: учеб. пособие. М.: Дело, 2002.- 888 с.
2. Методические рекомендации по оценке эффективности инвестиционных проектов: официальные издание (вторая ред.) «Экономика», 2000. - 421 с.

Шостыр Александр Иванович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, sashashostyr@gmail.com.

Научный руководитель: Севернёв Александр Михайлович, доцент кафедры информационных технологий автоматизированных систем БГУИР, доцент, severnev@bsuir.by.

РАЗРАБОТКА СИСТЕМЫ ОБРАБОТКИ ЗАДАЧ ОРГАНИЗАЦИИ

В данной работе рассматриваются существующие системы обработки задач организации. Указываются наиболее важные аспекты в разработке системы.

ВВЕДЕНИЕ

В организациях, где количество сотрудников более 3-4 человек, одним из ключевых факторов, определяющих успех предприятия, является грамотное управление задачами. Программный модуль обработки задач организации позволяет четко определять цели, границы и содержание работ, а так же роли участников команды.

I. АНАЛИЗ СУЩЕСТВУЮЩИХ СИСТЕМ

В настоящее время на рынке программных средств существует несколько приложений для управления задачами предприятия и организации рабочего процесса внутри коллектива. Ниже рассмотрены наиболее популярные из них.

Atlassian Jira. На сегодняшний день Jira является одним из самых известных и популярных решений для управления проектом. Atlassian Jira позволяет планировать задачи по версиям проекта, редактировать, добавлять комментарии, осуществлять полнотекстовый поиск.

Приложение существует в виде веб-приложения, программы на компьютер, а также имеет мобильный интерфейс.

К недостаткам системы можно отнести следующие пункты: *платная версия (можно получить бесплатную версию для Opensource-проекта); огромный функционал, препятствующий быстрому началу работы; не подходит для небольших команд;*

MS Project. MS Project – программа управления проектами, разработанная и продаваемая корпорацией Microsoft.

MS Project создан, чтобы помочь менеджеру проекта в разработке планов, распределении ресурсов по задачам, отслеживании прогресса и анализе объемов работ.

Наиболее очевидным преимуществом продукта является то, что он входит в семейство MS Office. Это обеспечивает следующие плюсы, характерные для всех продуктов Microsoft: *малое время обучения пользователей; богатые возможности по настройке в стиле формул Microsoft Excel; возможность адаптировать продукт под свою специфику путём программирования на базе Visual Basic;* К недостаткам MS

Project можно отнести следующее: *не поддерживает совместную работу; отсутствие локализации программы для СНГ-региона;*

II. РАЗРАБОТКА СИСТЕМЫ

Изучив требования организаций к системе обработки задач, а также проанализировав существующие решения, заключим, что для разработки успешного приложения необходимо уделить особое внимание следующим аспектам:

1. *Информационное обеспечение* Информационное обеспечение представлено входными и выходными данными.
2. *Организационное обеспечение* Организационное обеспечение — это совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами.
3. *Эргономическое обеспечение.* Эргономическое обеспечение — совокупность методов и средств, предназначенных для создания оптимальных условий эффективной деятельности человека в информационной системе, для ее быстрого освоения.
4. *Программная реализация* Программный продукт должен отвечать следующим требованиям: доступность, надежность, требование к времени хранения данных, безопасность, поддержка локализации.

III. ВЫВОДЫ

В работе рассмотрена роль системы обработки задач в организации, существующие альтернативы. На основании достоинств и недостатков существующих решений, а также анализа потребностей предприятий, освещены наиболее важные аспекты разработки данного программного продукта.

1. Advanta [Электронный ресурс]. – Режим доступа: <http://www.advanta-group.ru/about-system/sistemaupravlenia-proektami/>

Шихмуратов Эльдар Арифович, магистрант кафедры информационных технологий автоматизированных систем БГУИР, eldar.shykhmuradov@gmail.com

Научный руководитель: Свито Игорь Леонтьевич, доцент кафедры ТОЭ БГУИР, кандидат технических наук, доцент, kaftoe@bsuir.by.

АЛГОРИТМЫ УПРАВЛЕНИЯ РАСПРЕДЕЛЁННЫМИ ПРОГРАММНО-ОПРЕДЕЛЯЕМЫМИ СЕТЯМИ

Рассмотрена архитектура программно-определяемой сети, а так же её преимущества по сравнению с традиционными сетями. Проведён анализ протокола OpenFlow.

ВВЕДЕНИЕ

Современные тенденции, такие как рост числа подключенных к Интернету устройств, экспоненциальный рост объемов информации, развитие облачных технологий, корпоративная мобильность, большие данные, на глазах меняют корпоративный телеком. Идет наращивание объемов сетевого трафика, и у бизнеса все чаще возникает необходимость конфигурировать крупномасштабные сети. Упростить эту задачу могут технологии программно-определяемых сетей, которые позволяют перевести сетевые элементы под контроль настраиваемого ПО, сделать их более интеллектуальными, облегчить управление ими.

I. АРХИТЕКТУРА ПРОГРАММНО-ОПРЕДЕЛЯЕМОЙ СЕТИ

В программно-определяемой сети (SDN) уровни управления сетью и передачи данных разделяются за счет переноса функций управления (маршрутизаторами, коммутаторами и т. п.) в приложения, работающие на отдельном сервере (контроллере).

В архитектуре SDN можно выделить три уровня (см.рис.1.):

- инфраструктурный уровень, предоставляющий набор сетевых устройств;
- уровень управления, включающий в себя сетевую операционную систему, которая обеспечивает приложениям сетевые сервисы и программный интерфейс для управления сетевыми устройствами и сетью;
- уровень сетевых приложений для гибкого и эффективного управления сетью.

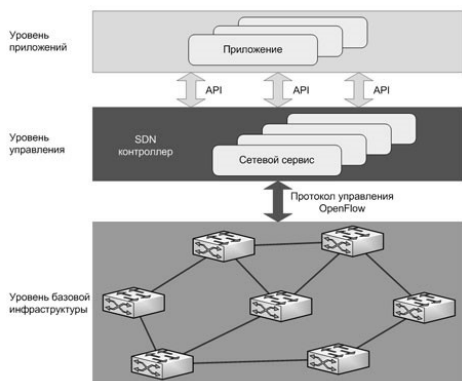


Рис. 1 – Архитектура программно-определяемой сети

Как видно из архитектуры, SDN контроллер поддерживает запуск на себе приложений управления сетью. Каждое SDN приложение, являясь собой интерфейс оптимизации сети под конкретное бизнес-приложение и его основная роль — изменение сети в реальном времени под текущие нужды обслуживаемой программы.

II. ПРОТОКОЛ OPENFLOW

Наиболее перспективным и активно развивающимся стандартом для SDN является OpenFlow – открытый стандарт, в котором описываются требования, предъявляемые к коммутатору, поддерживающему протокол OpenFlow для удаленного управления. Протокол представляет независимый от производителя интерфейс между логическим контроллером сети и сетевым транспортом.

Механизм работы коммутатора достаточно прост. У каждого пришедшего пакета «вырезается» заголовок (битовая строка определенной длины). Для этой битовой строки в таблицах потоков, начиная с первой, ищется правило, у которого поле признаков ближе всего соответствует заголовку пакета. При наличии совпадения, над пакетом и его заголовком выполняются преобразования, определяемые набором инструкций, указанных в найденном правиле. Инструкции, ассоциированные с каждой записью таблицы, описывают действия, связанные с пересылкой пакета, модификацией его заголовка, обработкой в таблице групп, обработкой в конвейере и пересылкой пакета на определенный порт коммутатора.

Управление данными осуществляется не на уровне отдельных пакетов, а на уровне их потоков. Правило в коммутаторе устанавливается с участием контроллера только для первого пакета, а затем все остальные пакеты потока его используют.

Протокол поддерживает три типа сообщений: контроллер-коммутатор, асинхронные и симметричные.

Сообщения типа контроллер-коммутатор инициируются контроллером и используются для непосредственного управления и слежения за состоянием коммутатора. Сообщения данного типа могут использоваться контроллером для установки параметров конфигурации коммутатора, для сбора статистики, для добавления, уда-

ления и модификации записей в таблицах потоков.

Асинхронные сообщения инициируются коммутатором для оповещения контроллера о сетевых событиях (прибытии пакетов или удалении записи из таблицы по истечении времени) и изменениях состояния коммутатора или ошибках.

Симметричные сообщения могут инициироваться коммутатором или контроллером без запроса и используются при установлении соединения, а также при измерении задержек, пропускной способности соединения контроллер-коммутатор или для проверки живучести соединения.

III. СРАВНЕНИЕ С ТРАДИЦИОННЫМИ СЕТЯМИ

Контур управления трафиком в SDN сети перемещается с уровня устройств и элементов сети (физических или виртуальных) на централизованный уровень специального ПО. С точки зрения операторов, интерес в SDN связан с повышением эффективности сетевого оборудования, снижением затрат, повышением сетевой безопасности и предоставлением возможности легко создавать новые сервисы и оперативно загружать их в сетевое оборудование. SDN формирует виртуальный уровень для сети, аналогично тому, как виртуальная машина делает это для серверов или настольных ПК.

Это в свою очередь приводит к двум основным преимуществам программно-определяемых сетей:

- Снижение капитальных и операционных затрат, совокупной стоимости владения сетью
- Быстрота внедрения и адаптации услуг

IV. ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ

При переходе на программно-определяемую модель функции управления устройствами и трафиком централизуются и переводятся в программную форму. Их команды обеспечивают слаженную работу всей инфраструктуры. На долю коммутатора остается только функция передачи данных. В результате внесенных изменений коммутатор становится более простым.

Переход на программно-определяемую модель позволяет упростить задачи масштабирования инфраструктуры, функции настройки и управления становятся более гибкими, появляются дополнительные ресурсы для работы с прикладной нагрузкой, возможности для оптимизации и исправления ошибок на уровне ПО, более высокую вычислительную мощность, нара-

щивать ресурсы хранения данных и сетевой коммутации, причем это достигается без выделения дополнительных серверов.

SDN создает совершенно новую сетевую архитектуру, в которой «интеллект» сети управляется централизованно с помощью софта, обеспечивающего повышенную автоматизацию, ловкость, гибкость, рентабельность — и простоту. SDN — результат того, что происходит, когда сети двигаются дальше, а инфраструктура переходит в фоновый режим. В современных сетях больше аспектов, нуждающихся в контроле и управлении, чем когда-либо — поэтому компании нуждаются в соответствующих решениях.

V. ВЫВОДЫ

Использование программно-определяемых сетей, вместе с протоколом OpenFlow имеет очевидные преимущества:

1. повышение эффективности использования пропускной способности каналов, балансировку нагрузок в сети;
2. упрощение управление сетью, повышение масштабируемости сети;
3. эффективную маршрутизацию;
4. снижение капитальных затрат;
5. увеличение надежности функционирования сети с помощью централизованного управления конфигурацией сети;
6. динамическая подстройка емкости виртуальной сети и других параметров под растущие потребности клиента без изменения физической топологии;
7. создание изолированных виртуальных сетей для каждого клиента ИТ-услуг на базе единой физической инфраструктуры.

Такой подход отвечает запросам операторов, производителей компьютерного и телекоммуникационного оборудования, владельцев дата-центров, финансовых структур и банков, хостинга и сервис-провайдеров.

1. Смелянский, Р. Л. Программно-конфигурируемые сети. – 2012 С. 15-26.
2. Вишняков, В. А., Мониц, Б. А. Технологии программно-определяемых сетей в облачных вычислениях. – 2018 С. 70-75
3. Hu, Fei Network Innovation through OpenFlow and SDN: Principles and Design – 2014.

Юхневич Олег Игоревич, магистрант кафедры информационных технологий автоматизированных систем БГУИР, whyoleg@gmail.com.

Научный руководитель: Хаджинов Михаил Касьянович, кандидат технических наук, доцент, kh_m@tut.by.

СИСТЕМНЫЙ АНАЛИЗ ГОРОДСКОЙ СРЕДЫ ПОСРЕДСТВОМ МОНИТОРИНГА СОЦИАЛЬНОЙ СЕТИ TWITTER

ВВЕДЕНИЕ

Социальные сети предоставляют большой объём географических данных и могут использоваться для создания модели городского пространства с целью планирования инфраструктуры для общественных мероприятий [1,2]. Для изучения городского пространства Индианаполиса использовался трафик социальной сети Twitter сгенерированный во время Суперкубка 2012 года.

I. АРХИВАЦИЯ И ФИЛЬТРАЦИЯ ТРАФИКА СОЦИАЛЬНОЙ СЕТИ

Для архивации трафика используется инструмент The Archivist (Microsoft, 2010), обеспечивающий доступ к базе данных Twitter и сохраняющий твиты с метками [3]. Сохранение всех твитов с метками Суперкубка теоретически возможно, но практически труднодостижимо [4]. Ограничения, существующие в Twitter, не могут гарантировать, что инструмент архивирования смог вернуть каждый твит с целевым хэштегом в течение определенного периода времени (программное обеспечение позволяет выгрузить не более 1500 твитов за раз [5]). Во время самой игры была захвачена лишь небольшая часть общего трафика, так как во время перерыва Суперкубка трафик Twitter превышал 10 000 твитов в секунду (хотя какая часть этих твитов содержала наши целевые хэштеги, неизвестно). На начальном этапе были выбраны официальные хэштеги мероприятия, а на последующих этапах фильтрации были добавлены неофициальные принятые пользователями сети для маркировки событий не предусмотренных расписанием [6,7]. Например, если знаменитость, посетившая город, написала в твиттере о том, что она находится в игре, люди начинают пользоваться хэштегом этой знаменитости, вместо официальных тегов Суперкубка. Для этого анализа были сохранены только твиты с 23 января 2012 до 9:00 утра 6 февраля 2012. Пользователи Twitter часто применяют несколько хэштегов в одном твите, поэтому многие твиты были записаны несколькими раз. Так как каждому твиту присваивается уникальный идентификационный номер, записи дубликатов легко идентифицируются и удаляются.

II. КОДИРОВАНИЕ ТРАФИКА СОЦИАЛЬНОЙ СЕТИ

Для эффективного анализа полученных данных необходимо выделить основные элементы городской среды, по которым в дальнейшем будет проведено кодирование собранных сообщений [9,10].

Таблица 1 – Базовые элементы городской среды

Название элемента	Описание
Маршруты	Улицы, тропы и другие маршруты, по которым движутся люди
Границы	Края или переходные зоны между двумя зонами. Делятся на природные (речные и горные) и искусственные барьеры (дороги, стены и железнодорожные пути).
Районы	Большие участки города, уникальными свойствами. Например спальный район, центр города или офисный квартал.
Ориентиры	Легко идентифицируемые объекты, которые служат ориентирами, такие как памятники и стадионы.
Узлы	остановки общественного транспорта, площади, перекрестки или вокзалы.

Вручную были соотнесены ключевые слова с пятью элементами городского пространства (табл. 1). Каждое из пространственных слов и словосочетаний, идентифицированных в ходе контент-фильтрации, было отнесено к одной из категорий IC, чтобы обеспечить машинное кодирование контента твита с помощью поиска по шаблону. Из 78 терминов 30 (38,5 процента) были узлами, 18 (10,3 процента) - маршрутами, 16 (20,5 процента) - названными ориентирами, 11 (14,1 процента) - районами и три (3,8 процента) - краями

III. АНАЛИЗ РЕАЛЬНЫХ ДАННЫХ

С 12:01 утра 23 января 2012 г. до 9:00 утра 6 февраля 2012 г. The Archivist зарегистрировал 648 893 твита, содержащих один или несколько целевых хэштегов. Удаление дубликатов сократило количество до 499 756 твитов. Среднее количество твитов в день составило 33 317. Зарегистрированный трафик "Твиттера" был относительно низким в течение большей части иссле-

двумя периода с тенденцией к росту объема в четверг, 2 февраля (рисунок 1), возможно, из-за требования многих отелей в центре города, чтобы посетители Суперкубка резервировали номера как минимум на три или четыре ночи. Пик дневного трафика в 250 000 твитов произошел в день игры 5 февраля. Среднее количество записанных твитов в час составило 1448,6 (рисунок 2).

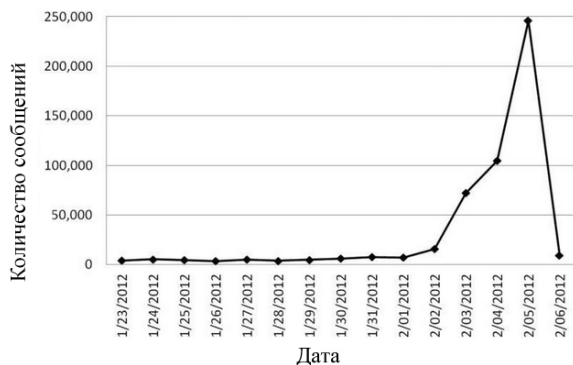


Рис. 1 – Зависимость количества сообщений от дня мероприятия



Рис. 2 – Зависимость количества сообщений в час от времени суток

IV. ЗАКЛЮЧЕНИЕ

Все элементы изображения города были найдены в специфическом для Индианаполиса трафике Twitter. В то время как узловые ссылки были наиболее распространены среди 78 географических идентификаторов. Чаще всего упоминались районы и ориентиры. Ориентиры и узлы обычно представляют собой отдельные объекты и легко запоминаются, служа в качестве ориентиров [8]. Упоминания границ практически отсутствуют и встречаются только в виде упоминания рек; это может указывать на их ограничен-

ную полезность в коротких текстовых описаниях городского пространства или на относительное отсутствие границ городского пространства Индианаполиса. Общая частота трафика в Твиттере и географических "твитов" увеличивалась с течением времени; тем не менее, они имели несколько иную схему распределения. Связанный с событиями трафик достигает пика в день игры, а географические ссылки достигают пика двумя днями ранее, что говорит о том, что посетители и местные жители по-разному используют социальную сеть в предыгровой период и в день игры. Частоты различных типов географических ссылок также различались по дням. Термины "район" "узел" и "маршрут" достигли своего пика 3 февраля, в отличие от точек, пик которых пришелся на 5 февраля. Это повышает вероятность того, что туристы чаще задавали общие вопросы о местоположении или описывали общие планы в режиме онлайн, когда они впервые приехали в город, а в день игры внимание переключилось на стадион (ориентир).

V. СПИСОК ЛИТЕРАТУРЫ

1. Гольденберг и М. Леви. Расстояние не умерло: социальное взаимодействие и географическая дистанция в эпоху Интернета. Arxiv, abs / 0906.3202, 2009.
2. Arribas-Bel, K. Kourtit, P. Nijkamp, and J. Steenbruggen, "Cyber Cities: Social Media as a Tool for Understanding Cities," Applied Spatial Analysis and Policy 8: 3 (2015) 231–247.
3. D. Caduff and S. Timpf, "A Framework for Assessing the Salience of Landmarks for Wayfinding Tasks," Cognitive Processing 1: 7 (2006) 23–23.
4. G. Cao, S. Wang, M. Hwang, A. Padmanabhan, Z. Zhang, and K. Soltani, "A Scalable Framework For Spatiotemporal Analysis of Location-Based Social Media Data," Computers, Environment and Urban Systems 51 (2015) 70–82.
5. S. Carr, L. Rodwin, and G. Hack, "Kevin Lynch: Designing the Image of the City," Journal of the American Planning Association 50: 4 (1984) 523
6. C. Chew and G. Eysenbach, "Pandemics in the Age of Twitter: Content Analysis of Tweets during the 2009 H1N1 Outbreak," PLoS ONE 5: 11 (2010) e14118.
7. Eagle, A. Pentland, and D. Lazer. Определение структуры сети дружбы с помощью данных мобильного телефона. PNAS, 2009.
8. Long, P. K. Baren, and R. Moore, "The Role of Space Syntax in Spatial Cognition: Evidence from Urban China," Conference Proceedings of the 6th International Space Syntax Symposium, (Istanbul, Turkey, June 2007).
9. A. Crooks, D. Pfoser, A. Jenkins, A. Croitoru, A. Stefanidis, D. Smith, S. Karagiorgou, A. Efentakis, and G. Lamprianidis, "Crowdsourcing Urban Form and Function," International Journal of Geographical Information Science (2015) 37–41.
10. Lynch, The Image of the City (Cambridge MA: MIT Press, 1960) 20-25

Якубович А.В., Кафедра информационных технологий автоматизированных систем Белорусский государственный университет информатики и радиоэлектроники, yakubovichandrei@gmail.com

Научный руководитель: Шилин Л.Ю., доктор технических наук, профессор, yakubovichandrei@gmail.com

Секция "Вычислительные методы и программирование"

Председатель: канд. тех. наук, доцент Кукин Д.П.
Члены жюри: канд. физ.-мат. наук Гуринович А.Б.
д-р физ.-мат. наук, проф. Колосов С.В.
канд. тех. наук, доцент Волковец А.И.
Секретарь ст. преп. Шестакович В.П.

ПОЛЕЗНЫЕ ПЛАГИНЫ ДЛЯ UNITY

В любом приложении можно столкнуться с проблемой нехватки функционала, не найти необходимых функций, эта задача с легкостью решается благодаря плагинам, бесплатным сторонним решениям для разработки, монетизации, аналитики. Плагины и ассеты позволяют разработчикам использовать в своём проекте готовые решения, созданные другими людьми.

ВВЕДЕНИЕ

Unity — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

I. ИГРЫ НА UNITY

На Unity написаны тысячи игр, приложений, визуализации математических моделей, которые охватывают множество платформ и жанров. При этом Unity используется как крупными разработчиками, так и независимыми студиями. Вот несколько игр, которые будут рассмотрены в работе: Hollow Knight, Cuphead, Outer Wilds.

II. ПЛАГИНЫ

Плагины и ассеты для создания игрового мира, персонажей и объектов:

- GeNa 2 – плагин для генерации природного ландшафта ;
- Ambient Skies – позволяет генерировать облака и погодные условия;
- Aura 2 – это плагин для создания объёмного света и тумана;

Аксак Юлия Дмитриевна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, yulia.aksak@mail.ru.

Змитрович Елизавета Владимировна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, zmitrovichliza0@gmail.com.

Гриневич Яна Григорьевна, ассистент кафедры ВМиП, ja.grinevich@bsuir.by.

Научный руководитель: Рак Татьяна Александровна, старший преподаватель кафедры ВМиП, tatianarak@bsuir.by.

- A* Pathfinding Project Pro – это плагин, который позволяет ИИ персонажей адекватно оценивать окружение и выполнять сложные задачи;
- Mega SFX Pack – содержит 6 тысяч звуковых эффектов;
- Final IK – это набор, который позволяет анимировать двуногих персонажей с использованием принципов инверсной кинематики;
- Dialogue System – диалоговая система, которая основана на нодах;
- Character Creator 2D – это конструктор, в котором доступно большое количество элементов, из которых и собирается герой.

III. НЕОБХОДИМОСТЬ ПЛАГИНОВ

В Unity система частиц и визуальных эффектов, а также освещения дает возможность получить неплохие результаты, но для этого стоит потратить не малое количество времени и сил, чтобы прийти к качественному контенту. А создание атмосферы в игре является одним из главных факторов восприятия игры в целом. Существует большое количество плагинов, которые значительно упростят работу с Unity разработчику и позволят сделать игру максимально атмосферной, красочной и правдоподобной.

Список литературы

1. DTF [Электронный ресурс]. - Режим доступа: <https://dtf.ru>.
2. unity Asset Store [Электронный ресурс]. - Режим доступа: <https://assetstore.unity.com>.

ОШИБКИ ОПРЕДЕЛЕНИЯ ЦЕЛЕВОЙ АУДИТОРИИ В ИГРАХ

I. ПОНЯТИЕ И КРИТЕРИИ ВЫДЕЛЕНИЯ ЦЕЛЕВОЙ АУДИТОРИИ

Во-первых игра создаётся для конкретной целевой аудитории. Если на ранней стадии разработки ЦА была определена с ошибками, то весь процесс пойдет в неверном направлении.

- Целевая аудитория – это пользователи, объединенные в группы по характеризующим их общим признакам, на которую рассчитан ваш продукт. По каким критериям геймдизайнеры, аналитики и маркетологи делят аудиторию?
- По демографическому принципу (пол, возраст, регион).
- Предпочитаемые жанры. Существует целый ряд классификаций игр по жанрам.
- Поведенческие особенности. Здесь подразумеваются психотипы. Ричард Аллан Бартл предложил систему сегментации игроков на четыре психотипа: киллеры, карьеристы, социальщики, исследователи.
- Казуальность. Это качественная характеристика, которая делит аудиторию на «хардкорщиков», «минкорщиков» и «казуалов».
- Принятие инноваций. Отношение игроков к новым фишкам, возможностям и типам геймплея. Вся аудитория делится на пять групп по степени принятия инноваций: инноваторы, ранние апологеты, ранее большинство, позднее большинство и консерваторы.
- Платёжеспособность.

II. ПРОБЛЕМА ОПРЕДЕЛЕНИЯ ЦЕЛЕВОЙ АУДИТОРИИ В КОНТЕКСТЕ ИГРОВОЙ ИНДУСТРИИ

Существует ряд ошибок, которые допускаются чаще всего. Рассмотрим их подробнее, а также найдём решение для каждой из этих проблем.

- 1) Выделение слишком конкретной целевой аудитории. Создание слишком детального

образа серьёзно ограничивает вашу аудиторию. Для решения этой проблемы нужно составить более универсальное описание клиента и включить в него только те детали, которые повлияют на решение клиента.

- 2) Использовании только сухих фактов при составлении ЦА. На выходе мы получаем социальный портрет клиента с основными параметрами: пол, возраст, доход, семейное положение. По сути это верно, однако, внутренняя аудитория разнообразна. Соответственно, нужно добавить к сухим фактам интересы, желания и страхи клиентов (проблемы, которые могут возникать у ЦА).
- 3) Мнение, что все ваши клиенты одинаковы. При тщательной проработке целевой аудитории может сложиться впечатление, что ваша аудитория однородна. Но на самом деле она внутренне делится на несколько сегментов. Чтобы не ошибиться, лучше всего сегментировать составленную ЦА, и далее разобрать каждый сегмент отдельно.
- 4) Ошибка объединение потребителя и платёжщика очень сильно влияет на рекламу. Решение этой проблемы очень простое: разделить потребителей и платёжщиков на разные сегменты, но оставить их как единую ЦА.

III. ВЫВОДЫ

Таким образом, рамках этой работы были проанализированы наиболее часто совершаемые ошибки при определении целевой аудитории в игровой индустрии, а также были рассмотрены их наиболее рациональные решения приведённых в докладе.

1. <https://kopchenkov.com/2019/08/27/tselevaya-auditoriya/> Основные ошибки при определении целевой аудитории
2. <https://habr.com/ru/company/mailru/blog/274263/> Сегментация аудитории при разработке игр

Борисевич Полина Владимировна,
polina.borisevich.86@gmail.com.

Научный руководитель: Коршикова Дарья Валерьевна, ассистент кафедры ВМиП БГУИРа,
korshikova@bsuir.by

студент факультета ИТиУ БГУИРа,

ОЦЕНКА РИСКОВ В БИЗНЕСЕ

Рассматриваются подходы к реализации задач, связанных с бизнесом и инвестированием, а также сделаны выводы о выгодности инвестиций в проекты с определёнными рисками.

ВВЕДЕНИЕ

Ведение бизнеса и, в частности, торговля акциями, можно сравнить с азартной игрой: акции могут вырасти или упасть в цене. Рассмотрим две математические модели. За определённый промежуток времени акции могут вырасти или упасть в цене. В первой модели изменение в цене акции не зависит от текущей цены (стоимость увеличивается или уменьшается на некоторое количество денежных единиц с некоторой вероятностью) и система обладает свойством эргодичности, во втором – зависит (стоимость увеличивается или уменьшается на определённый процент от текущей), и система неэргодична.

I. ЭРГОДИЧНАЯ МОДЕЛЬ

Можно предсказать изменение стоимости акций, если знать все возможные исходы и их вероятности. Математическое ожидание стоимости акции, если каждое следующее изменение стоимости акций компании не зависит от предыдущего, в таком случае, будет выглядеть следующим образом:

$$C_0 + n * \sum_{(i=1)}^m x_i * p_i$$

где C_0 – первоначальная стоимость одной акции, n – количество временных отрезков между покупкой акций и фиксацией прибыли, $x_{1..m}$ – изменение стоимости одной акции, $p_{1..m}$ – вероятность соответствующего изменения стоимости одной акции за промежуток времени.

Иллюстрация первой модели: на рынке торгуются акции различных предприятий. В начальный момент времени каждая акция каждого предприятия стоят по сто долларов. Каждый день акции предприятия с равной вероятностью или становятся дороже на 10 долларов, или дешевеют на 9,5 долларов. Спустя десять дней, разброс стоимости акций предприятий на рынке будет выглядеть, как показано в таблице 1. Исходя из таблицы, большинство предприятий увеличило стоимость своих акций, соответственно, акционеры большинства предприятий выиграли. Среднее значение стоимости акций – 102,5 долларов, столько же стоят акции самой крупной группы предприятий, у которых удачных и неудачных дней было равное количество. Это значение соответствует формуле математического ожидания.

Таблица 1 – Распределение компаний по конечной стоимости акций для эргодичной модели

Дней успеха	Процент	Стоимость
0	0,10%	5
1	0,98%	24,5
2	4,39%	44
3	11,72%	63,5
4	20,51%	83
5	24,61%	102,5
6	20,51%	122
7	11,72%	141,5
8	4,39%	161
9	0,98%	180,5

II. НЕЭРГОДИЧНАЯ МОДЕЛЬ

Математическое ожидание стоимости акций, если каждое следующее изменение стоимости акций компании зависит от предыдущего:

$$C_0 * \left(\sum_{(i=1)}^m x_i p_i \right)^n$$

Условные обозначения аналогичны обозначениям к предыдущей формуле. В случае, если изначально на рынке торгуется множество акций с одинаковым распределением вероятностей, модальную стоимость акций компании можно посчитать по формуле:

$$(C_0 * x_1 p_1 * x_2 p_2 * \dots * x_m p_m) n$$

Иллюстрация неэргодичной модели: на рынке торгуются акции различных предприятий, в начальный момент акции каждого предприятия стоят по сто долларов. Каждый день акции предприятия с равной вероятностью или становятся дороже на 10%, или дешевеют на 9,5%. Разброс стоимости акций предприятий на рынке спустя десять дней будет выглядеть, как показано в таблице 2. Исходя из таблицы, акционеры большинства предприятий потеряли деньги, и выиграли только акционеры меньшего, чем в первом случае, количества самых удачливых предприятий, причём их выигрыш больше выигрыша акционеров самых удачливых предприятий из первой задачи. В то же время, самые неудачливые компании потеряли в стоимости акций меньше, чем в первой модели.

Таблица 2 – Распределение компаний по конечной стоимости акций для неэргодичной модели

Дней успеха	Процент	Стоимость
0	0,10%	36,85
1	0,98%	44,80
2	4,39%	54,4544
3	11,72%	66,18
4	20,51%	80,44
5	24,61%	97,77
6	20,51%	144,44
7	11,72%	175,57
8	4,39%	213,39
9	0,98%	259,37

III. СРАВНЕНИЕ МОДЕЛЕЙ

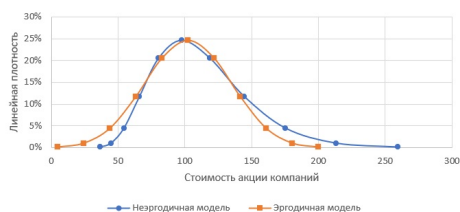


Рис. 1 – Плотность распределения компаний по стоимости акций

Вельков Дмитрий Евгеньевич, студент кафедры информационных технологий автоматизированных систем БГУИР, velkov2000@list.ru.

Научный руководитель: Гуринович Алевтина Борисовна, заместитель декана факультета информационных технологий и управления БГУИР, кандидат физ.-мат. наук, доцент, gurinovich@bsuir.by.

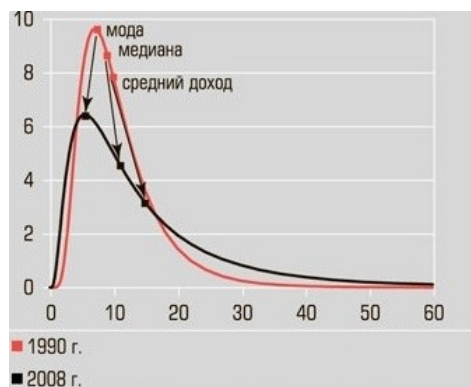


Рис. 2 – Плотность распределения населения России по доходу за 1990 и 2008 годы

IV. ВЫВОДЫ

Между графиком распределения неэргодичной модели и реальным графиком распределения доходов наблюдается некоторое сходство. Таким образом, можно сделать вывод, что неэргодичная математическая модель имеет большую степень сходства с реальностью, и даже если на рынке наблюдается общий тренд роста, это не значит, что большинство акционеров получают прибыль.

1. O. Peters, "The ergodicity problem in economics", Nature Physics volume 15, pages1216–1221, 2019.
2. O. Peters, M. Gell-Mann, "Evaluating gambles using dynamics", Chaos volume 26, 023103, 2016.
3. <https://expert.ru/expert/2011/14/zhit-stalo-luchshe-no-ne-veselee/media/79896/>

ОПТИМИЗАЦИЯ КОДА JIT-КОМПИЛЯТОРОМ

В работе проведено сравнение производительности компиляторов LLVM Clang(c++) и Hotspot-JIT C2(Java).

ВВЕДЕНИЕ

Существование компромисса между компиляторами и интерпретаторами привело к возникновению JIT-компиляторов, которые в свою очередь задумывались для повышения производительности интерпретируемых языков.

I. КРАТКАЯ СПРАВКА

Компилятор преобразует исходный код программы в машинный код и только после этапа компоновки программа становится исполняемой. Интерпретатор выполняет команды построчно. Отсюда и следуют их главные минусы: компилятор-долгая сборка проекта и отсутствие кроссплатформенности, интерпретатор-проблемы с производительностью конечного приложения. JIT-компилятор считается динамическим компилятором, т.к. обладает свойствами интерпретатора (построчное выполнение) и компилятора (выборочная компиляция). JIT-компилятору необходимы данные о работе программы, основываясь на которых он делает разнообразные оптимизации. JIT реализован в Java, .NET Framework, Javascript, Python(PyPy), Ruby.

II. НЕКОТОРЫЕ ОПТИМИЗАЦИИ, ПРОВОДИМЫЕ JIT

- Развёртка циклов
- Устранение общих подвыражений
- Устранение мёртвого кода
- Сворачивание констант
- Подмена операций со стеком на операции с регистрами
- Инлайнинг функций
- Скаляризация

Волощик Алексей Викторович, студент гр.961401, frscfi@gmail.com.

Научный руководитель: Гриневич Яна Григорьевна, ассистент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники, ja.grinevich@bsuir.by.

III. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Таблица 1 – Сравнение времени выполнения задач между JIT и Clang

Тест	Условия	JIT мс/оп	Clang мс/оп
Сумма N элементов массива (N заранее неизвестно)	Размер массива - 10^4	3.2	0.8
Сложение элементов двух массивов в третий	Размер массива - 10^9	61.9	62.4
Сложение константы и элементов массива в i элемент исходного	Размер массива - 10^9	30.0	31.6
Сложение двух полей массива структур	Размер массива - 2^{11}	1.32	1.18
Суммирование при условии заблокированных потоков	-	2.8	190

IV. ВЫВОД

Традиционно C++ считается достаточно быстрым языком программирования, однако, по результатам тестирования, можно увидеть, что Java с использованием технологий JIT-компиляции в некоторых случаях выступает на равных, а в некоторых даже обходит его по скорости выполнения операций. Технология JIT-компиляции активно развивается различными технологическими гигантами и поэтому вполне возможно, что в ближайшем будущем нас ждёт универсальный и производительный инструмент, позволяющий отбросить ограничения многих интерпретируемых языков и сделать разработку ещё более удобной для человека.

Список литературы

1. <https://hacks.mozilla.org/2017/02/a-crash-course-in-just-in-time-jit-compilers/>
2. <https://www.telerik.com/blogs/understanding-net-just-in-time-compilation>
3. <https://ionutbalosin.com/2019/04/jvm-jit-compilers-benchmarks-report-19-04/>
4. <https://www.infoq.com/articles/Graal-Java-JIT-Compiler/>

ОСОБЕННОСТИ СОЗДАНИЯ ИГРОВЫХ МОДЕЛЕЙ

Рассматриваются особенности создания 3D моделей, используемые в игровой индустрии.

ВВЕДЕНИЕ

Визуальная составляющая всех 3д игр это модели, практически всё что вы видите в игре, это модель, над которой старались моделлер и художник (иногда это один человек). Они тратят много времени на детальную проработку каждой модели и оптимизацию этих моделей для игрового движка. Однако у создания 3D моделей есть много особенностей, которые необходимо знать и учитывать при разработке и создании моделей.

I. ОСНОВНЫЕ СОСТАВЛЯЮЩИЕ 3D МОДЕЛИ

Любая 3D модель состоит из: Точек (vertex); Рёбер (edge); Плоскостей (tris, polygon, nGon).

Точки

Это простейший элемент модели, но это не просто точка, а точка, у которой, помимо её координат в пространстве, есть ещё один важный параметр - направленность.

Рёбра

2 точки образуют ребро. Это всегда прямая линия. Важно помнить, что у одного ребра всегда 2 точки, и у каждой есть своя вертекс нормаль.

Плоскости

Три точки создают треугольник. Три точки так же точки создают плоскость. В программах для моделирования полигон может состоять не только из трёх точек. Помимо треугольников, есть четырехугольники и многоугольники.

Все игровые движки и программы для 3D моделирования обладают единой общей особенностью — в них всё состоит из треугольников. Треугольники не удобны для моделирования, все программы и игровые движки научились триангулировать полигоны. У разных программ свои алгоритмы триангуляции.

II. СПОСОБЫ МОДЕЛИРОВАНИЯ

1. Полигональное моделирование
2. NURBS моделирование
3. Точное моделирование в Сапрах

Модели делятся по количеству полигонов. Чем больше полигонов, тем более детализирована модель, но требует больше ресурсов.

Власова Елизавета Андреевна, Выдрук Дарья Викторовна, студентки группы 724402, arthurdeep99@gmail.com dasha.vydruk@yandex.ru

Научный руководитель: Кукин Дмитрий Петрович, заведующий кафедрой вычислительных методов и программирования БГУИР, кандидат технических наук, доцент, Kukin@bsuir.by

III. СКУЛЬПТИНГ

В большинстве инструментов для моделирования цифровой скульптуры применяется деформация поверхности полигональной модели. Другие инструменты работают по принципу воксельной геометрии, объёмность которых зависит от используемого пиксельного изображения. Все инструменты по-разному деформируют геометрию модели, что облегчает и делает богаче процесс моделирования.

Ещё одна особенность заключается в том, что сохраняются несколько уровней детализации объекта. Если изменить поверхность модели на одном уровне, то эти изменения коснутся и других уровней, так как все уровни взаимосвязаны. Разные области модели могут иметь полигоны различной величины.

Главная особенность воксельной геометрии - она обеспечивает полную свободу над редактируемой поверхностью. Топология модели может постоянно изменяться в процессе её создания.

IV. РЕТОПОЛОГИЯ

Идея заключается в том, что у нас есть в качестве основы высокополигональная модель или хаотичный меш, вылепленный скульптингом. Этот меш используется в качестве опоры, чтобы начать рисовать геометрию, с которой можно работать. Есть несколько способов для ретопологии.

Это можно сделать при помощи специальных инструментов, модификаторов или плагинов в различных программах.

V. ЗАПЕКАНИЕ КАРТ НОРМАЛЕЙ

Запекание карт нормалей – это технология, используемая для имитации неровностей поверхности на объекте. Карты нормалей – это RGB изображения, где каждый из каналов интерпретируется в X, Y и Z координаты нормалей поверхности. Самый распространенный тип карты нормалей называется картой нормалей пространства касательных. Карта пространства касательных объекта используется гораздо реже в игровой индустрии, но, порой, ее использование позволяет решить некоторые проблемы. Существует множество способов убедиться в том, что запеченные карты нормалей будут выглядеть корректно в любом из игровых движков.

ВЗАИМОСВЯЗЬ ИГРОВОЙ ИНДУСТРИИ СО СФЕРАМИ НАШЕЙ ЖИЗНИ

Эта статья показывает вклад игровой индустрии в развитие и дополнение сфер нашей жизни. А также нововведения принятые игровой индустрией из этих же сфер.

ВВЕДЕНИЕ

В наше время уже заметно, что игровая индустрия одна из самых быстрорастущих IT индустрий и одна из самых разнообразных. За всё время в ней появилось множество многофункциональных инструментов и качественных игр. Это всё можно использовать не только для игр.

I. ВЗАИМОСВЯЗЬ КИНЕМАТОГРАФА С ИГРОВОЙ ИНДУСТРИЕЙ

Одна из основных задач кинематографа перенести зрителя в мир своего фильма. Сейчас это делается с помощью хромакеев и видеоредакторов. Использовать эти средства в создании сцен в фильмах достаточно трудный и затратный процесс, так как для этого нужно расставить хромакеи и отснять сцену целиком, после этого её надо загрузить в редактор и установить внешний вид сцены. Это забирает время съёмок у актёров и остального персонала, так как проверить результат их работы сразу нельзя.

Для решения поставленных проблем со средствами и временем в создание фильмов начали участвовать игровые движки. Самым популярным из них является Unreal Engine 4, обладающий множеством технологий и инструментов для работы с видео.

Sequencer - инструмент Unreal Engine 4, который решает задачи в реальном времени: 1) перемещение объектов сцены, 2) подбор объектов, 3) настраивание освещения, 4) установление фокусировки, 5) добавление спецэффектов. Его используют как хромакей в реальном времени, чтобы тратить на работу пару минут.

В создании фонов помогает шаблон UE4 под названием nDisplay: он позволяет проецировать изображение сразу на несколько экранов или поверхностей, создавая иллюзию присутствия актёров внутри этого пространства.

Stagecraft технология UE4, которая создаёт и меняет реалистичное окружение прямо на съёмочной площадке. Это позволяет сменить внешний вид любого объекта в кадре за пару кликов.

Филипчик Алексей Кириллович, студент 1 курса факультета ИТиУ БГУИРа, filipchick@mail.ru.

Гавриков Всеволод Валерьевич, студент 1 курса факультета ИТиУ БГУИРа, Sewa0105@mail.ru.

Пукало Евгений Анатольевич, студент 1 курса факультета ИТиУ БГУИРа, pukalo.evgen@gmail.com.

Научный руководитель: Коршикова Дарья Валерьевна, ассистент кафедры ВМиП БГУИРа, korshikova@bsuir.by

Сами разработчики, чтобы втянуть игрока в игру ещё сильнее, начали снимать кат-сцены - это эпизоды, которые прерывают обычный ход геймплея, чаще всего отбирают у игрока управление и показывают вам игровое кино, излагают сюжет.

II. ВЗАИМОСВЯЗЬ ИГРОВОЙ ИНДУСТРИИ С ВОЕННОЙ ОТРАСЛЮЮ.

С 1996 года в подготовку кадетов армии США вводят военные симуляторы – это игры, позволяющие создать различные боевые ситуации, в которых игрок должен применять разные тактические и стратегические умения. Внедрение таких симуляторов позволило значительно сократить расходы армии США на учебное оборудование для кадетов и военную технику, а также повысить качество будущих солдат.

III. ВЗАИМОСВЯЗЬ ИГРОВОЙ ИНДУСТРИИ С ОБУЧЕНИЕМ ПИЛОТОВ.

Пилоты самолётов – одна из самых опасных профессий. Для того чтобы стать пилотом надо быть сильным и здоровым человеком. Также пройти курс обучения, который переходит в игры. Авиасимуляторы позволяют сократить расходы на топливо, а также научить пилота действовать в опасных ситуациях без вреда пассажирам.

ВЫВОДЫ

Игровая индустрия показывает, что она может облегчить многие разработки и повысить качество обучения. Также виден её потенциал в улучшении текущих разработок как в играх, так и в сферах нашей жизни

1. <https://dtf.ru/cinema/99799-fundament-mandalorca-i-ubiyca-zelenogo-ekrana-kak-unreal-engine-ispolzuyut-v-kino>

2. <https://lenta.ru/articles/2006/02/16/games/>

МЕТОДЫ МОДЕЛИРОВАНИЯ

ВВЕДЕНИЕ

Существует несколько способов 3D моделирования, которые использует 3D моделлер: полигональное, сплайновое и NURBS моделирование. Они могут применяться как отдельно, так и комплексно.

I. ПОЛИГОНАЛЬНОЕ МОДЕЛИРОВАНИЕ

Если три точки координат задать как вершины и соединить их ребрами, то получится треугольник, который в 3D моделировании называют полигоном. В основном используют полигоны с 3 или 4 вершинами. Каждый полигон может иметь собственную текстуру и цвет, а объединив несколько полигонов можно получить модель любого объекта. Соединенные между собой полигоны образуют полигональную сетку или полигональный объект. Существуют высокополигональные и низкополигональные модели, отличающиеся количеством полигонов.

II. СПЛАЙНОВОЕ МОДЕЛИРОВАНИЕ

Сплайновое моделирование – это вид 3D моделирования, при котором модель создается при помощи сплайнов. Все сплайны сводятся к сплайновому каркасу, на основе которого уже будет создаваться огибающая трехмерная геометрическая поверхность. Преимущество сплайновых объектов в том, что они обладают гибкими настройками и всегда можно вернуться к изменению их формы. Сплайновое моделирование – более точное, и при масштабировании качество объекта не меняется.

III. NURBS МОДЕЛИРОВАНИЕ

NURBS моделирование или технология Non-Uniform Rational B-Spline – это технология неоднородных рациональных B-сплайнов, создание плавных форм и моделей, у которых нет острых краев, как у полигональных моделей. NURBS-кривые, используемые в данном моделировании, бывают двух видов: P (Point) кривые и CV (Control Vertex) кривые. Point кривые управляются вершинами, находящимися непосредственно на самой линии или объекте, а

Control Vertex кривые управляются точками, лежащими за пределами линии или объекта.

IV. СКУЛЬПТУРНОЕ МОДЕЛИРОВАНИЕ

После появления технологии пикселей (пикселей с координатой высоты) стало возможно разбивать полигональную модель на очень большое количество граней и работать с ней как с настоящей глиной или воском. Этот метод позволяет создавать мельчайшие детали на трехмерной модели, и чаще применяется для создания органических форм с высокой детализацией, а также для создания текстур нормалей.

V. ТЕКСТУРИРОВАНИЕ

Чтобы снизить нагрузку на GPU и в то же время создать как можно больше высококачественных моделей, используют текстурирование, один из базовых методов 3D-графики. Упрощенно – это накладывание изображения на полигон: к вершинам полигона привязываются конкретные пиксели плоского изображения, называемого картой текстуры. При рендеринге проводится интерполяция текстурных координат – к каждому пикселю в карте текстуры (или текстелю – так называют минимальную единицу текстуры трехмерного объекта) привязывают экранный пиксель на поверхности полигона

VI. ВЫВОДЫ

Таким образом, выбор того или иного метода зависит от вашей модели. Если необходимо создать простой объект, для которого гладкость не столь значительна, вы можете прибегнуть к стандартному полигональному моделированию. Если вам необходимо создать модель с большим количеством изгибов, лучшим решением будет использование сплайнового моделирования. Если же перед вами стоит задача создания реалистичной модели, к примеру, человека, то лучшим решением для более простой и удобной работы будет использование скульптурирования.

1. <https://koloro.ua/blog/3d-tehnologii/vidy-3d-modelirovaniya-poligonalnoe-splajnovoe-i-nurbs-modelirovanie.html>

Дюсов Александр Николаевич, студент 1 курса факультета ИТиУ БГУИРа, alexdusov04@gmail.com

Синяк Евгения Николаевна, студентка 1 курса факультета ИТиУ БГУИРа, evgenia.sinyak@yandex.by

Научный руководитель: Коршикова Дарья Валерьевна, ассистент кафедры ВМиП БРУИРа, korshikova@bsuir.by

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ГЕНЕРАЦИИ ПОДЗЕМЕЛИЙ НА ОСНОВЕ ПРОЦЕДУРНО СГЕНЕРИРОВАННЫХ УЗЛОВ ГРАФА

В данной статье описан алгоритм процедурной генерации уровней двумерного подземелья с заранее заданной структурой.

ВВЕДЕНИЕ

Данный алгоритм был написан для использования в собственных проектах. Алгоритм был составлен с применением литературы по дискретной математике, а реализация сделана в Unity 3D.

I. ОПИСАНИЕ АЛГОРИТМА

Узлы графа обозначают комнаты, а рёбра определяют связи между ними. Цель алгоритма — назначить каждому узлу графа форму и расположение комнаты таким образом, чтобы никакие две формы комнат не пересекались, и каждая пара соседних комнат могла соединяться дверьми. Граф генерируется на основе одного из простых связных графов из 5 узлов. Из четырех крайних узлов начинают генерироваться новые узлы (или небольшие графы) которые укладываются на плоскость так, чтобы их ребра не пересекались. Затем цикл может снова повторяться до тех пор, пока не будет получен граф необходимого размера.

II. КОМНАТЫ

Затем на место узлов становятся комнаты и начиная с первого узла сдвигаются друг к другу до тех пор, пока не состыкуются не пересекаясь друг с другом. Если в течении определенного промежутка времени не было найдено подходящее расположение, то у одной из конфликтующей комнат мы меняем форму и проверяем снова. В случае если после некоторого числа попыток не было получено положительного результата, конфликтный узел может быть удален если это не приведет к распаду подземелья на несколько несвязных частей. Следующим шагом расставляются двери согласно связям графа.

Жуковец Анна Александровна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, umei.kamila@gmail.com.

Доброгост Елизавета Валерьевна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, LizaDobrogost@outlook.com.

Научный руководитель: Гриневич Яна Григорьевна, ассистент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники ja.grinevich@bsuir.by.

III. РАЗНЫЕ ФОРМЫ ДЛЯ РАЗНЫХ КОМНАТ

Алгоритм позволяет задавать формы для каждой комнаты. Например, мы можем создать комнату начала уровня и комнату босса, которые должны иметь собственные наборы форм комнат, и один общий набор для всех других комнат.

IV. ЯВНО УКАЗЫВАЕМЫЕ ПОЗИЦИИ ДВЕРЕЙ

Алгоритм позволяет дизайнерам явным образом задавать возможные позиции дверей для отдельных форм комнат. Но создать шаблоны комнат таким образом, что двери в них могут быть практически где угодно. Благодаря этому мы накладываем на алгоритм меньше ограничений, поэтому часто он выполняется быстрее, а сгенерированные схемы могут казаться менее монотонными и более органичными. Для таких ситуаций есть возможность просто указывать длину дверей и насколько далеко они должны находиться от углов. Расстояние от углов — это своего рода компромисс между ручной расстановкой всех дверей и наличием дверей в любом месте.

V. ВЫВОДЫ

В ходе научной работы был составлен алгоритм, который позволяет генерировать подземелья случайным образом на основе случайного графа.

Список литературы

1. Иванов, О. А. ДИСКРЕТНАЯ МАТЕМАТИКА И ПРОГРАММИРОВАНИЕ В WOLFRAM MATHEMATICA / О. А. Иванов, Г. М. Фридман // СПб: Питер, – 2019. – 352с.
2. Новиков, Ф. А. Дискретная математика: Учебник для вузов / С. М. Львовский // СПб: Питер, – 2017. – 496с.

РАЗРАБОТКА ОСНОВНЫХ МЕХАНИК МОБИЛЬНОЙ СТРАТЕГИИ РЕАЛЬНОГО ВРЕМЕНИ НА UNITY

В работе рассматривается реализация основных механик мобильной стратегии реального времени на Unity.

ВВЕДЕНИЕ

В данный момент стратегии популярны на мобильных устройствах. Целью нашей работы является разработка основных универсальных механик стратегии реального времени, так как в дальнейшем это может значительно упростить разработку игр данного жанра. В качестве игрового движка был выбран Unity, так как он позволяет разрабатывать отдельные компоненты игры в виде независимых модулей, которые могут быть импортированы в другие проекты.

I. МЕХАНИКИ

В ходе работы были реализованы следующие основные механики:

1. Карта мира – область, покрывающая всю карту и дающая информацию о действиях как своих, так и вражеских юнитов.
2. Туман войны – игровая механика, скрывающая карту от игрока. Но игрок может исследовать закрытую территорию для обнаружения противников, получения ресурсов или информации о ходе игры путем перемещения своих юнитов. У каждого юнита есть определенный радиус, в пределах которого юнит видит определенную территорию. Вне этой окружности игрок видит темную зону, то есть ничего. Когда юнит начинает двигаться, окружность смещается игрок видит новую территорию, а то, что было видно до этого, покрывается туманом.
3. Система характеристик юнитов – есть возможность настраивать следующие характеристики юнитов: текущее здоровье и максимальное здоровье, мана, сила атаки, ско-

рость атаки, уклонение, скорость регенерации здоровья, скорость восстановления маны.

4. Система атаки – основана на настольных ролевых играх. Ее суть состоит следующим: имеется Dx кубик, где x – число граней. Кубик подбрасывается N раз. Каждый раз к начальному значению атаки юнита прибавляется значение, выпавшее на кубике.
5. Система кланов – юнитов можно разделять на команды, задавая целочисленный индекс. Это необходимо для реализации режимов игры, в которых принимают участие несколько игроков. Также индекс принадлежности команды определяют используемую модель для юнита.
6. Камера – скрипт, который позволяет управлять камерой: перемещать по свайпам и приближать (отдалять) с помощью сведения (разведения) двух пальцев.

II. ВЫВОДЫ

В ходе работы были реализованы основные механики стратегии реального времени для мобильных устройств. Данные механики могут быть использованы в дальнейшем для создания других различных проектов данного жанра. Также разработанные модули значительно ускоряют процесс прототипирования.

Список литературы

1. Хокинг Д. «Unity в действии. Мультиплатформенная разработка на C#».
2. Бонд Джереми Гибсон. «Unity и C#. Геймдев от идеи до релазации. 2-е издание».

Курко Кирилл Николаевич, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, kirillkurko2609@gmail.com.

Кресс Владислав Дмитриевич, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, vladkress@yandex.by

Научный руководитель: Рак Татьяна Александровна, старший преподаватель кафедры ВМиП, tatianarak@bsuir.by

РАЗРАБОТКА МОБИЛЬНОЙ ИГРЫ «MEMORY»

В работе рассматривается реализация мобильной «Memory» с использованием игрового движка Unity.

ВВЕДЕНИЕ

Последние годы на рынке мобильных игр все более и более популярными становятся казуальные и гиперказуальные проекты с низким порогом вхождения. Они набирают аудиторию за счет минималистичного визуального стиля, простого геймплея и возможности играть в них в любом месте. Поэтому было принято решение реализовать подобную игру на основе классической механики с известными многим правилами.

I. ОСНОВНАЯ МЕХАНИКА

В основе проекта лежит механика популярной игры «Memory», суть которой заключается в следующем: игрок видит стол, на котором расположено четное количество карточек. Карточка переворачивается по нажатию, и игрок видит изображение на ней. Если были открыты две карточки с одинаковыми изображениями, то они пропадают, а игрок получает очки. Игра завершается, когда на поле не остается карточек.

II. РЕАЛИЗАЦИЯ

В игре были реализовано следующее:

- основная механика, которая была описана в предыдущем разделе;
- механика подсчета очков – игрок получает большее количество очков, открывая пары одинаковых карточек подряд;
- уровни – возможность выбирать уровни для игры, различающиеся количеством карточек на столе;
- внутриигровая валюта – побеждая, игрок получает монеты, которые могут быть использованы в дальнейшем. Будет рассмотрена подробнее в одноименном разделе;
- настройки оформления – возможность выбирать цветовую тему и наборы карточек, используемых в игре;
- режимы - различные режимы игры, которые будут рассмотрены далее.

III. РЕЖИМЫ ИГРЫ

В игре представлены три основных режима:

1. стандартный - простой режим игры, количество ходов и время игры не ограничены;

2. на количество ходов - количество ходов ограничено, для победы игроку необходимо придерживаться наиболее оптимальной стратегии игры и не допускать ошибок. После победы игрок получает дополнительные очки, если количество совершенных им ходов меньше установленного в начале игры;
3. на время - время игры ограничено, одна открытая карта переворачивается обратно по истечении определенного времени. В данном режиме игрок также может получить дополнительные очки после победы, если время прохождения им уровня меньше отведенного.

IV. ВНУТРИИГРОВАЯ ВАЛЮТА

В игре присутствует внутриигровая валюта, представляющая собой монетки, которые могут быть использованы в дальнейшем.

Игрок зарабатывает монетки, побеждая в уровнях. На количество получаемых монет оказывают влияние следующие факторы:

- размер игрового поля;
- набранное количество очков;
- количество ходов (для всех режимов игры, но в режиме на количество ходов данный фактор является приоритетным);
- время прохождения уровня (только для режима игры на время).

Монеты используются для совершения внутриигровых покупок: приобретения тем цветового оформления и наборов оформления.

V. РЕЗУЛЬТАТЫ РАБОТЫ

В ходе работы была реализована игра «Memory» на игровом движке Unity. В игре были реализованы все описанные концепции и механики и достигнута цель по разработке простой и понятной с точки зрения геймплея игры на основе известной механики.

Список литературы

1. Хокинг Д. «Unity в действии мультиплатформенная разработка на C».

Курко Кирилл Николаевич, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, kirillkurko2609@gmail.com.

Научный руководитель: Рак Татьяна Александровна, старший преподаватель кафедры ВМиП, tatianarak@bsuir.by

ОКРУЖЕНИЕ В ИГРАХ

В работе приводится описание различных способов улучшения графического отображения окружения в компьютерных играх в зависимости от технических возможностей компьютера.

ВВЕДЕНИЕ

В связи с развитием игровой индустрии и улучшением технических возможностей персональных компьютеров, разработчики все чаще сталкиваются с повышениями требований к графической составляющей своих проектов. Сохраняя максимальную оптимизацию, игра должна обладать способностью рендерить графику в реальном времени и выглядеть приемлемо для человеческого глаза, т.е. обладать определённым уровнем графики. Зачастую высокобюджетные проекты стремятся к реалистичности и для ее обеспечения прибегают к различным технологиям и техникам обработки окружения.

I. ФОНОВОЕ ЗАТЕНЕНИЕ

Ambient occlusion (АО) — модель затенения, используемая в трёхмерной графике и позволяющая добавить реалистичности изображению за счёт вычисления интенсивности света, доходящего до точки поверхности. В отличие от локальных методов, как например затенение по Фонгу, ambient occlusion является глобальным методом, то есть значение яркости каждой точки объекта зависит от других объектов сцены. В принципе, это достаточно отдалённо напоминает глобальное освещение.

СГЛАЖИВАНИЕ

Сглаживание (англ. anti-aliasing) — технология, используемая для устранения эффекта

«зубчатости», возникающего на краях одновременно выводимого на экран множества отдельных друг от друга плоских или объёмных изображений. Основной принцип сглаживания — использование возможностей устройства вывода для показа оттенков цвета, которым нарисована кривая. В этом случае пиксели, соседние с граничным пикселем изображения, принимают промежуточное значение между цветом изображения и цветом фона, создавая градиент и размывая границу.

II. ФИЛЬТРАЦИЯ ТЕКСТУР

Фильтрация текстур — технология, используемая в компьютерной графике, преимущественно в 3D играх, с целью более детально выдавать на экран текстуры поверхности, которые находятся далеко или под сильным углом от персонажа или от камеры наблюдения. С появлением 3D-игр появилась проблема, которой в 2D-играх не было: вывод на плоский монитор трёхмерной картинке. Если объект находится параллельно плоскости экрана вблизи его, то одному пикселю соответствует один тексель. Если объект наклонен или находится вдали - на один пиксель приходится несколько текселей, и поскольку монитор имеет ограниченное количество пикселей, то цвет каждого рассчитывается из нескольких текселей при помощи процесса фильтрации.

Кушлеев Никита Юрьевич, студент 3 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, blackjetcat27@gmail.com.

Киселёва Наталья Михайловна, студентка 3 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, natakm99@yandex.ru.

Научный руководитель: Кужин Дмитрий Петрович, Кандидат технических наук, доцент, Kukin@bsuir.by

ИСПОЛЬЗОВАНИЕ ИГР В КАЧЕСТВЕ ИНТЕРАКТИВНОГО ПРИВЛЕЧЕНИЯ АБИТУРИЕНТОВ

В настоящее время игры являются уже не только развлекательными медиа, но и рабочим инструментом, с помощью которого можно влиять на игроков. На примере концепта игры для вступительной кампании БГУИРа можно рассмотреть возможность использования игр в качестве средства интерактивного привлечения абитуриентов и получения дополнительной информации об его устройстве посредством использования аспектов игровой рекламы и активного обучения.

ВВЕДЕНИЕ

Интерактивная подача информации нужна, чтобы в развлекательном формате помочь абитуриенту разобраться в незнакомых университетских порядках и помочь не растеряться в большом количестве текстов на сайте. Задача игрового проекта – подтолкнуть выпускника в нужное русло дальнейших поисков данных о заинтересовавшем его факультете/специальности после его краткого представления во время прохождения уровней.

I. ИСПОЛЬЗОВАНИЕ РЕКЛАМЫ

Аспект «рекламы» нужен для того, чтобы представить игроку факультеты и специальности, о выборе которых он не задумывался ввиду того, что не находил информацию о них из-за разрозненности на сайте или случайно пропустил в тексте. Таким образом, собрав все основополагающие данные о факультетах и связав их линейным сюжетом, можно наверняка проинформировать потенциального студента о всех возможных вариантах выбора места поступления. Далее абитуриент может найти уже более подробную и полную информацию о заинтересовавшей специальности на сайте. В общем подача рекламы с помощью игр не является новой практикой в индустрии, однако основной формат является фоновым показом продукта или компании, чего недостаточно для нормальной презентации чего-то нового, где требуются пояснения, и, следовательно, аспект «игрового обучения».

Метла Елизавета Александровна, студент кафедры вычислительных методов и программирования БГУИР, liza.metla@gmail.com.

Чирак Полина Викторовна, студент кафедры вычислительных методов и программирования БГУИР, polinachirak@gmail.com.

Ризмакова Екатерина Алексеевна, студент кафедры вычислительных методов и программирования БГУИР, katia.rizmakova@tut.by.

Олецкая Татьяна Валерьевна, студент кафедры вычислительных методов и программирования БГУИР, tanya.oletskaya@gmail.com.

Научный руководитель: Рак Татьяна Александровна, старший преподаватель кафедры вычислительных методов и программирования БГУИР, tatianarak@bsuir.by.

II. ИСПОЛЬЗОВАНИЕ ИГРОВОГО ОБУЧЕНИЯ

Игры относят к активному способу обучения, так как обучающий процесс происходит не только в форме пассивного слушания или чтения. Они имеют больше возможностей подстроиться под пользователя, ведь у него есть возможность влиять на происходящее на экране, к тому же информация, полученная в подобном интерактивном формате, обычно запоминается лучше, чем в процессе «зубрежного» запоминания. Задача игры-обучения – задать игроку мотивацию получить все возможные знания в процессе прохождения уровней. С помощью диалогов с фоновыми героями и встреч со специализированными персонажами-агитаторами, которые будут презентовать различные факультеты, игрок может узнать новую информацию об университете. Помимо этого, в процессе игры он ознакомится с порядком приемной комиссии, так как каждый уровень будет представлять собой какой-либо из этапов подачи документов.

III. ВЫВОДЫ

Представленный концепт игры может помочь потенциальным абитуриентам, в более простой и увлекательной форме, получить представление о факультетах и специальностях БГУИРа и определиться со своим выбором, чтобы в дальнейшем стать полноценным студентом.

1. <https://cyberleninka.ru/article/n/interaktivnost-i-interpassivnost-geymerov-v-videoigrovoyh-praktikah/viewer>

ШЛЕМ ДОПОЛНЕННОЙ ВИРТУАЛЬНОСТИ

I. Основной концепт технологии

Предлагаем идею для создания шлема, который будет использоваться в дополненной виртуальности. Дополненная виртуальность - это тип виртуальной реальности, где физический и цифровой объекты сосуществуют и взаимодействуют в реальном времени. Концепт шлема будет выполнен в полусферической форме, которая отчасти закрывает лобную и полностью затылочную, височную, теменную области, как и верхнюю часть лица, вплоть до носа, а также включать в себя следующие технологии: осознанных сновидений, быстрого засыпания и считывания нервных импульсов.

II. ОПИСАНИЕ ИСПОЛЪЗУЕМЫХ ТЕХНОЛОГИЙ

Осознанные сновидения: Осознанное сновидение - тип состояния сознания, при котором человек осознаёт, что видит сновидение, и может в той или иной мере управлять его содержанием. В это состояние человек будет попадать благодаря технологии, основанной на динамических индивидуальных диодных раздражениях в фазе быстрого сна. Технология будет встроена в глазную маску, закрепленную на основном корпусе шлема. Маска будет оснащена светодиодами и фотосенсорами, которые помогают контролировать сознание во время сновидения.

Быстрое засыпание: технология, воспроизводящая периодическое воздействие на низкой частоте в несколько герц: посредством звука, электрического тока, вибрации, воспринимаемое мозгом как помощь в засыпании. Шлем по бокам будет оснащен внешними динамиками, обладающими технологией 7.1 пространственного звучания, которые воздействуют посредством звука. Устройства для вибрации и испускания электрического тока будут внедрены в шлем по всей его области, кроме затылочной области головы и глазной маски.

Считывание нервных импульсов: в шлем будут поступать импульсы из среднего и продолговатого мозга, преобразовываясь в бинарный код. Затем пройдя обработку на процессоре и, проходя снова через шлем, изображение будет преобразовано в нервные импульсы, при попадании которых в мозг, сформируется изображение виртуального окружения.

III. Процесс погружения и сферы применения

Сам процесс погружения в дополненную виртуальность будет начинаться с подключения шлема к компьютеру и настройки его под себя: установки оптимальной громкости, частоты сигналов, яркости светодиодов и регулировки размеров. Ложимся в удобное положение, надеваем шлем, включаем игровой режим, в котором шлем вводит в состояние осознанного сна, и наслаждаемся процессом.

Данная технология может применяться в архитектуре: при коммуникации и обмене предложениями по архитектурному дизайну; в межличностной коммуникации: симулируя реальную встречу; в обучении работников компаний: для повышения осведомленности пользователей в опасностях определенных объектах; и в игровой индустрии: для полного погружения в игровой процесс. При входе в сеанс дополненной виртуальности человек попадёт в интерактивное виртуальное пространство, то есть он будет обладать возможностью управлять своими движениями в виртуальном пространстве. Если человек подумает о каком-то движении, например о ходьбе, мозг, на основе этих мыслей, пошлёт импульсы, которые примет шлем. В результате, человек будет идти в виртуальном пространстве. Так произойдёт с любым другим движением: бегом, прыжками, приседаниями и так далее. Также можно будет взаимодействовать с окружающим виртуальным миром, поднимать предметы, производить над ними всевозможные действия, которые осуществляются в действительности. Для распознавания голоса будет использоваться специальная система, которая преобразует импульсы, связанные с речью. В результате, вы сможете общаться внутри игры с другими игроками.

IV. Польза технологии

Польза технологии заключается в том, что импульсные токи, проникающие в полость черепа через глазницы, вызовут снижение загрузки коры головного мозга и усилят внутреннее торможение. Серотонин приведёт к снижению рефлекторной деятельности и эмоциональной активности. У человека, использующего шлем, наступит состояние осознанного сна, который может использоваться для лечения фобий, агрессивных состояний, депрессии и других проблем.

Анищенко Максим Павлович, Михайлов Ярослав Дмитриевич, студенты группы 924401, maksimych60@outlook.com, yaroslav.mihailov@bk.ru

Научный руководитель: Шатилова Ольга Олеговна, старший преподаватель кафедры вычислительных методов и программирования, магистр технических наук, shatilova@bsuir.by

SELF-MODIFYING CODE

Рассматривается альтернатива использования механизма самомодификации кода.

ВВЕДЕНИЕ

Считается, что выполняющаяся компьютерная программа отличается от данных, которые она обрабатывает. Однако это отличие размывается, когда компьютерная программа модифицирует сама себя. Модифицированная компьютерная программа затем выполняется как часть исходной программы.

SMC (Self-modifying code) - это изменение того что делает программа в режиме реального времени. В общем смысле, если программный код не делает одни и те же вещи при каждом его исполнении, то это и есть самомодифицирующийся код. SMC встречается во всех языках программирования в том или ином проявлении. В языках программирования высокого уровня есть множество различных механизмов, позволяющих достичь этого, к примеру: циклы, условные операторы, полиморфизм и т.д.

Однако есть ещё один тип самомодифицирующегося кода — это код, который изменяет свои собственные команды во время работы. Исполнение такого кода чаще всего ассоциируется с хакерским вмешательством, так как он усложняет исследование кода. Тем не менее, его можно применять и в легальных целях как инструмент, позволяющий достигать вещей, не доступных языкам высокого уровня из-за всяческого запрета такого подхода со стороны операционной системы.

По времени проведения модификации метод делится на:

- Модификация при инициализации – проводится один раз, перед запуском изменяемого кода;
- Модификация на лету (on-the-fly) – изменение состояния программы во время исполнения.

В обоих случаях изменение проходит непосредственно в машинном коде

I. ПРИНЦИП РАБОТЫ SMC

В отведенной на работу программы области памяти выделяют 3 основных сегмента:

- сегмент данных;
- сегмент стека;
- сегмент кода.

Все эти сегменты расположены в каком-то месте в RAM, и все они принадлежат страницам. Операционная система маркирует эти страницы различными правами доступа. Так некоторые страницы позволяют читать и записывать

информацию, некоторые позволяют исполнять себя и т.д. По умолчанию страница сегмента кода имеет права доступа чтения и исполнения, и операционная система заблокирует попытку вмешательства в сегмент кода на его изменение или перезапись. И, таким образом, имеется возможность только читать данные с этой страницы, либо исполнять данные на этой странице. В чём же хорош язык ассемблера так это в том, что он позволяет вручную разметить сегменты памяти и так же наделить их необходимыми правами доступа. Вследствие чего появляется возможность получить доступ к модификации кода.

II. КАК РАБОТАТЬ С SMC

SMC как таковой в полной мере доступен только на языках ассемблера. Однако его также можно использоваться как ассемблерные модули в многофайловых проектах программ высокого уровня.

Использование SMC подразумевает обладание возможностью оперирования кодом как данными. Для этого необходимо странице, где расположен сегмент целевого модифицируемого кода, предоставить права на запись. Делается это в объявлении сегмента кода. Таким образом появляется возможность изменять содержимое бит кода в режиме работы реального времени.

Модифицировать код можно как из самого модифицируемого сегмента кода, так и из других сегментов кода. Модификация кода обычно происходит в рамках блока какой-либо части подпрограммы. Чтобы безопасно оперировать кодом как данными, необходимо соблюсти следующие выведенные правила:

1. Размер модифицируемого блока должен быть больше или равен размеру записываемых машинных команд.
2. Модифицируемый блок не должен содержать в себе непарное количество инструкций работы со стеком.
3. После модификации кода для сохранения его непрерывности и целостности необходимо дополнить блок модифицируемого кода машинными кодами 90h, они же NOP (No Operation) – инструкции процессора, которая не делает ничего. Количество требуемых NOP равно разнице длин суммы машинных кодов модифицируемого блока кода и суммы машинных кодов записываемого кода в байтах.

Модификация кода происходит путём записи данных по адресу. Чтобы явно указать в коде ассемблера с какого момента необходимо начать проводить запись можно использовать метки. Метки – мнемоники ассемблера, которые сами по себе являются виртуальным адресом команды или данных. С их помощью можно «заключить» фрагмент интересующего кода в блок, тем самым во первых имея возможность указать место, с которого будет производиться запись, а во вторых узнать размер машинных команд в блоке, посчитав разницу адресов конца и начала блока.

III. Достоинства и недостатки SMC как подхода к написанию кода

SMC можно использовать для уменьшения размеров объектного кода. Так, к примеру, в коде алгоритма построения прямых Брезенхема можно уменьшить количество кода практически в 2 раза, определив по входным данным выбор оси инкремента и соответственно модифицировав инструкцию инкремента цикла, избавившись от дублирования основной части алгоритма.

Уменьшение кода это один из возможных подходов использования механизма SMC. Ниболее значимым является перспектива его применения в области алгоритмизации. SMC из-за своей абсолютной гибкости позволяет составлять с его использованием алгоритмы, которые подстраиваются под решение задачи во время выполнения. Или позволит составлять алгоритмы, которые составляют другие алгоритмы, которые в свою очередь выполняют требуемую задачу. И в данном случае у алгоритма всё не ограничивается набором возможных действий «на выбор», ведь алгоритм составляется из элементарнейших частей – машинных команд, которые в своих комбинациях имеют неограниченную вариативность в действиях для достижения желаемого результата.

SMC позволяет делать в режиме реального времени обновление программного обеспечения, транслировать байт-код в машинный код непосредственно во время работы программы для увеличения производительности.

Можно так же применять самомодификацию кода в критичных к скорости местах для ускорения работы. Так, например, во время исполнения можно уменьшить длину критического пути исполнения. Вместо установки и последующей многократной проверкой флагов с условными переходами можно всего лишь изменить адрес и тип перехода в машинном коде.

Панкевич Даниил Сергеевич, студент специальности информационных систем и технологий в игровой индустрии БГУИР, daniil.pankevich@gmail.com.

Научный руководитель: Рак Татьяна Александровна, старший преподаватель кафедры вычислительных методов и программирования БГУИР, tatianarak@bsuir.by.

К недостаткам SMC в программировании следует отнести отсутствие безопасного обращения с кодом, сложности с разработкой, тяжесть в длительной поддержке программного продукта, ограниченность в размерах целевых блоков кода. Последнего можно избежать использованием динамической компиляции.

Современные языки программирования и среды разработки с их компиляторами или трансляторами не позволяют раскрыть весь потенциал самомодифицирующегося кода. Языки Perl, PHP и Python позволяют программе создавать новый код во время выполнения и выполнять его, но не позволяют самомодифицироваться существующему коду. Существуют аналогичный механизм модификации работы программы, который достигается путём изменения указателей на функции. В действительности при таком подходе машинный код не изменяется.

В наше время самомодификация кода применяется в технологии JIT-компиляции (Just-in-time compilation). Эта технология позволяет увеличить производительность программных систем, использующих байт-код, путём компиляции байт-кода в машинный код или в другой формат непосредственно во время работы программы.

Для эффективной и безопасной работы с SMC необходимо разработать специализированный язык программирования и среду разработки для работы с этим языком соответственно. Тогда и только тогда появится возможность опираться на SMC не только как на особый «прием», а как на самодостаточный подход к проектированию систем.

Подводя итоги закреплю за этой концепцией два понятия:

- Code-oriented design (COD) – подход, при котором машинный код воспринимается и используется как данные;
 - Code-oriented programming (COP) – методология программирования, основанная на видении кода программы как набора начальных данных, осуществление изменения которых в режиме реального времени позволяет достигнуть поставленной задачи.
- Использование

1. Скэнлон Лео Джон. Персональные ЭВМ IBM PC и XT. / Программирование на языке ассемблера – Москва: Издательство «Радио и связь». Редакция переводной литературы, 1989.
2. Э. Танненбаум. Архитектура компьютера / Structured Computer Organization. – 5-е изд. – СПб.: Питер, 2013. – С. 476. – 884 с. – ISBN 978-5-469-01274-0

ТЕОРИЯ ВЕРОЯТНОСТЕЙ В АЗАРТНЫХ ИГРАХ. РАЗРАБОТКА СИМУЛЯТОРА ПАРАДОКСА МОНТИ ХОЛЛА

Возможно ли математически обосновать и рассчитать вероятность выигрыша в различных азартных играх? Меняет ли исход игры новая информация?

ВВЕДЕНИЕ

В современном мире азартные игры стали распространённым видом развлечения. Как правило, выигрыш в них зависит не от навыков играющих, а от случая. Возможно ли выстроить правильную модель поведения игрока, опираясь на теорию вероятностей, чтобы увеличить шансы на победу?

В качестве примера азартной игры была использована задача из популярного телевизионного шоу «Let's Make a Deal» 1960-х и 1970-х годов, на котором впервые были озвучены её правила.

I. УСЛОВИЕ ЗАДАЧИ

Новый легковой автомобиль находится за одной из трех дверей. За двумя остальными стоят козы. Чтобы выиграть автомобиль, необходимо угадать, за какой из дверей он находится. При этом отсутствуют какие-либо предварительные сведения, чтобы провести различие между дверями. После выбора двери, одна из оставшихся откроется – там находится одна из двух коз. Теперь доступны следующие действия: выбрать другую дверь или остановиться на изначально выбранном варианте.

II. В ЧЁМ СМЫСЛ ПАРАДОКСА?

Большинство участников шоу интуитивно не находят преимущество в выборе другой двери, предполагая, что вероятность выигрыша в обоих случаях одинакова. На самом деле, шансы выиграть автомобиль удваиваются при смене выбора. Действительно, изначально вероятность того, что за дверью находится автомобиль, составляет 33,3% во всех трех случаях, но после того, как нам показывают, за какой из дверей стоит коза, вероятность того, что автомобиль находится за оставшейся дверью, достигает 66,6%.

III. СИМУЛЯТОР НА UNITY

Для демонстрации данного парадокса и предоставления статистики нами было разработано приложение-симулятор в среде Unity (см.рис.1.).

Позняк Анна Сергеевна, студент 2 курса ФИТиУ, pazniak.anna@hotmail.com.

Кайдалова Дарья Дмитриевна, студент 2 курса ФИТиУ, d.kaidalov@mail.ru.

Научный руководитель: Волковец Александр Иванович, кандидат технических наук, доцент кафедры ВМиП, volkovets@bsuir.by.



Рис. 1 – Интерфейс симулятора

Для того, чтобы продемонстрировать увеличение шанса на выигрыш при изменении двери, была добавлена функция генерации 1000 попыток (см.рис.2.).



Рис. 2 – Статистика для 1000 попыток

IV. ВЫВОДЫ

Этот парадокс наглядно показывает, как легко попасть в ловушку, относясь к неслучайной информации так, как если бы она была случайной. Таким образом, в некоторых азартных играх новая информация меняет их исход, поэтому знание теории вероятностей значительно увеличивает шансы на победу.

1. Unity [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/Manual/ExpertGuides.html> – Дата доступа: 16.04.2020
2. Naumanerik.livejournal [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/Manual/ExpertGuides.html> – Дата доступа: 16.04.2020
3. Pinnacle [Электронный ресурс]. – Режим доступа: <https://www.pinnacle.com/ru/betting-articles/Betting-Psychology/the-monty-hall-problem/22W2ZME4LVEY4FKL> Дата доступа: 16.04.2020

ИСПОЛЬЗОВАНИЕ UNITY ПРИ СОЗДАНИИ ИГРЫ В ЖАНРЕ ПОШАГОВОЙ СТРАТЕГИИ

Рассматривается реализация пошаговой стратегии с элементами RPG на игровом движке Unity. Предлагается рассмотреть основные этапы при разработке игры.

ВВЕДЕНИЕ

Для решения задачи о реализации игры в жанре пошаговой стратегии предлагается использовать игровой движок Unity, работа которого основана на создании игры любой сложности и размеров. Для реализации всей логики игры и механик, будет использована среда разработки Visual Studio.

I. ИГРОВАЯ ОБЛАСТЬ НА СЦЕНЕ В UNITY

При создании сцены и игровой области, где будут происходить все действия нашей игры используются объекты игры. Они необходимы для создания главного меню, отдельных уровней, настроек. Каждый файл сцены можно рассматривать как отдельный уровень. В каждой сцене размещается окружение, препятствия, декорации, таким образом происходит составление игры из объектов. Далее необходимо размещения всех предметов в сцене для этого используется инструмент Transform. Добавочно можно напрямую задавать значения в Инспекторе для точного размещения и поворота.

Далее необходимо разместить все объекты в сцене для этого используется инструмент Transform, он позволяет позиционировать все на сцене. Добавочно можно напрямую задавать значения в инспекторе для точного размещения и поворота.

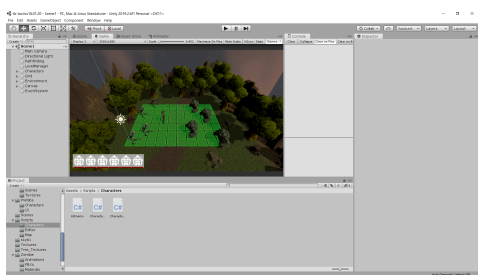


Рис. 1 – Игровая сцена

II. РЕАЛИЗАЦИЯ ИГРОВОЙ ЛОГИКИ

Для реализации игровой логики и создания скриптов используется среда разработки Visual

Studio на языке программирования C#. В ходе разработки необходимо было создать поле с клетками, по которым будут передвигаться персонажи игрока и сами враги. Для этого был реализован скрипт под названием Grid с логикой реализации игрового поля. Персонажи игрока и компьютера (т.е. враги) для каждого из них есть свое поведение и количество клеток хода персонажа, для этого использовались скрипты под названием PlayerController и AiEnemy. В данных скриптах реализуется логика движения и атаки, если дистанция позволяет атаковать врага он будет атакован, если же нет, то необходимо подойти ближе.

За все передвижения персонажей по клеткам, отвечает скрипт AiPath в нем используется алгоритм поиска кратчайшего пути к цели. Это необходимо для просчета нынешнего положения персонажа и того места куда ему нужно пойти за определенное количество шагов для него допустимых, исходя из этого, игрок может просчитывать ходы и придумывать свои тактики для выигрыша.

III. Выводы

Предлагаемый мной игровой движок Unity3d действительно мощный, с помощью которого можно создавать настоящие, работающие игры, что можно увидеть по скриншотам указанным выше. Данный игровой движок позволяет развивать идеи от простого к сложному. Сначала использование прототипирования с несложной логикой и обычными фигурами такими как: капсулы, кубы и др. До полноценного игрового продукта, с персонажами, окружением, анимацией и звуковым сопровождением.

1. Хокинг, Д. Юнити в действии. Мультиплатформенная разработка /. – 2020. – №1. – С. 140-200.
2. Э. Стилльмен, Д. Гринн. Изучаем C// /. – 2020. – №3. – С. 223-279.

Печурихин Виктор Олегович, студент специальности информационные системы и технологии в игровой индустрии БГУИР, vityapechurikhin@gmail.com.

Гриневич Яна Григорьевна, ассистент кафедры ВМиП, ja.grinevich@bsuir.by.

Научный руководитель: Рак Татьяна Александровна, старший преподаватель БГУИР, tatianarak@bsuir.by

ШИФРОВАНИЕ ХАФФМАНОМ

В работе рассматривается программная реализация алгоритма Хаффмана и его применение для сжатия и распаковки данных.

ВВЕДЕНИЕ

Коды Хаффмана – оптимальный префиксный код, используемый в сжатии данных без потерь. Коды Хаффмана являются результатом работы алгоритма Хаффмана, разработанного Дэвидом Хаффманом в 1952 году.

I. ПРИНЦИП РАБОТЫ

Все символы обычного текстового файла представляются одним (ASCII) или двумя (Unicode) байтами. В обеих кодировках каждый символ кодируется одинаковым количеством битов.

Символ	Двоичное представление	Десятичное представление
A	01000001	65
B	01000010	66
C	01000011	67
...
X	01011000	88
Y	01011001	89
Z	01011010	90

Главный принцип алгоритма Хаффмана состоит в создании специальных кодов, назначаемых каждому из символов сжимаемого сообщения таким образом, что наиболее часто встречаемому символу назначается самый короткий код Хаффмана.

Таким образом, часто встречаемые в сообщении символы будут уже представлены не восьмью, но уже меньшим числом бит, тем самым уменьшая размер всего сообщения.

Другое важное свойство кодов Хаффмана – это то, что ни один код не является префиксом другого кода. Делается это с целью избежать неопределенности при чтении потока данных.

Алгоритм Хаффмана можно разбить на несколько простых этапов:

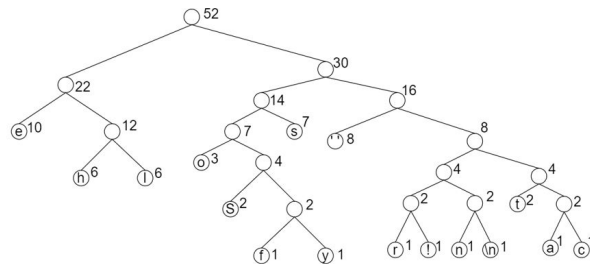
- Создание частотной таблицы символов сообщения.

Плотников Владислав Вадимович, студент 1 курса факультета компьютерных систем и сетей Белорусского государственного университета информатики и радиоэлектроники, vadimovich74@gmail.com

Кресс Владислав Дмитриевич, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, vladkress@yandex.by

Научный руководитель: Шатилова Ольга Олеговна, старший преподаватель кафедры ВМиП, shatilova@bsuir.by

- Построение дерева Хаффмана на базе частотной таблицы символов.
 - Обход дерева Хаффмана и присваивание соответствующего кода Хаффмана каждому символу исходного сообщения.
- Для сообщения «She sells seashells on the shore of the Seychelles!» дерево Хаффмана будет выглядеть следующим образом:



Как только дерево построено, можно приступить к шифрованию сообщения. Движемся от начала корня ко всем листьям, запоминая «маршрут»: переход в левое поддерево – ноль, а в правое – единица. То есть для символа *e* код будет 00, а для символа *h* – 010. Как можно заметить, самые встречаемые символы находятся ближе к корню дерева, а значит «маршрут» до них (т.е. код Хаффмана) будет самым коротким.

II. РЕЗУЛЬТАТЫ РАБОТЫ

- Был изучен алгоритм Хаффмана, его свойства, достоинства и недостатки.
- Было реализовано программное средство сжатия и распаковки данных методом Хаффмана на языке Java.

Список литературы

1. Роберт Лафоре «Структуры данных и алгоритмы Java».
2. Khalid Sayood «Introduction to Data Compression».

РЕАЛИСТИЧНЫЙ РЕНДЕРИНГ НА ОСНОВЕ МЕТОДА ТРАССИРОВКИ ЛУЧЕЙ

Рассматривается базовая механика метода трассировки лучей – технологии построения реалистичных изображений, достоинства и недостатки.

ВВЕДЕНИЕ

Одним из методов рендеринга в компьютерной графике является метод «бросания лучей» (ray casting), при котором растровое изображение строится на основе замеров пересечения лучей с визуализируемой поверхностью в пространстве. Этот термин впервые использовался в компьютерной графике в 1982 году в работе Скотта Рота [1], который применил его для описания метода рендеринга CSG-моделей.

Первый алгоритм рейкастинга, используемый для рендеринга, был представлен Артуром Аппелем в 1968 году [2]. В основе рейкастинга лежит идея генерировать лучи из точки наблюдения сцены, один луч на пиксель, и находить самый близкий объект, который блокирует путь распространения этого луча. Используя свойства материала и эффект света в сцене, алгоритм рейкастинга может определить затенение данного объекта.

I. ПРИНЦИП МЕТОДА ТРАССИРОВКИ ЛУЧЕЙ

В основе метода трассировки лучей лежит идея генерировать лучи из точки наблюдения сцены, один луч на пиксель, и находить самый близкий объект, который блокирует путь распространения этого луча. Используя свойства материала и эффект света в сцене, алгоритм рейкастинга может определить затенение данного объекта.

В реальной природе источник света испускает луч света, который, проходя через пространство, пересекает какую-либо преграду, которая прерывает распространение этого светового луча.

Алгоритм трассировки лучей (ray tracing) применяет рейкастинг для расчета первичных пересечений луча с объектами сцены и дополняет его генерацией дополнительных лучей для формирования световых бликов, теней, отражений, тем самым повышая уровень фотореалистичности изображения.

Для построения первичного луча и определения первых пересечений с объектами сцены вводятся понятия источника лучей и плоской области обзора. В основе модели лучевого эмиттера лежит механизм упрощенной камеры-обскуры (рис. 1) с бесконечно малым отверстием, через которое свет проникает на область обзора.

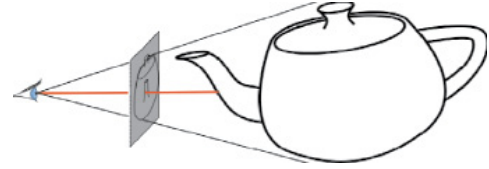


Рис. 1 – Принцип построения изображения

II. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ АЛГОРИТМА ТРАССИРОВКИ ЛУЧЕЙ

Определим плоскую область обзора как массив пикселей $\{pxm\}$, где n и m – количество пикселей в ширину и высоту соответственно, используя предварительно подготовленные структуры данных.

Установим начало координат в точку $O(x_0, y_0, z_0)$, центр камеры в точку $C(x_c, y_c, z_c)$. Зададим фокусный центр камеры точкой $F(x_F, y_F, z_F)$. Вектор $\vec{w} = \vec{FC}$ определяет направление камеры.

Для корректного определения направления распространения луча от позиции камеры к центру каждого пикселя плоской области обзора понадобится сформировать ортогональный базис $\{\vec{u}, \vec{v}, \vec{w}\}$ при помощи операций векторного произведения:

$$\vec{u} = \vec{w} \cdot \vec{y}_{0,1,0} \quad (1)$$

$$\vec{v} = \vec{u} \cdot \vec{w} \quad (2)$$

Направление первичного луча \vec{R} (рис. 2) определяется по следующим формулам:

$$\alpha = \tan\left(\frac{fov_x}{2}\right) \cdot \left(\frac{i - \left(\frac{n}{2}\right)}{\frac{n}{2}}\right) \quad (3)$$

$$\beta = \tan\left(\frac{fov_y}{2}\right) \cdot \left(\frac{\left(\frac{m}{2}\right) - j}{\frac{m}{2}}\right) \quad (4)$$

$$\vec{R} = \vec{OC} + \frac{\vec{w} + \alpha \vec{u} + \beta \vec{v}}{|\vec{w} + \alpha \vec{u} + \beta \vec{v}|} \quad (5)$$

где α, β – величины смещения луча по осям X, Y области обзора; i, j – целочисленные координаты пикселя, для которого генерируется луч; fov_x, fov_y – горизонтальный и вертикальный углы обзора камеры.

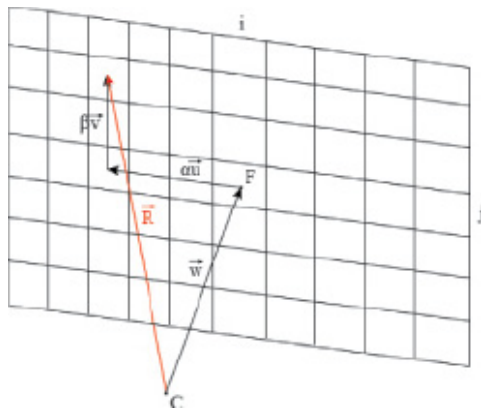


Рис. 2 – Вычисление направления распространения луча от позиции камеры к центру каждого пикселя координатной плоскости

Пусть $\vec{d} = \frac{\vec{w} + \alpha \vec{u} + \beta \vec{v}}{|\vec{w} + \alpha \vec{u} + \beta \vec{v}|}$.

Таким образом, можно записать векторное уравнение пучка лучей, исходящих из точки O с направляющими векторами \vec{d} :

$$\vec{R}_{i,j} = \vec{OC} + t \vec{d}, t \geq 0 \quad (6)$$

где t – расстояние от начала луча до любой точки на нем; \vec{d} – вектор направления распространения луча.

Рак Татьяна Александровна, магистрант кафедры информационных технологий автоматизированных систем БГУИР, tatianarak@bsuir.com.

Научный руководитель: Алёхина Алина Энодиевна, доцент кафедры экономической информатики БГУИР, кандидат технических наук, доцент, ae.alekhina@gmail.com.

III. Выводы

К достоинствам метода трассировки лучей можно отнести следующие положения:

- возможность рендеринга гладких объектов без аппроксимации их полигональными поверхностями;
- вычислительная сложность метода слабо зависит от сложности сцены;
- отсечение невидимых поверхностей, перспектива и корректные изменения поля зрения являются логическим следствием алгоритма.

Серьезным недостатком метода является производительность. Метод растеризации и сканирования строк использует когерентность данных, чтобы распределить вычисления между пикселями. В то время как метод трассирования лучей каждый раз начинает процесс определения цвета пикселя заново, рассматривая каждый луч наблюдения в отдельности.

1. Roth, S. D. Ray casting for modeling solids // Computer Graphics and Image Processing. – 1982. – №18. – P. 109-144.
2. Appel, A. Some techniques for shading machine renderings of solids // AFIPS spring joint computer conference. IBM Research Center, Yorktown Heights, N.Y. – 1968. – P. 37-45.

ТЕОРИЯ ВЕРОЯТНОСТИ В КОНТЕКСТЕ ОСНОВ ИГРОВОГО БАЛАНСА. ВЕРОЯТНОСТЬ И СЛУЧАЙНОСТЬ

В ходе работы были изучены возможные проблемы, с которыми имеют дело геймдизайнеры при столкновении игрока с вероятностными системами.

ВВЕДЕНИЕ

В наше время теория вероятностей - наука, представляющая значительный общетеоретический и прикладной интерес и находящая применение в множестве сфер. Разработка игр - не исключение, ведь вероятность победы и проигрыша - это именно то, на чем строится вовлечение в игру и построение игрового баланса. Однако у большинства людей совершенно отсутствует чутьё на подлинные шансы. Даже сделав случайные элементы игр абсолютно справедливыми, большее количество игроков будет всё равно воспринимать игру как несправедливую. Поэтому геймдизайнеры должны не только вникать в истинные вероятности, но и предполагать, как игроки будут воспринимать эти вероятности в наших играх и насколько это восприятие будет отличаться от истины.

I. ОСНОВНЫЕ ПРОБЛЕМЫ

- Погрешность выборки: редкие, но производящие большое впечатление события воспринимаются как более вероятные. Игроки переоценивают свои возможности, потому что помнят только самые грандиозные победы, и не важно, что их было намного меньше, чем поражений.
- Погрешность эгоизма: игроки всегда считают себя более ловкими и умными, чем другие, поэтому выигрывая в 70% случаев вместо предсказанных ими 95%, они расстраиваются и считают это неправильным.
- Погрешность отнесения: случайный положительный исход игроки всегда интерпретируют как их собственное знание механизма, опыт в игре или как результат правильно выбранной стратегии - как всё, но только не как случайность. В то время, как отрицательный исход - это всегда невезение,

Филистович Оксана Дмитриевна, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, oksanprivet@gmail.com

Рогач Александра Юрьевна, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, aleksandra.rogach256@gmail.com

Научный руководитель: Волковец Александр Иванович, кандидат технических наук, доцент кафедры ВМиП, volkovets@bsuir.by

ошибка или игра и вовсе работает против них.

- Фиксация: игроки придают слишком большое значение первому числу, которое они увидят в игре, заикливаются и переоценивают его и свои шансы, если это число даёт положительный эффект для дальнейшей игры.
- Ложный вывод Монте-Карло — предположение, что люди не ожидают, что кажущиеся им маловероятными события могут происходить несколько раз подряд. К этому относятся как положительные для игрока события, например, падение монет именно так, как выгодно игроку, так и отрицательные события, такие как проигрыш в игре 3-4 раза подряд, когда игроку кажется, что система работает против него.
- Ложный вывод «горячей руки» — предположение, что если игрок достиг успеха в чём-либо 2-3 раза подряд, то скорей всего он достигнет его и в следующие разы.

II. ВЫВОД

Исходя из этого, можно сделать вывод, что у игроков крайне искажённое понимание вероятностей. Если показывать игроку их настоящие шансы и разрабатывать игру которая честно генерирует случайные числа — игроки будут жаловаться, потому что из-за искажённого восприятия вероятности им будет казаться, что в игре «что-то не так». Таким образом, геймдизайнерам остаётся либо жаловаться на игроков, не зная теории вероятностей, либо обращать эти знания себе на пользу и использовать их в разработке, чтобы делать ещё более вовлекающие и интересные игры.

Список литературы

1. Шрайбер Я. «Принципы игрового баланса».

КРИВАЯ 4-ГО ПОРЯДКА ЛЕМНИСКАТА БЕРНУЛЛИ

В работе рассмотрена замечательная кривая 4-го порядка лемниската Бернулли, приведены собственные свойства кривой.

ВВЕДЕНИЕ

Лемниската Бернулли не утратила своего значения, так как применяется в технике, в частности, в качестве переходной кривой на закруглениях малого радиуса, как это имеет место на железнодорожных линиях в горной местности и на трамвайных путях. В связи с этим, необходимо знать длину и область площади, ограниченной данной кривой. Для решения этой задачи необходимо вывести уравнения кривой; вычислить длину кривой; вычислить область площади, ограниченной кривой; вывести уравнения касательной к кривой; вывести формулы кривизны кривой; изобразить кривую на плоскости. Целью работы является исследование лемнискаты Бернулли.

I. ОПРЕДЕЛЕНИЕ КРИВОЙ

Лемниската Бернулли – плоская алгебраическая кривая 4-го порядка. Определяется как геометрическое место точек, произведение расстояний от которых до двух заданных точек (фокусов) постоянно и равно квадрату половины расстояния между фокусами[1].

Лемниската по форме напоминает восьмёрку или символ бесконечности. (см.рис.1.)

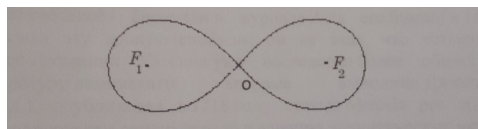


Рис. 1 – Лемниската Бернулли, где F_1 и F_2 – Фокусы, O – узловая (двойная) точка.

Точка, в которой лемниската пересекает саму себя, называется узловой или двойной точкой.

II. СОБСТВЕННЫЕ СВОЙСТВА КРИВОЙ

1. Кривая является геометрическим местом точек, симметричных центру равносторонней гиперболы относительно её касательных;
2. Отрезок биссектрисы угла между фокальными радиусами-векторами точки лемнискаты равен отрезку от центра лемнискаты до пересечения её оси с этой биссектрисой;

3. Материальная точка, движущаяся по лемнискате под действием однородного гравитационного поля, пробегает дугу за то же время, что и соответствующую хорду;
4. Площадь полярного сектора $\phi \in [0, a]$, при $0 \leq a \leq \frac{\pi}{4}$:

$$S(a) = \frac{c^2}{2} \sin 2a \quad (1)$$

В частности, площадь каждой петли $2S(\frac{\pi}{4}) = c^2$, то есть площадь, ограниченная кривой, равна площади квадрата со стороной $c\sqrt{2}$;

5. Перпендикуляр, опущенный из фокуса лемнискаты на радиус-вектор какой-либо её точки, делит площадь соответствующего сектора пополам;
6. Длина дуги лемнискаты между точками $\phi_1 = 0$ и $\phi_2 = \phi$ выражается эллиптическим интегралом I рода:

$$L(\phi) = c \int_0^\phi \frac{d\phi}{\sqrt{1 - 2 \sin^2 \phi}} = \frac{c}{\sqrt{2}} F(\theta, \frac{1}{\sqrt{2}}), \quad (2)$$

где $2 \sin^2 \phi = \sin^2 \theta$

В частности, длина всей лемнискаты

$$4L(\frac{\pi}{4}) = 2c\sqrt{2}K(\frac{1}{\sqrt{2}}) \approx 5,9c \quad (3).$$

ЗАКЛЮЧЕНИЕ

В ходе работы была описана теория лемнискаты Бернулли: вычислена длина кривой; вычислена площадь области, ограниченной данной кривой; выведены уравнения кривой; описаны собственные свойства кривой.

Список литературы

1. Савелов А. А. Плоские кривые. – М., 1960. – 293 с.
2. Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления. – М., 1969, т.1. -607 с.
3. Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления. – М., 1969, т.2. -800 с.

Сильванович Юлия Васильевна, студентка 2 курса инженерно-экономического факультета БГУИР, uliasilvanovic440@gmail.com.

Научный руководитель: Баженова Ирина Владимировна, старший преподаватель кафедры вычислительных методов и программирования БГУИР, канд. физ.-мат. наук, доцент, liv@bsuir.by.

ОСОБЕННОСТИ ДИНАМИЧЕСКОЙ СИМУЛЯЦИИ ОГНЯ

В работе приводится описание физики процесса горения и особенностей визуального восприятия огня, дается описание структуры симуляции огня, приводится анализ различных техник моделирования, анимации и визуализации огня.

ВВЕДЕНИЕ

Дым и огонь могут значительно влиять на визуальное восприятие объектов сцены, а также влиять на свойства других объектов сцены. По этой причине огонь и дым являются важными составляющими во многих прикладных областях, таких как симуляция полетов, ландшафтный дизайн, анимация и киноиндустрия. Анимация и визуализация данного явления сложной задачей и представляет определенный научный интерес. Симуляция трехмерного огня в режиме реального времени находит свое применение в различных интерактивных приложениях. Среди интерактивных приложений, анимации огня наиболее востребованы в видеоиграх. В компьютерной графике довольно часто требуется найти компромисс между скоростью и реализмом. Основной проблемой рендеринга в реальном времени является поиск таких алгоритмов, которые позволяют получить достаточную реалистичность, при которой частота кадров будет не менее минимального порога. Целью работы является обзор основных физических особенностей поведения и внешнего вида огня, анализ алгоритмов для моделирования этих атрибутов.

I. ФИЗИКА ОГНЯ

В разговорном русском языке нет четкого смыслового разделения между словами "пламя" и "огонь" [1], однако в зарубежной литературе присутствует четкое разделение понятий огонь (fire) и пламя (flame). Также в некоторых технических русскоязычных словарях приводится трактовка этих понятий. Несмотря на то, что стандарт СТ СЭВ 383–87 [2] уже устарел, в нем

дается точное определение для ключевых терминов, используемых в данной работе. **Огонь** – процесс горения, сопровождающийся пламенем или свечением. **Пламя** – зона горения в газовой фазе с видимым излучением. Среди процессов химических веществ бывают случаи, когда вещество сгорает без пламени [3]. Обычно люди воспринимают горящий объект, словно объект горит сам по себе. Однако, на самом деле пламя образует не сам объект, а топливо, которое он выделяет в окружающую среду. Топливо подымается над поверхностью объекта из-за тепла, испаряется, вступает в контакт с кислородом и воспламеняется. Таким образом, огонь нуждается в совместном присутствии трех элементов: топлива, тепла и кислород [4].

ЗАКЛЮЧЕНИЕ

Анализ литературных источников показал наличие множества междисциплинарных связей в решениях данной проблемы. Для успешного решения задачи симуляции огня необходима разработка специализированных решений, сфокусированных на моделировании ограниченного числа атрибутов огня.

Список литературы

1. Пламя. – URL: <https://ru.wikipedia.org/wiki/Пламя> (дата обр. 28.04.2020).
2. СТ СЭВ 383–87. Пожарная безопасность в строительстве. Термины и определения. – 07.1987.
3. Огонь. – URL: <https://ru.wikipedia.org/wiki/Огонь> (дата обр. 28.04.2020).
4. US Army. Publication TM 5–315: Firefighting and Rescue Procedures in Theaters of Operation / US Army. – US Army Corps of Engineers Internet Publishing Group., 05/1971. – Chap. 3: characteristics, chemistry, and physics of fire.

Стаховский Антон Владимирович, магистрант 2 курса факультета компьютерных систем и сетей Белорусского государственного университета информатики и радиоэлектроники, anton.stakhouski@gmail.com.

Научный руководитель: Кукин Дмитрий Петрович, доцент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, заведующий кафедрой, kukin@bsuir.by.

ЖАНРОВАЯ КЛАССИФИКАЦИЯ ВИДЕОИГР

Как появляются и классифицируются жанры видеоигр.

Создание видеоигр – относительно молодая индустрия. Она зарождалась стихийно, первые видеоигры создавались лишь как занятные игрушки. Однако индустрия бурно развивалась и очень быстро появилось множество самых разных игр. Сначала это были игры на аркадных автоматах, после на консолях и ПК, а теперь и на мобильных платформах. Уже в 80-ые годы возникла потребность как-то классифицировать игры, отмечать их сходства и различия. Одной из первых классификаций стало разделение игр по жанрам.

Традиционно, видеоигры принято разделять на следующие жанры: стратегии, ролевые игры, симуляторы, песочницы, игры действия, головоломки, платформеры и многие другие. Но на основе чего принято такое разделение и как оно появилось? Одной из первых работ, в которой рассматривалась классификация видеоигр является “Искусство проектирования компьютерных игр” Криса Кроуфорда [1]. В ней Кроуфорд предложил классификацию основанную на геймплее: разделение на игры Навыка и действия и на Стратегии или, иначе, Когнитивные игры. Первая категория призвана включать в себя все игры, в которой основой геймплея является скорость и точность реакции. Вторая – игры, в которых игровой процесс зависит в первую очередь от “умственных усилий”, а скорость реакции, если и требуется, то не является определяющей.

Уже Кроуфорд отметил что отдельные “семьи” игр рождаются путём повторения удачного опыта игры-предшественника. То есть после успеха некоторой новой игры многие хотят повторить её успех и копируют из неё геймплей, так рождаются многие игры клоны. Часть из них вводит новые механики, но при этом они все сохраняют схожесть с игрой-предком. Похоже, что именно этот механизм является способом зарождения новых жанров. Сначала жанр могут называть просто набором игр клонов игры-предка, как, например было с клонами Doom. Постепенно, с ростом количества и увеличением разнообразия игр-клонов у них появляется новое цепкое название, которое характеризует общие черты геймплея данных игр. Таким образом клоны

Doom сейчас именуется как Шутеры от первого лица.

Такое стихийное появление жанров на данный момент привело к тому что не существует однозначности в количестве жанров и их границах. Периодически возникают вопросы, к какому жанру отнести ту или иную игру. Таким образом вопрос о том как классифицировать существующие жанры весьма актуален. Необходимо однако отметить, что хотя на данный момент многие согласны с определением жанра как наличия определённого вида геймплея, на самом деле существовало два различных подхода к самому определению жанра: людологический и нарратологический [2]. Суть первого подхода была описана выше, а нарратологический подход именуется жанром способ повествования, аналогичный жанрам театра и кино. Для жанра в людологическом смысле нарратологи предлагали термин “тип” игры.

Одна из актуальных классификаций жанров предложена сайтом gameisart.ru [3]. Путём декомпозиции игровых механик были получены 15 элементарных жанров, которые затем были объединены в три группы: игры Информации, Действия и Контроля. При этом элементарные жанры, которые находятся на краю одной группы, имеют общие черты с соседней группой. Таким образом, все игры можно представить в виде определённого набора элементарных жанров, что может отметить её как преимущественно принадлежащей к одной группе, так и игры имеющей черты разных групп. Эта классификация достаточно полна и позволяет описать жанр игры одним словосочетанием, давая определённое название комбинации элементарных жанров.

1. Chris Crawford. The Art of Computer Game Design [Электрон. ресурс] – Internet Archive, 1984 - Режим доступа: <https://archive.org/details/artofcomputergam00chri/>
2. Gerald Voorhees. Genre Troubles in Game Studies: Ludology, Agonism, and Social Action [Электрон. ресурс] – Режим доступа: <https://www.kinephanos.ca/2019/genre-troubles-in-game-studies-ludology-agonism-and-social-action/>
3. Кириллов Александр. Классификация жанров компьютерных игр [Электрон. ресурс] – Режим доступа: <https://gameisart.ru/janr.html>

Тымуль Андрей Евгеньевич, студент ФИТиУ БГУИР, andreixinbox@mail.ru.

Мельникова Виолетта Валентиновна, студентка ФИТиУ БГУИР, vitaemaximus@gmail.com.

Научный руководитель: Шатилова Ольга Олеговна, старший преподаватель кафедры вычислительных методов и программирования БГУИР, магистр технических наук, shatilova@bsuir.by

РАЗРАБОТКА МОБИЛЬНОЙ ИГРЫ С ИСПОЛЬЗОВАНИЕМ МЕХАНИКИ 2048 «МАЛЫШКА, ТЕБЕ КРЫШКА»

В ходе работы был разработан концепт казуальной игры и реализована механика 2048 на Unity.

ВВЕДЕНИЕ

С каждым годом люди становятся всё ленивее, поэтому при выборе игры они руководствуются правилом «Чем проще, тем лучше», прослеживающееся во всех казуальных играх, не требующих высокой концентрации и большого количества времени, затраченного на «разобраться, что да как». Этими факторами и обусловлена популярность игры 2048 и множества её интерпретаций. Поэтому, чтобы внести в игровой мир что-то новое, приходится устанавливать новую планку и придумывать оригинальную концепцию, что мы и сделали в данной работе.

I. В ЧЁМ СУТЬ И КАК ИГРАТЬ

Проблемы есть у всех, но у девушек их всегда больше. Победить их в игре – первый шаг к тому, чтобы победить их в жизни. На старте игры на поле масштабированием 2 квадрата заполняются рандомно проблемами и их решениями.

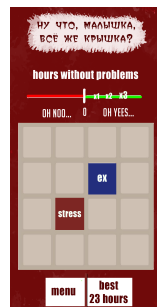


Рис. 1 – Старт игры

Игрок свайпает вверх, вниз, вправо или влево, смещает в выбранную сторону все объекты. Если два объекта при смещении соприкасаются и это соответствует правилам объединения, то эти два объекта сливаются в один квадрат; при каждом свайпе игрока в произвольном пустом квадрате с вероятностью 50/50 появляются новые проблемы и решения. Задача игрока набрать как можно больше очков.

Филистович Оксана Дмитриевна, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, oksanprivet@gmail.com

Курко Кирилл Николаевич, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, kirillkurko2609@gmail.com

Научный руководитель: Рак Татьяна Александровна, старший преподаватель кафедры ВМиП, tatianarak@bsuir.by

II. КАК ВСЁ РАБОТАЕТ

Правила объединения: два квадрата сливаются в один, если

- соприкасаются две проблемы > появляется удивленно-недовольное лицо;
- соприкасаются два решения проблем > появляется довольное лицо;

Проблемы, решения и результаты никак друг с другом при свайпах не взаимодействуют. Правила работы шкалы:

- при совмещении двух проблем ползунок смещается со своего текущего положения влево по шкале на 1 деление;
- при совмещении двух решений ползунок смещается со своего текущего положения вправо по шкале на 1 деление;

Конец игры наступает, когда игровое поле полностью заполняется игровыми элементами.

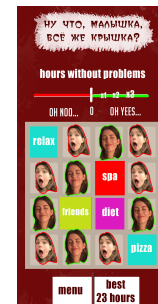


Рис. 2 – Заполненное поле

III. РЕЗУЛЬТАТЫ РАБОТЫ

Реализовав всё это с помощью Unity, мы получили конечный продукт – игру «Малышка, тебе крышка».

Список литературы

1. Хокинг Д. «Unity в действии. Мультиплатформенная разработка на C».

ГЕНЕРАЦИЯ ПАРОЛЕЙ В БАНКОВСКОМ ДЕЛЕ

Рассматриваются методы генерации случайных чисел в компьютерных технологиях, а также в банковском деле в частности. Анализ наиболее популярных сервисов для генерации паролей.

ВВЕДЕНИЕ

Случайные числа и случайность имеют множество применений в криптографии, науке, играх, искусстве, а также во многих других областях. Для разных задач, требуется генерация разного качества, именно по этой причине существует потребность в разнообразных методах генерации случайных чисел. Генераторы случайных чисел (ГСЧ) делятся на два основных типа: Генераторы псевдослучайных чисел (ГПСЧ) и Генераторы случайных чисел (ГСЧ). С развитием информационных технологий и интернета, возросла потребность в качественной генерации случайных чисел, не только у специалистов, но и у обычных людей. В данной публикации мы рассмотрим основные методы генерации случайных чисел их преимущества и недостатки, а также их практическое применение.

I. ГЕНЕРАТОРЫ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ (ГПСЧ)

Из-за простоты и дешевизны для выработки случайных чисел чаще всего используются генераторы, созданные как соответствующие программы на ЭЦВМ. С помощью этих программ по некоторому алгоритму получают последовательности чисел. Алгоритм построен так, что знаки 0 и 1 появляются в среднем одинаковое число раз и отсутствует зависимость между появлениями этих знаков и сформированными из них многозначными числами. Числа получаемые с помощью таких генераторов называются псевдослучайными или квазислучайными. Простейшим примером может служить разложение в десятичную дробь иррациональных чисел. Качественные требования, предъявляемые к ГПСЧ:

- Достаточно длинный период, гарантирующий отсутствие заикливания последовательности в пределах решаемой задачи. Длина периода должна быть математически доказана
- Эффективность – быстрота работы алгоритма и малые затраты памяти
- Воспроизводимость – возможность заново воспроизвести ранее сгенерированную последовательность чисел любое количество раз
- Портруемость – одинаковое функционирование на различном оборудовании и операционных системах

- Быстрота получения

Никакой детерминированный алгоритм не может генерировать полностью случайные числа, он может только аппроксимировать некоторые их свойства.

II. ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ (ГСЧ)

Генераторы истинно случайных чисел генерируют последовательности случайных чисел на основе измеряемых, хаотически изменяющихся параметров физического процесса. Работа таких устройств часто основана на использовании надёжных источников энтропии таких, как тепловой шум, дробовой шум, фотоэлектрический эффект, квантовые явления, погодные явления и другие физические процессы. Простейшим примером генератора может служить подбрасывание монетки или игральной кости. Основная проблема аппаратных генераторов случайных чисел – это их относительная медленная по сравнению с ГПСЧ работа.

Алгоритмы проверки PIN

На данный момент, в основном, используются следующие 2 алгоритма проверки PIN: Visa PVV и IBM 3624 PIN offset.

Visa PVV

Данный алгоритм первоначально был разработан платежной системой Visa, но, в настоящее время является рекомендованным алгоритмом проверки PIN как для карт Visa, так и для MasterCard. В основе данного алгоритма лежит значение PVV (PIN verification value), которое является криптограммой, получаемой на основе следующих величин:

- Номер карты (далее PAN)
- Индекс ключа проверки PIN (PIN verification key index, далее, PVKI)
- Ключ проверки PIN (PIN verification key, далее, PVK)
- Сам PIN-код карты

Для получения PVV формируется блок из PAN (последние 11 цифр, кроме контрольного числа карты), PVKI, PIN (строго, первые 4 цифры), который зашифровывается с помощью PVK, после чего из него, с помощью специальной

функции, извлекаются 4-х значное число, которое и является значением PVV. Данное значение PVV является эталонным для проверки PIN кода. Т.е. при получении операции с введенным PIN для его проверки на основании PAN, PVKI, PVK формируется новое значение PVV и сравнивается с эталонным PVV для карты. Если значения совпадают, то PIN считается верным, если не совпадают – неверным.

К особенностям данного алгоритма можно отнести следующие «ограничения»:

- Принципиальная невозможность восстановления PIN из значения PVV
- Использование PIN-кода размером строго 4 цифры

IBM 3624 PIN offset

Данный алгоритм первоначально был разработан компанией IBM для использования в банкоматах IBM 3624. Как именно планировалось его использовать, история умалчивает, а авторы статьи не знают, но, в данном случае, это не принципиально. В настоящее время данный алгоритм считается устаревшим, но достаточно успешно используется по нескольким причинам:

- карточные системы «старых» регионов (Западная Европа, Северная Америка) достаточно консервативны и, во многом, работают на «достаточно» старых системах, что их вполне устраивает
- данный алгоритм позволяет восстановить значение PIN кода из проверочного значения (см. далее), что м.б. весьма полезно при определенных условиях

В основе данного алгоритма лежит значение PIN offset (PIN verification value), которое является криптограммой, получаемой на основе следующих величин:

- Контрольное значение (Validation data, далее VD) – некоторое значение (обычно – часть номера карты, но это не обязательно)
- Децимализационная таблица (Decimalization table, далее DT)
- Ключ проверки PIN (PIN verification key, далее, PVK)
- Сам PIN-код карты

Для простоты дальнейшего описания под ключом проверки PIN в случае метода IBM 3624 PIN offset будем совокупность ключа PVK и значения таблицы децимализации DT.

Для получения PIN offset контрольное значение VD зашифровывается с помощью ключа

PVK, после чего из полученного значения с помощью таблицы децимализации DT получается блок из 16 десятичных цифр.

Из полученного блока берутся первые N цифр, где N – длина PIN (метод IBM 3624 позволяет проверять PIN с длиной до 16 цифр), далее из каждой цифры PIN по модулю 10 вычитается соответствующая цифра полученного блока. Полученное значение и будет значением PIN offset.

Проверка PIN

Терминология

Для упрощения дальнейшего описания введём некоторые термины:

- PIN блок – значение PIN кода карты, некоторым специальным образом упакованной в блок из 8 байт. Стоит пояснить, что никакого шифрования в данном процессе не используется. Способы упаковки, в данном случае, бывают разные, но это не принципиально.
- Зашифрованный PIN блок – значение PIN блока, зашифрованное по алгоритму DES/3DES с помощью ключа (ключа терминала, банка, платежной сети, пр.), специально выделенного для целей шифрования PIN блока.
- Проверочное значение PIN – PVV или PIN offset в зависимости от того, какой метод проверки PIN используется.
- Дополнительные данные проверки PIN – данные, кроме PIN и проверочного значения PIN, необходимые для проверки PIN в соответствии с алгоритмами Visa PVV/IBM 3624 PIN offset в соответствии со списком, приведенным в описании алгоритмов (см. выше).

Требования платежных систем

В части проверки PIN можно указать следующее:

- Открытые значения PIN и PIN блока не должны никаким образом передаваться, храниться или обрабатываться вне специально отведенных программно аппаратных комплексов (HSM на стороне процессинговых систем или ЕРР и некоторых других страшных аббревиатур на стороне терминальных устройств(банкоматы, POS-терминалы и пр.)).
- Зашифрованный PIN блок не должен храниться после завершения операции в системах, отвечающих за онлайн/оффлайн обработку транзакций (есть ещё системы, от-

вечающие за выпуск самих карт, их это требование не касается).

Просмотр PIN

Как мы уже определились ранее, для проверки PIN нам необходимы следующие данные:

- Сам PIN, который мы будем проверять
- Проверочное значение PIN
- Дополнительные данные проверки PIN

С PIN все достаточно просто. Как уже указано выше, открытое значение PIN мы получить не можем ни при каких условиях. Таким образом, нам остается только зашифрованный PIN блок. В дополнение к нему нам нужен ключ для его расшифровки. Назавем этот ключ РПК (PIN protection key, термин взят из документации на HSM фирмы SafeNet).

Далее необходимо определиться с проверочным значением PIN и дополнительными данными.

Первый вариант – это хранение проверочного значения на магнитной полосе карты после поля Service Code. Модифицированную версию ISO 7813 с указанием того, где хранится PVV, можно посмотреть здесь. По приведенному описанию формата треков стоит добавить, что под 5- и символьным значением PVV подразумевается следующая последовательность 1 символ PVKI и 4 символа самого PVV, а для PIN offset – значение PIN offset для PIN из 5 цифр. Если PIN имеет отличную от 5 цифр длину, то размер PIN offset, соответственно, изменится. Какие плюсы у этого метода. Безусловно – возможность проверять PIN для любого, кто будет иметь необходимые для проверки ключи. Здесь стоит заметить, что при запуске нового карточного продукта в платежную сеть, обычно, передаются ключи, на которых выпущена карта. Таким образом, при использовании данного метода возможность проверки PIN появляется как у самого эмитента карты, так и у платежной сети. К недостаткам такого метода можно отнести то, что данный вариант делает PIN карты статическим до тех пор, пока карта не будет перевыпущена.

Второй вариант – это хранение проверочного значения в некотором хранилище, обычно, БД системы, отвечающей за выполнение проверок при авторизации карты. В этом случае при проверке PIN необходимо извлечь проверочное значение из этого хранилища, а уже потом, выполнять проверку, используя это значение. Как следствие, при использовании данного метода, невозможно выполнять проверку PIN во внешней системе (в той же платежной системе) и она м.б. выполнена только в той системе, которая имеет доступ к хранилищу проверочных значений. Однако, такая система позволяет изменять PIN код

карты без каких либо затрат на смену пластика (для чего это нужно, что при этом необходимо сделать и какие после этого м.б. проблемы, описывать не буду, т.к. это находится за рамками данной статьи).

Независимо от того, каким образом и кем (эмитент карты или платежная сеть) была получена вся необходимая информация, сама проверка PIN выполняется на HSM, который для выполнения проверки получает ключ РПК в защищенном виде, ключ проверки PIN в защищенном виде, зашифрованный PIN блок, проверочное значение PIN и дополнительные данные проверки, в ответ на что возвращается только результат проверки: верный PIN, неверный PIN, прочая ошибка. Т.е. в процессе проверки система, отвечающая за авторизацию, с самим открытым значением PIN кода никак не соприкасается.

III. БАНКОВСКОЕ МОШЕННИЧЕСТВО

Реквизиты банковской карты – это секретная информация. Если она попадет в руки не тех людей, вы можете потерять деньги. Реквизиты – это всё, что написано на карте: номер из 16 цифр, имя и фамилия владельца, срок действия и трехзначный код безопасности на обратной стороне. Для удобства мы отнесём к реквизитам и смс-код, который присылает вам банк, когда вы платите в интернете или переводите деньги.

По правилам платёжных систем реквизиты нельзя сообщать посторонним. Если банк узнает, что ваши реквизиты попали в чужие руки, то сразу заблокирует карту. Однако кое-что сообщать всё-таки можно. Разберёмся на примере, какую информацию содержит Ваша пластиковая карта.



1. Наименование и/или логотип банка-эмитента – наименование и/или логотип банка, выпустившего платёжную карточку.
2. Чип – микросхема, вшитая в пластик и выполняющая ту же роль, что и магнитная полоса, т.е. обеспечивающая проведение расчётов с помощью платёжной карточкой.
3. Номер карточки – 16 цифр, идущих в ряд.

4. Имя и фамилия держателя платёжной карточки.

5. Срок действия – указывается на карточке в формате ММ/ГГ и показывает до какого момента времени (включительно) действительна карта.

6. Бренд платёжной системы.

7. Магнитная полоса – полоса, содержащая необходимые данные для проведения расчётов с использованием платёжной карточки.

8. CVV2 (CVC2) – трехзначный код на оборотной стороне карточки, обеспечивающий дополнительную безопасность, предназначенный специально для проведения расчётов в сети.

9. Полоса для подписи – место, где держатель ставит свою подпись.

Что может сделать мошенник, который завладел вашими реквизитами?

Фролов Ярослав Ильич, студент кафедры информационных технологий автоматизированных систем БГУИР, iaraslav_frolov@mail.ru.

Лемеза Марк Викторович, студент кафедры информационных технологий автоматизированных систем БГУИР, suslik2034@gmail.com.

Научный руководитель: Гуринович Алевтина Борисовна, заместитель декана по научно-методической работе БГУИР, кандидат физ.-мат. наук, доцент кафедры ВМиП, gurinovich@bsuir.by.

Ничего	Почти ничего	Залплатить в некоторых интернет-магазинах	Забронировать отель или авто, привязать карту к Гугл-плюс, залплатить на Литресе	Залплатить где угодно в интернете, сделать любой платеж или перевод
Номер карты	Номер карты Имя и фамилия	Номер карты Имя и фамилия Срок действия	Номер карты Имя и фамилия Срок действия Код безопасности	Номер карты Имя и фамилия Срок действия Код безопасности Код из смс



IV. Вывод

Таким образом, на нынешний момент ваши деньги хорошо защищены. Но в каждом деле есть исключения. Поэтому огромная просьба: не передавайте реквизиты своих карт посторонним людям и соблюдайте технику безопасности.

Список использованных источников

1. Иванова, В. М. Случайные числа и их применение, 1984
2. <https://www.belinvestbank.by/individual/page/moshennichestvo-v-seti-internet>
3. Visa Payment Technology Standards Manual

РАЗРАБОТКА ИНДИ-ИГРЫ НА ДВИЖКЕ UNITY 3D

В статье ставится задача рассмотреть процесс создания инди-игры с помощью игрового движка Unity3d. Рассмотрены частые проблемы в процессе разработки.

ВВЕДЕНИЕ

Инди-игры — это проекты, созданные независимыми авторами. То есть людьми, которые работают не на крупных издателях, а на себя. Как правило, инди-студии состоят всего из нескольких, но встречаются и разработчики-одиночки. Инди-студии — главный источник экспериментальных проектов в игровой индустрии. Ввиду своей независимости инди-разработчики не имеют ограничений. Это позволяет разработчикам сфокусироваться на особенностях игры, будь это красивая графика, необычная геймплейная механика или захватывающий сюжет, и отшлифовать их.

I. СОЗДАНИЕ ИГРЫ

Для эффективной работы мы создали удобную для нас коммуникативную среду, которая включала ряд инструментов:

1. Для таск-менеджмента и распределения работы мы использовали Trello по системе To Do, In progress, Done. Это намного повысило эффективность работы над нашей игрой, так как всегда можно увидеть кто над чем работает, какой объем работ еще предстоит выполнить и т.д.
2. Для хранения материалов мы использовали Google Drive.

В качестве игрового движка был выбран Unity. Unity — один из самых популярных игровых движков. Большинство игр для мобильных платформ создаются именно на Unity. Ключевую роль при выборе сыграла его доступность, удобство и умение работать с данным движком.

Любая игра начинается с идеи. В самом начале у нас было несколько идей для будущего проекта, поэтому сначала мы приступили к разработке прототипов. Так из всех идей мы выбрали наиболее удачную. Мы остановили выбор на игре в жанре match3. Match3 можно назвать самым популярным жанром на рынке мобильных

игр. Основная задача игрока — составлять комбинации из 3-х и более одинаковых объектов, хаотично расположенных на игровом поле. Игры жанра match3 — одни из самых простых и прекрасно подходят тем, кому требуется просто расслабиться на некоторое время.

Следующим этапом стало написание игровой документации и усовершенствование прототипа — добавление новых игровых механик и игрового интерфейса. На этом этапе мы окончательно определились со стилем и визуальным оформлением нашей игры. Было решено делать игру в нежно-розовых тонах, а в качестве объектов для перемещения были выбраны милые котки. Так же у нашей игры появилось название — Move the cat.



Рис. 1 — Главный экран

II. ВЫВОДЫ

По завершении работы над нашей игрой нам нужно было узнать мнение со стороны, мнение живых игроков, которые бы впервые увидели игру. В основном мы получили похвальные отзывы о визуальном стиле, звуковом сопровождении, интересности игрового процесса, понятности и удобности интерфейса.

Хобта Анна Эдуардовна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, ank hobta@gmail.com.

Доброгост Елизавета Валерьевна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, Lizadobrogost2000@tut.by

Научный руководитель: Гриневич Яна Григорьевна, ассистент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники, ja.grinevich@bsuir.by .

АНАЛИЗ СПОСОБОВ ОПТИМИЗАЦИИ ОТРИСОВКИ КАДРОВ В РЕАЛЬНОМ ВРЕМЕНИ

ВВЕДЕНИЕ

Начинающие специалисты трёхмерного моделирования и разработчики игр регулярно сталкиваются с задачами по оптимизации трёхмерных моделей с целью уменьшения “стоимости” отрисовки кадров. К сожалению, не всегда специалисты разбираются в том, как же отрисовывается кадр, из-за чего специалисты допускают грубые ошибки. [1].

I. КАК УСТРОЕНА ОТРИСОВКА КАДРА. ВЫЗОВ ОТРИСОВКИ

Отрисовка кадра - комплексный процесс, направленный на поэтапный расчёт и отрисовку кадра в реальном времени, который происходит внутри игрового движка. Отрисовка кадра имеет стоимость, которая измеряется необходимым временем и ресурсами. Для оценки времени рендера недостаточно знать только количество отрисованных кадров в секунду. Вызов отрисовки (draw call) — это одна графическая команда, которая должна что-то отрисовать. Сложность кадра можно оценивать именно с помощью вызовов отрисовки. Чем их больше, тем дольше они обчитываются, тем медленнее выполняется 1 кадр, тем меньше кадров в одной секунде и тем меньше итоговая частота кадров. [2–3].

II. БАТЧИНГ (BATCHING)

Динамический батчинг работает следующим образом: движок сам анализирует геометрию и сшивает те объекты, которые подходят под лимиты. Обычно у динамического батчинга очень жёсткие лимиты; он не позволяет сшивать тяжёлую геометрию и занимается в основном мелкими деталями в кадре. Например, если от стены динамически отделились мелкие осколки, то они они, скорее всего, будут сшиты динамическим батчингом в каждом отдельном кадре. Статический же батчинг происходит «заранее». Для этого нужно указать что этот объект статичен, и никогда не будет изменен или сдвинут. Отличие от динамического батчинга в том, что он позволяет перерабатывать огромное количество объектов. За один батч Unity может сшить

объектов общим количеством до 64 000 треугольников. Кроме того, динамический батчинг будет пытаться пересобрать геометрию почти в каждом новом кадре, — а значит будет постоянно увеличивать время расчёта. [4–5].

III. УРОВЕНЬ ДЕТАЛИЗАЦИИ (LOD)

Level of Detail (LOD, или уровень детализации) — простая техника, которая позволяет уместить огромное количество объектов в кадре, не отнимая время на обработку лишней геометрии. [6].

IV. ПЕРЕРАСЧЁТ ПРОЗРАЧНОСТИ (ALPHA OVERDRAW)

Рендер-пайплайн должен оптимизировано рассортировать и нарисовать сцену — обработать шейдеры, геометрию, текстуры и цветовые фильтры. К тому же он должен учитывать, что объекты могут быть прозрачными. Чем больше площадь прозрачности в кадре, и чем больше прозрачные объекты наслаиваются друг на друга — тем дольше рендер-пайплайн будет рисовать итоговый пиксель, ведь ему придётся каждый раз его перерисовывать. Это называется Alpha overdraw. [7].

ИТОГ

Как мы выяснили, практически нет беспроблемного варианта в случае с отрисовкой кадра: технологии, которые позволяют что-то оптимизировать, не работают по нажатию нужной галочки. Всегда нужно понимать, для чего мы используем тот или иной инструмент. Нынешнее железо — даже мобильное, — может обрабатывать огромное количество треугольников. Объёмы памяти и скорость растут. Чипы на мобильных устройствах и их батареи, которые перегреваются при высоком FPS, тоже становятся всё совершеннее.

Список литературы

1. <https://simonschreibt.de/gat/renderhell/>
2. Куданков Д. Оптимизация: почему время важнее полигонов
3. Гладили Д. Проблемы и решения при создании графики для мобильного шутера Guns of Boom

Шек Даниил Сергеевич, студент 3 курса ФИТиУ, grinder8705@gmail.com.

Папакуль Денис Сергеевич, студент 3 курса ФИТиУ, denis.papakul.dp@gmail.com.

Научный руководитель: Кукин Дмитрий Петрович, к.т.н, заведующий кафедрой ВМиП, kudin@bsuir.by

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ПАЛАСИОСА В АРКАДНОМ ПРОЕКТЕ

В данной работе предоставляются результаты применения алгоритмов Паласиоса в аркадном проекте.

ВВЕДЕНИЕ

Стоит подумать об искусственном интеллекте – и в уме возникает множество ассоциаций: от простых моделей поведения, таких как преследование или убегание от игрока, до методов машинного обучения или процедурной генерации. Одним из программистов, специализирующийся на разработке и реализации искусственного интеллекта и игровой логики, является Хорхе Паласиос.

Хорхе Паласиос – программист- профессионал, автор книги «Unity 5.x. Программирование искусственного интеллекта в играх». Свою книгу он предоставляет как исчерпывающий справочник, помогающий расширить навыки программирования искусственного интеллекта в играх, где рассматриваются основные приёмы работы с агентами, программированием перемещения и навигации в игровой среде, принятием решений и координацией.

I. ПРИМЕНЕНИЕ АЛГОРИТМОВ ПАЛАСИОСА

В данной работе были предприняты успешные попытки применения алгоритмов Паласиоса в аркадном проекте. Главной целью игры является захват и перемещение одного из трех персонажей в определенную зону несколько раз. Сами же персонажи пытаются убежать от игрока, но если один из них был схвачен, то оставшиеся персонажи попытаются спасти пленника. (см.рис.1.)

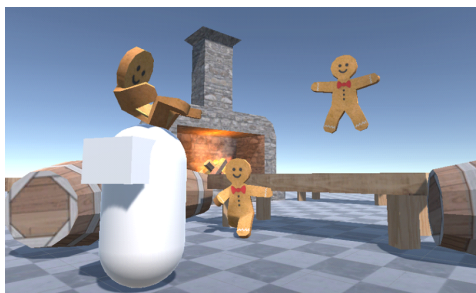


Рис. 1 – Демонстрация игрового процесса

Основными алгоритмами для реализации искусственного интеллекта были: достижение цели и уход от погони, блуждание вокруг, уклонение от стен и алгоритм принятия решения [1].

Достижение цели и уход от погони – небольшой алгоритм, основанный на учете расстояния до цели и вычислении направления движения. Является основой перемещения персонажей.

Блуждание вокруг – алгоритм, суть которого состоит в том, что неигровой персонаж раз в n-ное количество секунд случайно определяет точку перемещения. Алгоритм помогает оживить игровой процесс и создать не предсказуемые ситуации.

Уклонение от стен – достаточно интересный алгоритм, суть которого в том, что по направлению движения персонажа отбрасываются лучи, которые в случае пересечения с препятствием заставляют персонажа перемещаться в новую позицию. Алгоритм позволяющий устранить большинство проблем, связанных с застреванием персонажей в статических объектах.

Алгоритм принятия решения является важнейшим из всех перечисленных. Он представляет из себя систему, состоящую из возможных состояний персонажей, переходов между этими состояниями и условиями переходов. Его роль заключается в изменении логики поведения персонажей на основе входящих данных, что позволяет вдохнуть жизнь в персонажей.

II. ВЫВОДЫ

Таким образом использование алгоритмов Паласиоса описанных в его книге достаточно для реализации аркадного проекта с достаточно интересным игровым процессом, предоставляющий вызов игроку.

1. Паласиос Х. Unity 5.x. Программирование искусственного интеллекта в играх: пер. с англ. Р. Н. Рагимова. - М.: ДМК Пресс, 2017. -272 с.: ил.

Жулковский Павел Игоревич, студент БГУИР, shadowpi1999@gmail.com.

Ярошевский Алексей Евгеньевич, студент БГУИР, aleksriox@gmail.com.

Научный руководитель: Шатилова Ольга Олеговна, старший преподаватель кафедры вычислительных методов и программирования БГУИР, shatilova@bsuir.by.

Секция "Электрические цепи"

Председатель: канд. тех. наук, доцент, зав. каф. ТОЭ Журавлев В.И.
Члены жюри: канд. тех. наук, доцент Свито И.Л.
канд. тех. наук, доцент Петровский И.И.
Секретарь инж. Ермалович О.И.

АНАЛИЗ СОЕДИНИТЕЛЬНЫХ УТР-КАБЕЛЕЙ

Рассматривается соединение источника и приёмника как цепь с распределёнными параметрами

Целью исследования является анализ соединительных проводов, используемых для соединения компьютеров или, например, колонок с усилителями на основе цепи с распределёнными параметрами.

Такая цепь описывается телеграфными уравнениями:

$$\begin{cases} -\Delta u = (ri + L \frac{\partial i}{\partial t}) \Delta x \\ -\Delta i = (gu + C \frac{\partial u}{\partial t}) \Delta x \end{cases}$$

И графически представляется в виде:

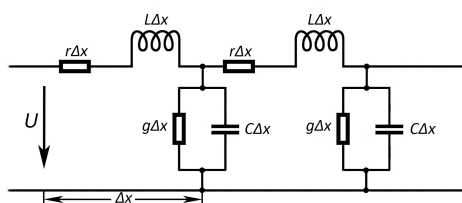


Рис. 1 – Элементарный участок цепи с равномерно распределёнными параметрами

Вышеизложенное – математическая модель цепи с распределёнными параметрами.

Чтобы поддерживать сигнал с минимальной потерей амплитуды, необходимо знать фазовую скорость, которая находится по формуле $\frac{dx}{dt} = v = \frac{\omega}{\alpha}$, где α – коэффициент фазы, мнимая часть коэффициента распространения $\gamma = \sqrt{(r + j\omega L)(g + j\omega C)} = \beta + j\alpha$. При этом она должна быть одинаковой на всех частотах.

Скорость распространения электромагнитной волны по кабельной линии $v = \sqrt{\frac{2\omega}{rC}}$ зависит от фазовой скорости, а фазовая скорость, в свою очередь, зависит от частоты. Как видно из формулы, чем выше диапазон частот, тем выше скорость распространения.

В случае, когда по линии идёт неискажённый сигнал, такую линию можно считать так называемой линией без искажений, где форма сигнала в начале и в конце линии одинакова. Чтобы последнее выполнялось, для устранения искажений, вызванных несогласованностью сопротивления приёмника с линией, то есть во избежание возникновения отражений на приёмном конце, сопротивление приёмника должно быть рассчитано по формуле $Z_C = \rho = \sqrt{\frac{r}{g}} = \sqrt{\frac{L}{C}}$.

Тогда фазовая скорость в линии без искажений $v = \frac{1}{\sqrt{LC}}$. Следует отметить, что условие $\frac{L}{r} = \frac{C}{g}$, как правило, не выполняется, так как обычно $\frac{L}{r} < \frac{C}{g}$. Поэтому приходится искусственно увеличивать индуктивность L .

Рассмотрим передачу сигнала без искажений на примере длинной биметаллической двухпроводной линии (с учётом того, что параметры распределены равномерно вдоль всей линии). В качестве такой линии можно использовать, например, усилитель с колонкой, связанные посредством УТР-кабеля (кабеля типа «витая пара»).

Выделяют, как правило, семь категорий кабелей типа «витая пара». По мере повышения категории увеличивается частота работы и, как следствие, скорость передачи данных.

Категории CAT1-4 признаны устаревшими, так как не поддерживают необходимую скорость передачи данных. Самой популярной считается категория 5е, которая может состоять из 2-х или 4-х пар проводников. Преимущество категории CAT5е над категориями 6 и 7, заключается в оптимальном соотношении цена / качество, так как в ней используется аналог линии без искажений (процент шумов составляет примерно 0,1%). Кабеля категории 6 и 7 требуют дополнительного экранирования.

Для обеспечения требуемой скорости передачи данных между источником и приёмником на расстоянии до 100 метров целесообразно использовать кабель CAT5 / 5е. Если в цепи фигурирует больше одного приёмника, то для поддержания достаточной скорости передачи необходимо включать в цепь разветвитель, если же требуется передать данные на большее расстояние, то необходим репитер.

Конечной целью является подбор кабеля с параметрами, при которых скорость распространения электромагнитной волны повышается до максимально возможного значения, где помехи минимизированы.

1. Атабеков Г. И. Теория линейных электрических цепей

Бондарчук Артем Дмитриевич, студент кафедры информационных технологий автоматизированных систем БГУИР, artembondarchuck@yandex.by.

Жуковский Дмитрий Сергеевич, студент кафедры информационных технологий автоматизированных систем БГУИР, agaton615@gmail.com.

Научный руководитель: Шилин Леонид Юрьевич, декан факультета информационных технологий и управления БГУИР, доктор технических наук, профессор, dekfitu@bsuir.by.

СИСТЕМЫ ВПРЫСКА ТОПЛИВА БЕНЗИНОВЫХ ДВИГАТЕЛЕЙ

Рассматриваются современные системы впрыска топлива бензиновых двигателей для выделения наиболее эффективной и оптимальной из них.

ВВЕДЕНИЕ

Система впрыска топлива применяется для дозированной подачи топлива в двигатель внутреннего сгорания в строго определенный момент времени. От характеристик данной системы зависит мощность, экономичность и экологический класс двигателя автомобиля. Системы впрыска могут иметь различную конструкцию и варианты исполнения, что характеризует их эффективность и сферу применения.

I. Моновпрыск, или центральный впрыск

Схема с центральным впрыском предусматривает наличие одной форсунки, которая расположена во впускном коллекторе. Она состоит из регулятора давления, форсунки впрыска, дроссельной заслонки, блока управления, датчика положения коленчатого вала двигателя, положения дроссельной заслонки, температуры и т.д. При запуске двигателя датчики считывают и передают информацию о состоянии системы в блок управления. Полученные данные сравниваются с эталонной характеристикой, и, на основе этой информации, блок управления рассчитывает момент и длительность открытия форсунки. На электромагнитную катушку направляется сигнал об открытии форсунки, что приводит к подаче топлива во впускной коллектор, где он смешивается с воздухом. Смесь топлива и воздуха подается в цилиндры. Преимуществами данной системы являются простота и надежность, а недостатками - повышенный расход топлива, низкие экологические показатели.

II. РАСПРЕДЕЛЕННЫЙ ВПРЫСК

Система с распределенным впрыском состоит из аналогичных элементов, но в такой конструкции предусмотрены отдельные форсунки для каждого цилиндра, которые могут открываться одновременно, попарно или по одной. Смешение воздуха и бензина происходит также во впускном коллекторе, но, в отличие от моновпрыска, подача топлива осуществляется только во впускные тракты соответствующих цилиндров. Принцип действия распределенного впрыска:

В двигатель подается воздух. При помощи ряда датчиков определяется объем воздуха, его температура, скорость вращения коленчатого вала, а также параметры положения дроссельной заслонки. На основе полученных данных электронный блок управления определяет объем топлива, оптимальный для поступившего количества воздуха. Подается сигнал, и соответствующие форсунки открываются на требуемый промежуток времени. Является самой распространенной системой впрыска бензиновых двигателей. Ее отличает умеренное потребление топлива, низкий уровень вредных выбросов, невысокие требования к качеству топлива.

III. НЕПОСРЕДСТВЕННЫЙ ВПРЫСК ТОПЛИВА

Система предусматривает подачу бензина отдельными форсунками напрямую в камеры сгорания каждого цилиндра под высоким давлением, куда одновременно подается воздух. Эта система впрыска обеспечивает наиболее точную концентрацию топливовоздушной смеси, независимо от режима работы мотора. При этом смесь сгорает практически полностью, благодаря чему уменьшается объем вредных выбросов в атмосферу. Такая система впрыска имеет сложную конструкцию и восприимчива к качеству топлива, что делает ее дорогостоящей в производстве и эксплуатации. Поскольку форсунки работают в более агрессивных условиях, для корректной работы такой системы необходимо обеспечение высокого давления топлива, которое должно быть не менее 5 МПа.

IV. Выводы

Система непосредственного впрыска топлива наиболее экономична и экологична из всех существующих систем впрыска топлива бензиновых двигателей. В отличие от остальных она позволяет точно контролировать порции топлива, подаваемые в камеры сгорания двигателя.

1. Ерохов, В. И. Системы впрыска бензиновых двигателей. Конструкция, расчет, диагностика / В. И. Ерохов // Горячая Линия - Телеком. – 2011. – С. 30-120.

Дегтярик Денис Васильевич, магистрант кафедры вычислительных методов и программирования БГУИР, ddegtyarik@gmail.com.

Научный руководитель: Кургулев Александр Петрович, профессор, кандидат технических наук, kaftoe@bsuir.by.

ОПТИМИЗАЦИЯ ЭНЕРГОПОТРЕБЛЕНИЯ МИКРОКОНТРОЛЛЕРОВ НА ОСНОВЕ RISC-V АРХИТЕКТУРЫ

Рассматриваются основные способы оптимизации энергопотребления микроконтроллеров на основе RISC-V архитектуры, их достоинства и недостатки

ВВЕДЕНИЕ

Сегодня при разработке системы на кристалле наиболее важной проблемой является увеличение потребляемой мощности.

RISC-V – это недавно представленная архитектура набора команд. Она бесплатна и изначально предназначена для поддержки исследований в области компьютерной архитектуры. В настоящее время является стандартом архитектуры в отрасли создания систем на кристалле[1]. Наиболее эффективными являются следующие методы управления питанием для оптимизации энергопотребления, применяющиеся к данной архитектуре:

- оптимизация напряжений коммутаций;
- ограничение часов;
- множественное напряжение питания;
- отключение энергии.

I. ОПТИМИЗАЦИЯ НАПРЯЖЕНИЙ КОММУТАЦИЙ

Элементы с различными пороговыми значениями напряжения коммутации используются в «методе оптимизации напряжений коммутаций» для оптимизации основных ограничений конструкции СБИС, а именно мощности, времени отклика и площади.

В то время как элементы с низковольтной коммутацией быстрее реагируют на входные сигналы, их потребляемая мощность выше, тогда как элементы с высоковольтной коммутацией переключаются с меньшей скоростью, но потребляемая мощность меньше.

II. ОГРАНИЧЕНИЕ ЧАСОВ

В типичном рабочем режиме часы непрерывно переключаются при каждом такте, без привязки к какой-либо функции или операции, связанной с операцией переключения. В случае, если данные загружаются в регистры нерегулярно, то заметное количество энергии теряется. Таким образом, ограничение часов гарантирует, что, когда блоки находятся в состоянии простоя,

мощность не теряется. Это ограничение тактовых импульсов может происходить либо в регистре, либо на более высоком уровне в тактовом дереве. Рассеиваемая мощность уменьшается за счет синхронизации по причинам, указанным ниже:

- в период простоя мощность не рассеивается, если функция стробирования отключает регистр;
- в самой схеме ограниченных часов энергия сохраняется.

III. МНОЖЕСТВЕННОЕ НАПРЯЖЕНИЕ ПИТАНИЯ

Исходя из требований к производительности, работа разных блоков при разных напряжениях является функцией «множественного напряжения питания». Данный метод – ключ к снижению энергопотребления, но для сигналов, где уровень напряжения меняется, требуются преобразователи уровня[2]. Таким образом, разработчики чипа используют данный метод на разных блоках в зависимости от требований к производительности.

IV. ОТКЛЮЧЕНИЕ ЭНЕРГИИ

Отключение энергии является одним из эффективных методов оптимизации энергопотребления, когда неиспользуемые элементы микросхемы отключаются. Неограниченное включение/выключение питания является источником ошибок, которые могут привести к деградации чипа. Таким образом, существует необходимость всесторонней проверки, гарантирующей, что микросхема работает так, как она должна для выключенных блоков, в то время как система может восстанавливаться после включения элементов.

Список литературы

1. Leena Singh, Leonard Drucker and Neyaz Khan, A SystemC Based Approach for Successful Tapeout, Kluwer Academic Publishers, Norwell, MA, 2004
2. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 6, Issue 3, Ver. II (May. -Jun. 2016)

Карташов Андрей Алексеевич, магистрант кафедры информационных технологий и программирования БГУИР, kartashow1998@gmail.com.

Научный руководитель: Журавлёв Вадим Игоревич, заведующий кафедрой теоретических основ электротехники БГУИР, кандидат технических наук, vadh@bsuir.by.

РАССМАТРИВАЕТСЯ АРХИТЕКТУРА И ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ СИСТЕМ НА КРИСТАЛЛЕ

Рассматривается архитектура и проблемы проектирования систем на кристалле.

ВВЕДЕНИЕ

Стиль проектирования систем на кристалле начал применяться на этапе освоения технологических процессов производства уровня 350-250 нм, когда стало возможным осуществлять интеграцию всех основных цифровых компонентов на одной кремниевой подложке. То, что раньше являлось набором нескольких чипов, теперь благодаря совершенствованию процессов производства стало одной интегральной СнК системой[1].

I. АРХИТЕКТУРА СнК

Система на кристалле (СнК) или однокристалльная система – электронная схема, выполняющая функции целого устройства и размещенная на одной интегральной схеме.

Типичная СнК содержит (см.рис.1.):

- Один или несколько микропроцессоров, микроконтроллеров.
- Блок памяти, состоящая из ОЗУ, ПЗУ, ППЗУ, кэш-память.
- Таймеры, счетчики, цепи задержки.
- Интерфейсы подключения внешних устройств: USB, FireWire, Ethernet, USART, SPI. Также могут поддерживаться протоколы беспроводной сети, такие как Wi-Fi, Bluetooth, 6LoWPAN.
- Блоки АЦП и ЦАП.

средств проектирования программного обеспечения СнК содержат компиляторы С/С++. Программное обеспечение современных СнК часто состоит из множества приложений – от систем жесткого реального времени, до систем с высокой пропускной способностью.

II. ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ СнК

СнК схемы обычно потребляют меньше энергии, имеют меньшую стоимость и большую надежность, чем много кристалльные системы, которые они заменяют. Однако общая стоимость одного большого кристалла выше, нежели у нескольких маленьких схем с той же функциональностью. С переходом к «узлам» менее 130 нм в конструировании интегральных схем возникли новые проблемы, связанные с тем, что традиционная конструкция МОП-транзистора перестает работать из-за различных паразитных эффектов, проявляющихся в малоразмерных конструкциях.

Самостоятельное проектирование разработчиком СнК всех сложно-функциональных блоков (СФ-блоков) – процессоры, таймеры, АЦП, различные интерфейсные блоки (UART, SPI, CAN, Ethernet и т.д), входящих в ее состав, не всегда целесообразно. Поэтому в данный момент широкое распространение получила практика разработки отдельных СФ-блоков для их последующего представления на рынок.

III. Выводы

С появлением систем на кристалле проектирование электронных устройств вышло на новый уровень. Как результат, современные программируемые электронные модули стали меньше, быстрее, мощнее и менее энергозатратными. Однако при проектировании СнК имеются трудности с обеспечением миниатюризации и высокой стоимостью разрабатываемого устройства. Дальнейшее развитие профильных САПР и методов проектирования таких систем поможет найти оптимальное решение.

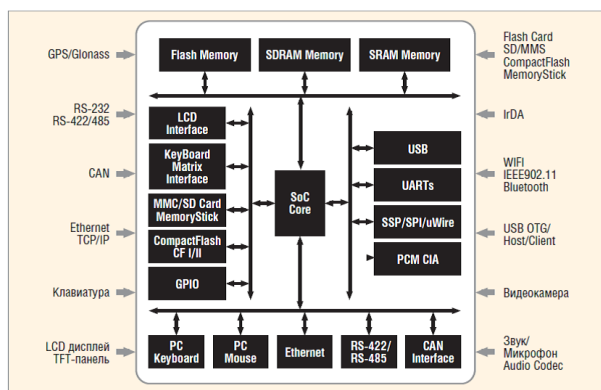


Рис. 1 – Архитектура СнК

Аппаратная часть всего проекта, так и его отдельных частей, описывается на языках Verilog, VHDL. Большинство современных

1. Немудров, В. Развитие СнК проектирования / В. Немудров, Г. Мартин // Системы на кристалле. – Москва, 2004. – Гл. 1. – С. 15–24.

Кенть Дмитрий Андреевич, студент кафедры теоретических основ электроники БГУИР, dima.kent.09@mail.ru.

Научный руководитель: Журавлев Вадим Игоревич, заведующий кафедры теоретических основ электроники БГУИР, кандидат технических наук, доцент, vadzh.@bsuir.by

СИСТЕМА ОПЛАТЫ ПРОЕЗДА В НАЗЕМНОМ ОБЩЕСТВЕННОМ ТРАНСПОРТЕ БАНКОВСКИМИ БЕСКОНТАКТНЫМИ КАРТАМИ

Рассматривается архитектура автоматизированной системы оплаты проезда в наземном общественном транспорте с помощью банковских карт и мобильных платежей.

ВВЕДЕНИЕ

Целью внедрения бесконтактных способов оплаты в транспорте является обеспечение большего удобства для клиентов, использующих безналичную систему оплаты с помощью банковских карт (без необходимости пополнять транспортные смарт-карты), а также сокращение эксплуатационных расходов при производстве и обслуживании транспортных карт.

I. РАЗРАБОТКА РЕШЕНИЯ ДЛЯ ВАЛИДАЦИИ БЕСКОНТАКТНОЙ ОПЛАТЫ ПРОЕЗДА

Внедренное решение должно быть гибким, чтобы его можно было адаптировать к различным применениям разработанной системы. Таким образом, система должна позволять пользователю выбирать тип используемого билета (электронные билеты, представленные оплатой с помощью банковских бесконтактных карт, мобильных телефонов с поддержкой NFC, а также бумажные билеты с QR-кодами) или позволить системе автоматически рассчитывать тариф, который будет взиматься с пользователя. На рисунке 1 представлена архитектура системы с дополнительным сервером

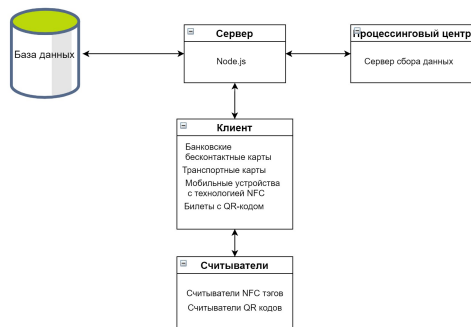


Рис. 1 – Архитектура системы

Разработанный прототип следует общей клиент-серверной архитектуре, в которой сервер обрабатывает запросы, поступающие от разных клиентов, представленных мобильным устройством или банковской картой. Сервер может быть реализован двумя способами: либо как

внутренняя служба серверов существующих операторов, либо как внешний сервер, который обращается к существующим серверам. Последний добавит еще один уровень в систему, что потребует от нового сервера отправки запросов на существующие серверы для выполнения любой необходимой выборки или обработки информации. В реализации прототипа используется внешняя база данных. Помимо этих компонентов, система содержит пассивные компоненты - считыватели кодов, которые размещаются внутри транспортных средств.

II. КЛЮЧЕВЫЕ ФУНКЦИИ СИСТЕМЫ

Основное внимание прототипа было сосредоточено на процессе проверки и возможности нескольких активных поездок. Одна важная концепция системы, которую следует объяснить, это концепция поездки. Каждый пользователь имеет свои активные и прошлые поездки, и каждая поездка может быть связана с определенным типом билета (например, транспортная смарт-карта с безлимитным проездным) или автоматическим билетом, который требует от пользователя проверки, чтобы система могла рассчитать, какую сумму нужно списать для этой поездки. Каждая поездка имеет связанные с ней контрольные точки, которые представляют собой сканирование кода пользователем при посадке в транспортное средство.

III. ВЫВОДЫ

Предлагаемая система является мультисервисной коммуникационной платформой, специально предназначенной для установки на борту транспортного средства. Она обеспечивает безопасную передачу данных между бортовыми терминалами и центральными устройствами, а также может быть адаптирована к внедрению новых технологий.

1. Мытник, К. Я. Смарт-карты и информационная безопасность / К. Я. Мытник, С. П. Панасенко // Информационные технологии. – 2018. – С. 264-270, 299.

Куль Виктория Александровна, студент кафедры теоретических основ электротехники БГУ-ИР, viktoriyakul7@gmail.com.

Научный руководитель: Курулев Александр Петрович, профессор, кандидат технических наук, alexparakuru@yahoo.com.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ УМНЫМ ДОМОМ НА ОСНОВЕ ПЛАТФОРМЫ ANDROID

Целью настоящей работы является разработка приложения для платформы домашней автоматизации Home Assistant, в котором будет возможность отправки MQTT сообщений.

I. ВВЕДЕНИЕ

Понятие «умный дом» говорит само за себя. Это сложная система, объединяющая в себе различные функции и программы. Она позволяет не только нажатием нескольких кнопок контролировать все процессы на расстоянии, но и полностью доверить управление дома системе.

II. ВЫБОР ПЛАТФОРМЫ ДЛЯ ДОМАШНЕЙ АВТОМАТИЗАЦИИ

Шлюз умного дома - это выделенный сервер, который выступает в качестве центрального узла связи между всеми устройствами.

В идеале платформа для домашней автоматизации должна соответствовать следующим критериям:

1. Иметь возможность определения текущего состояния устройства, а также управление данным состоянием в рамках локальной сети. Данная возможность позволяет не только значительно уменьшить задержку при коммуникации с девайсами. Но также обеспечивает приватность, т.к. данные не отправляются на сторонние сервисы.
2. Поддерживать как можно большее количество устройств от разных производителей
3. Поддерживать протокол передачи данных MQTT.
4. Поддерживать протоколы связи, такие как: Wifi, Bluetooth, Z-Wave, Zigbee.

На текущий момент времени, перечисленными выше требованиями в полной мере обладают 2 платформы: OpenHAB и Home Assistant. Было решено остановиться на выборе именно Home Assistant. Т.к. в сравнении с OpenHAB обладает следующими преимуществами:

1. Первоначальная настройка более простая.
2. Имеет функцию шаблонизации, которая позволяет применять функцию высшего порядка “map” ко всем входным и выходным данным.
3. Проще создать "кастомную" интеграцию для устройства. Сам проект обладает более простой архитектурой и написан на Python.

Островский Владислав Андреевич, студент ФИТиУ БГУИР, ostrovskiy565@gmail.com.

Научный руководитель: Нехаичик Елена Владимировна, старший преподаватель кафедры теоретических основ электротехники БГУИР, Nehajchik@bsuir.by.

III. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Несмотря на то, что официальное приложение под iOS для Home Assistant существует уже достаточно давно, версия под Android появилась относительно недавно. Релиз состоялся в ноябре 2019 года. И на данный момент имеет не так много функций.

Существуют приложения от сторонних разработчиков. Пример таких: Atriel, WallPanel, HomeHabit. Среди функций данных приложений можно отметить следующие:

1. Встроенный MQTT клиент, предназначенный для отправки значений с датчиков смартфона, а также текущего месторасположения
2. Постоянное соединение с Home Assistant, независимо от типа сети (локальная, внешняя)
3. Наличие пуш-уведомлений
4. Возможность использования смартфона в качестве камеры видеонаблюдения

IV. ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ УМНЫМ ДОМОМ

Несмотря на то, что перечисленные выше решения имеют встроенный MQTT клиент. Данный клиент предназначен только для отправки данных с датчиков смартфона, а также текущего месторасположения.

Ни одно приложение не имеет функции отправки MQTT сообщений с указанием конкретного топика и полезной нагрузки (англ. payload). Также нет и функции по удаленной конфигурации смартфона, например задание частоты отправки значений с датчиков смартфона.

В конечном итоге, создание прототипа в котором будут реализованы данные функции является целью данной работы

Список использованных источников:

1. <https://developers.home-assistant.io/>
2. <https://smarthome.university/>

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПОЖАРОТУШЕНИЯ СКЛАДА

Разработана автоматизированная система пожаротушения склада, предназначенная для ограничения распространения и уничтожения пожара, сохранения человеческих жизней и имущества.

ВВЕДЕНИЕ

В настоящее время проблема пожаров на различных объектах является особо острой. Это связано в том, что угрозы и риски, которые возникают в результате пожара, часто превосходят возможные последствия от других происшествий. Исходя из этого, созданию систем пожарной безопасности в современном обществе уделено огромное внимание.

I. ОПИСАНИЕ РАБОТЫ УСТРОЙСТВА

Автоматизированная система пожаротушения склада – это комплекс, объединяющий системы охранной и пожарной сигнализации, системы автоматического пожаротушения, контроля доступа и диспетчеризации объектов.

Основными элементами автоматизированной системы пожаротушения склада являются: интерфейс RS-485; устройство управления; датчики положения дверей и извещатели; считыватель (для включения или отключения автоматики); контрольно-пусковой блок; исполнительные устройства (световое и звуковое оповещение, а также газовые модули пожаротушения); пульт контроля и управления; запоминающее устройство; кнопка ручного пуска; система дымоудаления; дисплей.

Структурная схема данной системы представлена на рисунке 1.

Устройство управления (микроконтроллер) является центральным устройством и выполняет следующие функции: осуществляет контроль состояния извещателей, датчиков положения дверей, устройств дистанционного пуска через интерфейс RS-485, выхода огнетушащего вещества, исправности оборудования установки пожаротушения, передачу служебных и тревожных сообщений на пульт контроля и управления через интерфейс RS-485, дистанционный (поступающий из пульта контроля и управления) и ручной (с панели прибора) запуск и останов средств пожаротушения, автоматический запуск средств пожаротушения при срабатывании двух пожарных извещателей в одном либо в двух шлейфах сигнализации, управляет включением звукового и

светового пожарного оповещения, осуществляет контроль сетевого и резервного электропитания, отключение резервного питания при разряде аккумулятора.

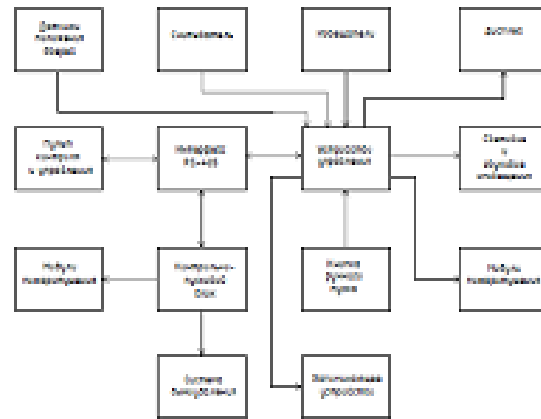


Рис. 1 – Структурная схема автоматизированной системы пожаротушения склада

Контрольно-пусковой блок управляет средствами пожаротушения, осуществляет контроль исправности цепей подключения исполнительных устройств, осуществляет передачу служебных и тревожных сообщений на пульт контроля и управления через интерфейс RS-485, управляет приточно-вытяжной вентиляцией, а также входным клапаном автоматической системы противопожарной вентиляции).

II. ЗАКЛЮЧЕНИЕ

Таким образом, автоматизированная система пожаротушения склада служит для быстрого реагирования на признаки возгорания и предотвращения пожара, а использование газовых модулей пожаротушения делает ее безопасной для человека и материальных ценностей.

Список литературы

1. Бабуров В.П., Бабурич В.В., Фомин В.И., Смирнов В.И. Производственная и пожарная автоматика. - 2007., М.
2. Бубыр Н.Ф., Бабуров В.П., Мангасаров В.И. «Пожарная автоматика» // Стройиздат. - 1984., М.

Полулех Алеся Владимировна, студент 4 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, poluleh.lesya@gmail.com.

Научный руководитель: Батюков Сергей Валентинович, старший преподаватель кафедры теоретических основ электротехники Белорусского государственного университета, магистр технических наук, batiukov@bsuir.by.

СИСТЕМА АНАЛИЗА ВЫБРОСОВ ОТРАБОТАВШИХ ГАЗОВ В АВТОМОБИЛЕ

Целью данной работы является изучение состава отработавших газов, рассмотрение структуры и принципа работы газоанализаторов, а также выявление наиболее важных факторов, определяющих качество газоанализаторов, формирующие требования к моделированию усовершенствованного устройства.

ВВЕДЕНИЕ

Диагностика, анализ и корректировка параметров автомобильных двигателей внутреннего сгорания (ДВС) - это одно из наиболее значимых направлений развития автомобильной промышленности, направленное на уменьшение вредного воздействия отработавших газов автомобиля, улучшения параметры расхода топлива, износостойкости составляющих двигателя и сроков его эксплуатации. Такие цели можно достичь при использовании специального оборудования по анализу и диагностике автомобильной техники, включающий анализатор отработавших газов автомобилей, который необходим для изучения состава газовой смеси и корректировки параметров.

I. АНАЛИЗ СОСТАВА ВЫХЛОПНЫХ ГАЗОВ ДИЗЕЛЬНОГО ДВИГАТЕЛЯ

Во время сгорания дизельного топлива зачастую образовывается большое количество различных веществ. Состав соединений зависит от самого двигателя и его устройства, его мощности, количества оборотов и других параметров. Однако, при полном сгорании дизельного топлива, существенно уменьшается концентрация вредных соединений. Обеспечить полное сгорание возможно при точном соблюдении соотношения состава топливно-воздушной смеси, высокой точности процесса впрыска топлива. Основным продуктом на выходе получается вода, кислород, оксид углерода 2 и другие соединения в малых количествах: оксид углерода (CO), несгоревшие углеводороды (HC, CH), окислы азота. Оксид углерода образуется при недостаточной точности распыления топлива. Кислород может в некоторых случаях не вступать в реакцию с топливом. Концентрация оксидов азота определяется, по большей мере, температурными показателями. При высокой температуре количество оксидов азота будет увеличиваться. Ещё одной причиной появления оксидов азота является также неточность при распылении топлива и слишком большое давление при сгорании. Некоторые приборы, такие как анализаторы газовых смесей, способны различать разные компоненты газов,

в том числе и автомобильных. С помощью результатов анализов выхлопных газов автомобиля можно контролировать и улучшать процессы и механизмы работы двигателей внутреннего сгорания.

II. ПРИНЦИП РАБОТЫ ГАЗОАНАЛИЗАТОРОВ

Спектрометрический блок анализатора автомобильных газов поглощает часть энергии светового потока, которая проходит через весь объем газа. Молекулы отработавших газов автомобиля представляют собой колебательную систему, которая способна поглощать излучение в инфракрасном спектре строго определенной длины волны. При пропускании стабильного потока инфракрасного диапазона через герметичную ёмкость, содержащую исследуемый газ часть энергии поглощается газом (эта часть называется часть спектра светового потока или абсорбционный максимум). Чем выше концентрация газа в герметичной ёмкости, тем больше поглощение энергии будет заметно.

Зная, что газы имеют разные абсорбционные максимумы, концентрацию газа в газовой смеси можно измерить с помощью поглощения соответствующей длины волны.

III. ВЫВОДЫ

Качество газоанализатора определяется точностью и надёжностью спектрометрического блока. Для увеличения точности показаний спектрометрического блока необходимо произвести очистку выхлопных газов следующим образом: с помощью фильтра на входе прибора произвести грубую очистку от сажи и крупных механических частиц, с помощью отделителя влаги удалить конденсат из накопителя, с помощью фильтра удалить мельчайшие механические частицы.

1. Павленко, В. А. Газоанализаторы / В. А. Павленко, В. В. Бражников // - 1965. - С. 296.
2. Берикашвили, В. Ш. Волоконно-оптические датчики газоанализаторы и системы контроля / В. Ш. Берикашвили, Д. И. Мирвицкий // Датчики и системы - 2000. - С. 45-50.

Савинков Артём Олегович, студент кафедры теоретических основ электротехники БГУИР, savinkov-artem@mail.ru.

Научный руководитель: Курулев Александр Петрович, профессор, кандидат технических наук, alexrapakuru@yahoo.com.

МИКРОПРОЦЕССОРНАЯ СИСТЕМА ОХРАННОЙ СИГНАЛИЗАЦИИ

Предлагается способ построения автономной системы охранной сигнализации для обеспечения безопасности жилых помещений с использованием микроконтроллера в качестве управляющего устройства.

ВВЕДЕНИЕ

В целях обеспечения безопасности постоянно разрабатываются и совершенствуются существующие методы и средства борьбы с преступлениями, связанными с хищением и порчей имущества. Наиболее надежной и экономически выгодной формой защиты является охрана объектов с помощью технических и программных средств, объединенных в систему охранной сигнализации.

I. СОСТАВ СИСТЕМЫ

Автономная система охранной сигнализации работает круглосуточно и обеспечивает контроль обстановки на объекте, в случае обнаружения несанкционированного проникновения на охраняемый объект формирует соответствующее оповещение. В общем случае охранная сигнализация состоит из приемных устройств, охранных извещателей, оповещателей, линий связи, источников и цепей питания.

II. СТРУКТУРНАЯ СХЕМА

Центральным элементом системы является контрольная панель. В ее состав входит блок управления и тампер (датчик вскрытия корпуса) для формирования тревожного сигнала при обнаружении попытки вскрытия панели. Питание блока управления осуществляется от сетевого напряжения через понижающий преобразователь, на случай отключения электроэнергии предусмотрена аккумуляторная батарея. Блок управления включает в себя микроконтроллер, радиомодуль для подключения беспроводных охранных датчиков и звуковых оповещателей, а также модуль GSM для информирования о срабатывании сигнализации путем отправления SMS-сообщения на телефон пользователя. Для контроля доступа на охраняемый объект используется кодовая панель, мгновенная активация системы оповещения осуществляется с помощью тревожной кнопки. В системе используются раз-

личные типы датчиков – открытия входной двери, движения, разбития стекла. Оповещение о несанкционированном проникновении на объект осуществляется с помощью звуковой сигнализации.



Рис. 1 – Структурная схема системы

Преимущества данной разработки, в сравнении с другими системами автономной охранной сигнализации, следующие: использование кодовой панели и персональных кодов доступа на объект снижает вероятность взлома; беспроводные датчики и оповещатели позволяют расширять площадь покрытия системы путем подключения дополнительных технических средств охраны, а также существенно снизить затраты на монтаж и установку; SMS-оповещение пользователя о проникновении или о попытке проникновения повышает уровень информативности системы, что не позволяет правонарушению остаться незамеченным.

1. Вальпа, О. Микроконтроллерная система безопасности с использованием GSM-канала / О. Вальпа // Современная электроника. – 2007. – №9. – С. 34-37.
2. Роскач, А. А. Охранные системы на основе геркона и GSM связи / А. А. Роскач // Материалы 71-й науч.-техн. конференции студентов и аспирантов «Актуальные проблемы энергетики»: сб. науч. тр. / Электронное издание, 2015. – С. 385-388.

Семченко Ольга Владимировна, студент 4-го курса факультета информационных технологий и управления БГУИР, olgadaineko4275396@gmail.com.

Научный руководитель: Ивануцкая Наталья Александровна, старший преподаватель кафедры теоретических основ электротехники БГУИР, nat.bguir@yandex.by.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ТРЁХЗОННОГО КЛИМАТ-КОНТРОЛЯ ГРУЗОВОГО ТЯГАЧА

Разработана структурная схема автоматизированной системы трёхзонного климат-контроля грузового тягача, предназначенная для повышения уровня комфорта водителей современных грузовиков.

ВВЕДЕНИЕ

Число грузовых перевозок растет с каждым годом. Грузовые автомобили преодолевают всё большие расстояния. Обеспечение соответствующего микроклимата в салоне грузового автомобиля становится все более важным аспектом комфорта и безопасности водителя. Разработка современной многозонной системы климат-контроля позволит вывести понятие комфорта в грузоперевозках на новый уровень.

I. ОПИСАНИЕ РАБОТЫ УСТРОЙСТВА

Автоматизированная система трёхзонного климат-контроля позволяет поддерживать установленную температуру в диапазоне от $+16^{\circ}\text{C}$ до $+30^{\circ}\text{C}$ в автоматическом режиме для трёх зон кабины: место водителя, место пассажира, спальный отсек. Такое разделение кабины на зоны позволяет добиться максимального комфорта для водителя и его сменщика.

Данная система разрабатывается для современных моделей тягачей, в которых кабина оборудована набором мультимедийных сенсорных панелей, являющихся частью системы современного интегрированного управления. Для подключения блока управления климат-контроля к бортовой системе грузовика используется интерфейс CAN.

Всё управление: включение, настройка и переключение режимов работы климат-контроля – производится посредством программного обеспечения бортовой системы управления. Для управления системой, настройки параметров и выбора режимов работы используется основная сенсорная панель на приборной панели и дополнительная, установленная в спальном отсеке.

II. СТРУКТУРНАЯ СХЕМА

Структурная схема системы климат-контроля представлена на рисунке 1. Климат-контроль состоит из основных элементов: блок кондиционера с отопителем в едином корпусе, блок управления на основе микроконтроллера,

сеть воздухопроводов и набор датчиков, измеряющих температуру, влажность воздуха, а также производящих контроль качества воздуха.

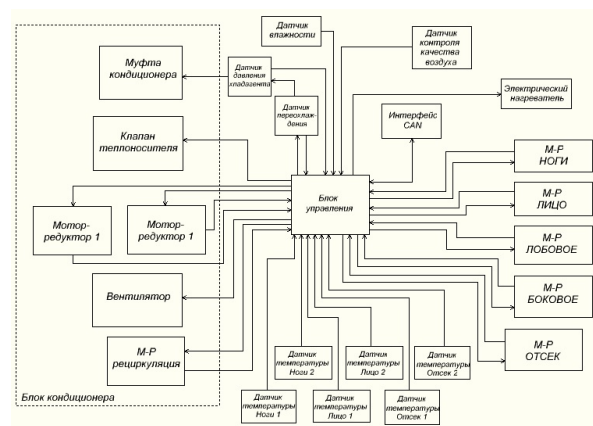


Рис. 1 – Структурная схема трёхзонного климат-контроля

Поддержание температуры на заданном уровне в зонах водителя и пассажира основано на использовании двух независимых температурных заслонок, управляемых мотор-редукторами 1 и 2. Для поддержания температуры в зоне спального отсека применяется дополнительный электрический нагреватель в воздуховоде с холодным воздухом.

III. ЗАКЛЮЧЕНИЕ

Увеличение числа зон, на которые делится замкнутый салон кабины, значительно увеличивает число датчиков необходимых для точного эффективного регулирования температуры в каждой отдельной зоне. Многозонная система климат-контроля позволит водителю и его сменщику чувствовать себя комфортно в любой части кабины.

1. Басыров Р.Р. Комфортабельность автомобилей: учебное пособие. / Басыров, А.Д. Галимянов, В.Н. Никишин. – Казань: Изд-во Казанского университета, 2018. – 104 с.
2. Хернер А, Риль Ханс-Юрген. Автомобильная электроника и электроника // За рулём – 2013.

Серый Александр Андреевич, студент 4 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, alex.wonder.ser@gmail.com.

Научный руководитель: Батюков Сергей Валентинович, старший преподаватель кафедры теоретических основ электротехники Белорусского государственного университета, магистр технических наук, batiukov@bsuir.by.

ЦИФРОВОЙ ТАХОМЕТР ДЛЯ ЛЕГКОВОГО АВТОМОБИЛЯ

Целью данной работы является изучение принципов работы цифровых тахометров, изучение электронных и функциональных схем, а так же выявление недостатков для последующей модернизации устройства.

ВВЕДЕНИЕ

Тахометр — мера скорости, измерительный прибор, предназначенный для измерения частоты вращения (количество оборотов в единицу времени) различных вращающихся деталей, таких как роторы, валы, диски и др., в различных агрегатах, машинах и механизмах. Обычно тахометры помимо собственно датчика скорости вращения включают в себя и показывающий прибор — индикатор, таким образом они состоят из двух частей, связанных электрической или иной связью.

I. ПРИНЦИП РАБОТЫ АВТОМОБИЛЬНОГО ТАХОМЕТРА

Тахометр используется для проверки производительности двигателя и помогает автомеханику понять его состояние для оптимизации функционирования с допустимыми параметрами. Принцип работы автомобильного электронного тахометра прост. Система зажигания запускает импульс напряжения электромеханической части тахометра, которая реагирует на среднее напряжение импульсов пропорционально частоте вращения двигателя. Сигнал передаётся двойным экранированным кабелем к индикатору.

Тахометры имеют температурную компенсацию для обработки измерений в диапазоне -20 до +70 С окружающей среды. Он позволяет водителю выбирать подходящие настройки дроссельной заслонки и шестерни во время движения, поскольку длительное использование на высоких скоростях вызывает недостаточную смазку, влияющую на двигатель, создаёт перегрев и приводит к ненужному износу трущихся деталей и к отказу машины.

II. ТИПЫ СОВРЕМЕННЫХ ТАХОМЕТРОВ

1. Аналоговые. Состоят из измерителя и интерфейса набора номера. Они не имеют возможности хранить базу данных, а также не вычисляют средние показания и их отклонения. Скорость движения преобразовывается в напряжение с использованием внешнего преобразователя частоты.

2. Цифровые — состоят из ЖК-дисплея или светодиодного индикатора и памяти для хранения информации. Они осуществляют статистические операции и подходят для точного измерения и мониторинга любых видов времени. Цифровые тахометры чаще встречаются в наши дни, они дают числовые показания вместо использования циферблатов.

3. Контактный тип, контактирует с вращающимся валом, прикрепляется к дизелю или электродвигателю. Например, оптический кодер или магнитный датчик измеряет обороты. Они способны измерять скорость вращения в пределах от 0, 5 об / мин до 10 тыс. об / мин, имеют ЖК-дисплей, работает с диапазоном рабочих температур от 0 до +40 С.

4. Бесконтактный тип не нуждается в физическом контакте с вращающимся элементом. В этом типе лазерный или оптический диск соединён с валом, результат считывается инфракрасным лучом или лазером. Этот тип замеряет скорость от 1 до 99,999 об/мин, угол обмера составляет меньше 120 градусов.

5. Временной, который вычисляет скорость по интервалу между входящими импульсами. Разрешение этого тахометра не ограничено, поэтому он более точен при измерениях низкой скорости.

6. Частотный, который вычисляет скорость по частоте импульсов. Этот тип работает с использованием красного светодиода, а оборот его зависит от вращающегося элемента. Он используется для высокоскоростных измерений.

III. ВЫВОДЫ

Тахометр — относительно простое и очень полезное устройство, которое серьезно облегчает пользование автомобилем. Цифровой тахометр намного нагляднее и удобнее в эксплуатации. К тому же выносные модели обычно имеют множество дополнительных функций.

1. Тараненко, В. В. Электронный тахометр / В. В. Тараненко, Э. П. Борноволоков // В помощь радиобителю. — 1984. — С. 79.

Страковский Егор Юрьевич, студент кафедры теоретических основ электротехники БГУИР, egorstrakovskij@gmail.com.

Научный руководитель: Курулев Александр Петрович, профессор, кандидат технических наук, alexparakuru@yahoo.com.

СТАБИЛИЗАТОР ТОКА ВОЛЬТМЕТРА УНИВЕРСАЛЬНОГО КЛАССА ТОЧНОСТИ 0,01

Рассматривается реализация генератора стабильных токов (ГСТ) для измерения сопротивления в составе универсального вольтметра. Предлагается оснастить генератор схемой защиты от перегрузок и использовать усилитель слежения за потенциалом для увеличения точности.

ВВЕДЕНИЕ

Для измерения сопротивления необходимо чтобы в состав измерительного прибора входил генератор стабильных токов, который, в зависимости от выбранного предела измерений, генерирует ток определенного значения. Ток пропускается через подключенную нагрузку, в это время на ней измеряется напряжение и путем деления напряжения на силу тока вычисляется сопротивление (см. рис. 1) [1].

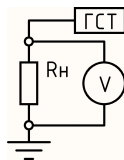


Рис. 1 – Схема измерения сопротивления.

I. ГЕНЕРАЦИЯ СТАБИЛЬНЫХ ТОКОВ

Плата стабилизатора тока соединяется непосредственно с блоком аналого-цифрового преобразования (АЦП) вольтметра. Питание доставляется с этого же блока, однако в устройстве предусмотрен преобразователь постоянного напряжения для получения напряжения +3 В и -3 В (см. рис. 2).

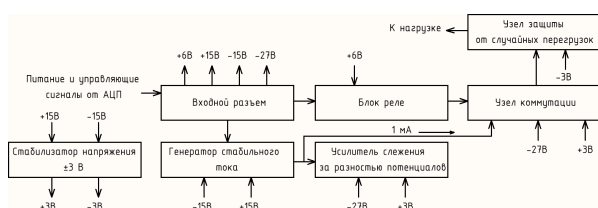


Рис. 2 – Структурная схема стабилизатора тока.

Связь осуществляется через входной разъем. Для выбора предела измерения сопротивления применяются герконовые реле, сигналы управления которыми приходят с блока АЦП.

ГСТ на операционном усилителе (ОУ) КД140УД1408А обеспечивает стабильный ток силой 1 мА, необходимый для измерения сопротивления нагрузки. Этот ток может быть изме-

нен в узле коммутации, путем подключения последовательно соединенных резисторов с помощью герконовых реле, для обеспечения возможности изменять пределы измерения сопротивления. Ток силой 1 мкА соответствует сопротивлению 200 Ом и 2 кОм, 0,01 мА – 20 кОм и 200 кОм, 1 мА – 2 МОм. [2].

II. УВЕЛИЧЕНИЕ ТОЧНОСТИ И ЗАЩИТА ОТ ПЕРЕГРУЗОК

Для того чтобы значение тока на выходе ГСТ не зависело от внешних воздействий, проводник, по которому он протекает, на печатной плате окружается тонким экраном, который соединяется с общим проводником. Нулевую разность потенциалов выхода ГСТ и экрана обеспечивает усилитель слежения на ОУ КД140УД1408А [3].

Также для защиты от случайных перегрузок, например, вследствие подачи напряжения на вход вольтметра в режиме измерения сопротивления реализована схема защиты, которая выдерживает напряжение до 100 В.

III. ВЫВОДЫ

Разработанный стабилизатор обеспечивает точность задания измерительного тока 0,01 %. Важным моментом является использование прецизионных элементов в цепях задания стабильного тока, чтобы обеспечить максимальную точность измерения сопротивления. Преимущества разработанного устройства – высокая точность, защита от перегрузок и использование более чувствительных и надежных герконовых реле вместо электромагнитных.

1. Электротехника [Электронный ресурс] – Режим доступа: <https://electrono.ru/elektroizmeritelnye-pribory-i-metody-izmerenij/103-izmerenie-elektricheskogo-soprotivleniya>
2. Э.Г. Попов. Основы аналоговой техники. Учеб. пособие для студ. радиотехнических спец. – Мн.: БГУИР, 2006 - 276 с.
3. A. J. Peyton, V. Walsh. Analog electronics with Op Amps: a source book of practical circuits. – Cambridge University Press, 1993.

Шаблинский Илья Вячеславович, студент 4-го курса факультета информационных технологий и управления БГУИР, ilyia9963@gmail.com.

Научный руководитель: Иванецкая Наталья Александровна, старший преподаватель кафедры теоретических основ электротехники БГУИР, nat.bguir@yandex.by.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА КОНТРОЛЯ МИКРОКЛИМАТА В ТЕПЛИЧНОМ КОМБИНАТЕ

Разрабатывается структурная схема хорошо масштабируемой автоматизированной системы контроля микроклимата в тепличном комбинате, состоящим из трёх теплиц

ВВЕДЕНИЕ

В современном мире очень важны инновационные автоматизированные системы контроля микроклимата в сельском хозяйстве. Обусловлено это необходимостью роста производительности. Но существующие системы микроклимата разбиты на несвязанные системы. Только для крупных предприятий создаются уникальные дорогостоящие комплексные системы. Поэтому была поставлена задача разработки автоматизированной системы контроля микроклимата, которая будет подходить для любого предприятия и обладать свойством масштабируемости.

1. СИСТЕМА КОНТРОЛЯ МИКРОКЛИМАТА ТЕПЛИЧНОГО КОМБИНАТА

Рассмотрев все преимущества и недостатки существующих решений, были выбраны следующие подсистемы: воздушного отопления, полива и досвечивания. Данные подсистемы контролируют влажность почвы и воздуха, температуру и освещённость внутри теплицы.

Система воздушного отопления была выбрана для проветривания и одновременного поддержания необходимых показателей температуры и влажности воздуха внутри теплиц. Данная система будет использовать наружный воздух, нагревая его и повышая влажность до необходимых значений с помощью калорифера водяного со встроенным мелкодисперсным распылителем. Управление температурой теплоносителя калорифера осуществляется с помощью клапанов подачи горячей и холодной воды. Система обеспечивает нагрев или охлаждение поступающего воздуха.

Система полива предназначена для поддержания необходимой влажности почвы. Данная система осуществляет измерение текущей влажности почвы с помощью регистрирующего устройства (датчика) и на основании этого значения принимает решение о поливе. В качестве устройства полива используется компенсированный капельный шланг. Это гарантирует подачу воды только в прикорневую зону растения. Основное преимущество компенсационного шланга заключается в том, что при его использовании

расход воды из встроенных капельниц практически не зависит от протяжённости линии и уклона укладки.

Система досвечивания растений предназначена для поддержания необходимого уровня освещённости в теплицах для выращивания сельхозкультур с учётом уровня внешней солнечной радиации и времени суток. Система состоит из датчика освещённости и устройства досвечивания, которое представляет собой натриевые лампы высокого давления, поскольку их свет наиболее близок для растений. Натриевые лампы являются наиболее экономически выгодными, а также наиболее эффективным источником света.

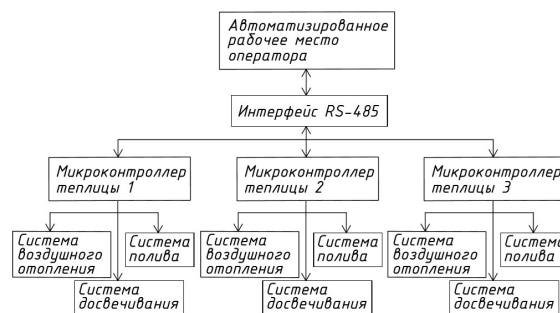


Рис. 1 – Структурная схема автоматизированной системы контроля микроклимата в тепличном комбинате

Структурная схема контроля микроклимата тепличного комбината содержит три микроконтроллера высшего уровня, предназначенных для контроля подсистемами каждой из теплиц. Все микроконтроллеры связаны с автоматизированным рабочим местом оператора с помощью интерфейса *RS-485*.

Благодаря такому построению системы, возможно увеличение количества теплиц в тепличном комбинате без добавления дополнительных систем. Интерфейс *RS-485* позволяет увеличить количество подсоединяемых микроконтроллеров, соответственно теплиц, до 32-х.

1. Тигранян Р.Э. Микроклимат. Электронные системы обеспечения - Москва: РадиоСофт, 2005. — 112с.

Шайпак Анастасия Александровна, студентка 4 курса факультета информационных технологий и управления БГУИР, nastassia.shaypak@mail.ru.

Научный руководитель: Батюков Сергей Валентинович, старший преподаватель кафедры теоретических основ электротехники БГУИР, магистр технических наук, batiukov@bsuir.by.

Секция "Гуманитарные дисциплины"

Подсекция «История Беларуси»

Председатель: канд. ист. наук, доцент, зав. каф. ГД Николаева Л.В.
Члены жюри: канд. ист. наук, доцент Литвиновская Ю.И.
канд. ист. наук, доцент Куракевич Н.И.
Секретарь преподаватель Мякинская А.В.

Подсекция «Великая Отечественная война советского народа»

Председатель: канд. ист. наук, доцент, зав. каф. ГД Николаева Л.В.
Члены жюри: канд. ист. наук, доцент Литвиновская Ю.И.
канд. ист. наук, доцент Сугако Н.А.
Секретарь преподаватель Мякинская А.В.

Подсекция «Политология»

Председатель: канд. ист. наук, доцент Вашкевич И.В.
Члены жюри: преподаватель Галицкая Е.М.
канд. ист. наук, доцент Киселев А.А.
преподаватель Борисов Е.А.
Секретарь преподаватель Янковский Ю.Ю.

Подсекция «Социология»

Председатель: канд. ист. наук, доцент Качалов И.Л.
Члены жюри: преподаватель Галицкая Е.М.
канд. ист. наук, доцент Сугако Н.А.
Секретарь преподаватель Пащева А.Г.

ЗИМНЯЯ ВОЙНА: НОВЫЕ ПОДХОДЫ

В статье рассматриваются основные причины, ход, этапы и итоги советско-финляндской войны.

После подписания советско-германского договора о ненападении и секретного протокола к нему, в советскую сферу влияния вошла Финляндия. Было принято решение отодвинуть границу от Ленинграда (30 км от финской границы). Мирные пути решения (земли Карелии взамен на передвижения границы на Карельском перешейке) отвергались финнами. Поводом войны является инцидент у села Майнила 25 ноября 1939 года, находившиеся на Карельском перешейке в 800 метрах от финской границы. Майнила подверглась артиллерийскому обстрелу, в результате которого погибло 4 мирных жителя и ранено 8 бойцов Красной Армии. Руководство СССР возложило вину на финскую сторону. Финское правительство ответило следующим: обстрел действительно был, но велся с советской стороны. Но, что случилось на самом деле, до сих пор не известно. Существуют 2 версии (советская и финская):

- 1) Финляндия обстреляла территорию СССР;
- 2) Это была провокация, подготовленная НКВД. Зачем эта провокация нужна была Финляндии? Историки говорят о двух причинах:

1. Финны были инструментом политики в руках англичан, которым была нужна война;
2. Они хотели расширить свои территории и влияние.

Советско-финляндская война может быть разделена на два этапа:

- 1) декабрь 1939 г. – 7 января 1940 г.;
 - 2) 7 января 1940 г. – 12 марта 1940 г.
- 30 ноября 1939 г. Красная армия пересекла государственную границу Финляндии. Советское руководство полагало, что война будет завершена в несколько недель (3 недели – крайний срок), т.к. не рассматривало Финляндию как противника, рассчитывая уничтожить её в кратчайшие сроки. Если говорить более конкретно, то план советского командования был примерно следующим:
- 1). Вести войска;
 - 2). Создание рабочего подконтрольного СССР правительства;
 - 3). Молниеносное наступление (выход к Хельсинки за 1,5-2 недели);
 - 4). Склонение реального правительства Финляндии к миру и полной капитуляции в пользу просоветского правительства.
- Первые два пункта удалось реализовать в первые дни войны, но дальше начались проблемы.

Блицкриг не получился, и армия застряла в финской обороне. Хотя в начальные дни войны казалось, что все идет по плану – советские войска продвигались вперед. Но очень скоро (4 декабря) они наткнулись на линию Маннергейма. Она насчитывала 101 бетонное укрепление, 5800 бетонных сооружений. Справедливости ради следует отметить тяжелые условия местности линии Маннергейма (болота и озера). Войска продвигались вглубь страны медленно. С каждым днем темпы продвижения только уменьшались. 7 января 1940 г. штаб Северо-западного фронта возглавил маршал С.К. Тимошенко. Ему предстояло решить проблемы и организовать успешное наступление РККА. В этот момент советско-финская война взяла передышку. До февраля активных действий не велось. С 1 по 9 февраля начались мощные удары по линии Маннергейма. Предполагалось, что 7-я и 13-я армии должны были решительными фланговыми ударами прорывать линию обороны и занять участок Вуоксы-Кархул. После этого планировалось движение на Выборг и перекрытие железнодорожных и шоссежных дорог, ведущих на Запад. 11 февраля 1940 г. на Карельском перешейке началось генеральное наступление советских войск. Это был переломный момент Зимней войны, поскольку частям РККА удалось прорвать линию Маннергейма и начать продвижение вглубь страны. Продвигались медленно из-за специфики местности, сопротивления финской армии и сильных морозов. В начале марта советская армия была уже на западном побережье Выборгского залива. На этом фактически война закончилась, поскольку было очевидно, что у Финляндии нет большого сил и средств для сдерживания красной армии. С этого времени начались переговоры о мире, в которых свои условия диктовал СССР, а В.Ф. Молотов постоянно подчеркивал, что условия будут жесткими, ведь финны вынудили начать войну, в ходе которой пролилась кровь советских солдат. СССР 28 февраля 1940 г. передало Финляндии свои условия для заключения мира:

1. СССР получал Карельский перешеек вместе с Выборгом;
2. Западное и Северное побережье Ладожского озера, вместе с городами Кексгольм, Суоярви и Сортавала;
3. Острова в Финском заливе;
4. Остров Ханко с морской территорией и базой сдавался в аренду СССР на 50 лет;
5. Договор между Финляндией и СССР от 1920 г. утратил свою силу;

6. С 13 марта 1940 года прекращаются боевые действия.

Почему война тянулась так долго а для подавления финнов собирали дивизии по всей стране? Причин тому несколько: плохая организация войск (это касается плохой работы командного состава, но большая проблема – слаженность между родами войск); плохая обеспеченность (армия терпела нужду практически во всем. Война велась зимой и на севере, где температура воздуха к концу декабря опустилась ниже -30. И при этом армия не была обеспечена зимней одеждой); недооценка противника (планировалось быстро подавить финнов и решить проблему без войны); поддержка Финляндии другими странами (Англия, Италия, Венгрии, Швеция оказывали помощь Финляндии во всем: оружие, припасы, продовольствие, самолеты и так далее); болотистая и лесистая местность, отсутствие дорог.

Сталин очень нервничал из-за того, что война затягивалась. Он повторял – Весь мир смотрит за нами. И он был прав. Поэтому Сталин требовал решения всех проблем, наведения порядка в армии и скорейшего решения конфликта.

Итак, негативные моменты данной войны: кошмар первых месяцев войны и огромное количество жертв. По большому счету именно декабрь 1939 г. и начало января 1940 г. продемонстрировали всему миру, что Красная армия слаба. Но в этом были и позитивный момент: советское руководство увидело реальную силу своей армии. Перед войной с Финляндией руководство СССР витало в облаках, полагая, что имеет сильную армию. Но декабрь 1939 года показал, что это не так. Армия была крайне слаба. Но, начиная с января 1940 года, вносились изменения, которые изменили ход войны, и которые во многом подготовили боеспособную армию для Великой Отечественной войны. Доказать это очень легко. Практически весь декабрь 1939 г. РККА штурмовала линию Маннергейма – результата не было. 11 февраля 1940 г. линия Маннергейма была прорвана за один день. Этот прорыв был возможен, поскольку его осуществляла уже другая армия, более дисциплинированная, организованная, обученная. И против такой армии у финнов не было ни единого шанса, поэтому Маннергейм, занимавший пост министра обороны, уже тогда стал говорить о необходимости мира.

Автухович А.А., студент факультета информационных технологий и управления БГУИР.
Научный руководитель: Мякинська Анна Владимировна, преподаватель кафедры гуманитарных дисциплин БГУИР mav@bsuir.by

ОРШАНСКАЯ БИТВА 1514 Г. КАК ПРИМЕР ВОЕННОГО ИСКУССТВА ВКЛ

В докладе представлен новый взгляд на крупнейшее сражение в ходе русско-литовской войны 1512 – 1522 гг. – битву под Оршей.

Война 1512–1522 гг. была самой продолжительной в серии русско-литовских войн конца XV – первой половины XVI вв. Битва под Оршей стала одним из эпизодов этой войны. Ей предшествовал захват Смоленска войсками Московского великого княжества 31 июля 1514 г., ключевого пункт оборонительной линии Великого Княжества Литовского (далее – ВКЛ) на восточной границе. Русские войска устремились вглубь территории ВКЛ, последовательно заняв города Кричев, Дубровну и Мстиславль, которые не оказали сопротивления. По московским источникам, численность армии составляла около 13 000 чел.

Тем временем армия короля польского и великого князя литовского Сигизмунда I выдвинулась в район Борисова, откуда продолжила путь в направлении Орши. Командование осуществляли гетман ВКЛ Константин Острожский. Основная часть войска короля состояла из поляков и наемников-европейцев, которых насчитывалось до 2/3 всего войска (до 9000 человек). Собственно войско ВКЛ занимало на поле боя только правое крыло. Общая численность армии Сигизмунда равнялась примерно 15 000 человек.

Генеральное сражение между русскими и войсками ВКЛ произошло 8 сентября 1514 г. Сражение началось с перестрелки – полевые пушки ВКЛ сделали залп, в ответ «москвиты» выпустили стрелы. Сначала их правое крыло стремительно атаковало. Но остальные части бездействовали. Наступление левого крыла на позиции посполитого рушения развивалось поначалу удачно. Но, увлекшись атакой, русские открыли свой тыл артиллерийской засаде. В панике отряды обратились в бегство.

Тем временем удар передового полка русских увяз в пехотном строю центра войск противника, пикинеры сдержали конницу, а наемники с огнестрельным оружием нанесли ей большие потери. После продолжительной схватки гетман К. Острожский бросил в бой тяжелую конницу, которая врезалась в большой полк русских, где уже царил неразбериха из-за оголенных флангов. Войско ВКЛ прижало русских к реке Крапивне. Основные потери русская армия понесла не в ходе сражения, а при беспорядочном отступлении. Из 11 больших московских воевод в плен попало 6, двое было убито и только три спаслись. Победа под Оршей была использована официальной пропагандой ВКЛ для создания неблагоприятного имиджа своего противника. Европа узнала о грандиозном сражении, в ходе которого из 80 000 врагов 30 000 было убито, 8 главных воевод, 37 князей, баронов и знатных дворян, помимо этого 1500 воинов попало в неволю. Данные о численности войск вызывают сомнения у некоторых российских исследователей. Несомненно, битва 8 сентября 1514 г. являлась крупным сражением. Однако она не переломила ход кампании. В конечном итоге войны 1512–1522 гг. ВКЛ потеряло Смоленскую землю.

Список литературы

1. Лобин, А.Н. Взятие Смоленска и битва под Оршей 1514 г./ А.Н. Лобин [Электронный ресурс]. – Режим доступа: <https://www.rulit.me/books/vzyatie-smolenska-i-bitva-pod-orshej-1514-g-read-516754-28.html>. – Дата доступа: 20.03.2020.
2. Лобин, А.Н. Битва под Оршей 8 сентября 1514 года. К 500 –летию сражения/ А.Н. Лобин [Электронный ресурс]. – Режим доступа: <https://nemaloknig.com/read-373654/?page=1booktxt>. – Дата доступа: 20.03.2020.

Акулов Никита Игоревич, студент 1 курса факультета информационных технологий и управления БГУИР, nikita.akulov.1@mail.ru
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by

СОЦИАЛЬНО-ЭКОНОМИЧЕСКОЕ ПОЛОЖЕНИЕ ЗАПАДНОЙ БЕЛАРУСИ В СОСТАВЕ ПОЛЬШИ

Доклад посвящен характеристике социально-экономического положения Западной Беларуси в составе Польши.

В соответствии с Рижским мирным договором от 18 марта 1921 г. к Польше отошла территория Западной Беларуси площадью 112,9 тыс. км² с населением в 1931 г. 4,6 млн. человек. Около 70% его составляли белорусы. Польскими властями эта территория была разделена на четыре воеводства: Полесское, Новогрудское, Виленское и Белостокское. Историю западнобелорусских земель во время польской аннексии следует рассматривать, обращаясь к нескольким важным вопросам. Вопрос первый – экономический. Западная Беларусь являлась отсталой окраиной Польши. Составляя 24% территории и 13% населения, удельный вес промышленности Западной Беларуси едва превышал 3%. Большинство действующих предприятий были мелкими (до 20 рабочих) и занимались переработкой продукции сельского хозяйства и некоторых видов местного сырья. Некоторое развитие получила деревообрабатывающая промышленность. С 1921 г. по 1936 гг. было высечено 400 тыс. га лесов. Более 70% древесины в необработанном виде вывозилось за границу. В промышленности, на транспорте было занято около 128 тыс. человек. Рабочий день продолжался 10-12 часов. Заработная плата рабочих была в 2 раза ниже, чем в этнической Польше. Высоким был уровень безработицы. Пособие выдавалось не всем безработным. Вопрос второй – аграрный. Земельные отношения в Западной Беларуси характеризовались господством крупного помещичьего землевладения. Помещичьи и магнатские имения, которые в начале 1920-х гг. не насчитывали и 1% от всех хозяйств, имели 40,5% всех земельных угодий. В среднем на помещичье хозяйство приходилось 500 га, а на крестьянское хозяйство 7 га земли. От 2 до 5 га земли владели 34,2% крестьянских хозяйств. В целях снижения со-

циальной напряженности власть пошла на аграрные реформы. В июле 1925 г. сейм Польши принял «Закон об осуществлении земельной реформы». Составной ее частью являлось проведение так называемой парцелляции, т.е. продажа мелкими участками (парцеллами) части помещичьей и государственной земли, а также комасации (хуторизация) и ликвидации сервитутов. Власти использовали парцелляцию для насаждения польских военных колонистов – осадников, которые стали их военно-политической опорой. С 1921 г. по 1930 г. в трех западнобелорусских воеводствах поселилось 4 436 военных осадников, которые получили бесплатно или по низкой цене земельные наделы до 45 га. С 1921 по 1925 гг. в трех западнобелорусских воеводствах под видом парцелляции помещики продали около 450 тыс. га земли. Основными покупателями земли были торговцы, предприниматели, чиновники, осадники. С 1923 г. по 1938 г. было переселено на хутора 259,3 тыс. крестьянских хозяйств, что составляло 43% хозяйств и 50% крестьянской земли. Почти 14% урожая крестьяне вынуждены были продавать, чтобы рассчитаться с налогами. Недоимки по налогам в 1934 г. составили 80 млн. злотых. Таким образом, социально-экономическое положение земель Западной Беларусь в составе Польши являлось весьма сложным. Польские власти в основном занимались их экономической эксплуатацией и не заботились о развитии экономики региона.

Список литературы

1. Социально –экономическое положение на территории Западной Беларуси в 20 –е – 30 –е годы [Электронный ресурс]. – Режим доступа: <https://library.by/portalus/modules/belarus> – Дата доступа: 10. 12. 2019.

Баранов Дмитрий Максимович, студент 1 курса факультета информационных технологий и управления БГУИР, dmitrybar13@gmail.com
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by

ВЛИЯНИЕ АВАРИИ НА ЧЕРНОБОЛЬСКОЙ АЭС НА СОЦИАЛЬНО-ЭКОНОМИЧЕСКУЮ СИТУАЦИЮ В БЕЛАРУСИ (1986 – 1990-Е ГГ.)

Доклад посвящен проблеме влияния аварии на ЧАЭС на социально-экономическое положение Беларуси в 1986 – 1990-х гг.

26 апреля 1986 г. на Чернобыльской АЭС произошла авария. Она стала результатом неудачных испытаний реактора 4-го энергоблока, вызвавших взрыв. Ранним утром 27 апреля началась эвакуация сначала 10-километровой, а потом и 30-километровой зоны. Ликвидацией последствий аварии на ЧАЭС занимались люди со всего СССР. Основная часть работ была выполнена в 1986-1987 гг. Общее число ликвидаторов составило 600 тысяч человек. Для ограничения поступления радиоактивных веществ в окружающую среду к ноябрю 1986 г. вокруг 4-го энергоблока был построен саркофаг – «Укрытие». Позже был возведен второй изолирующий объект «Укрытие-2». Он представлял собой стальную конструкцию высотой 108 м и длиной 150 м, под которой поместился старый саркофаг и вспомогательные сооружения. В ноябре 2016 г. объект «Укрытие-2» или НБК был перемещен из зоны монтажа и занял проектное положение над «саркофагом». Летом 2019 г. новый безопасный конфайнмент был сдан в эксплуатацию. Однако последствия Чернобыльской катастрофы дали о себе знать на территории Беларуси. 23% нашей страны были загрязнены радиоактивными изотопами. Для сравнения, территория Украины пострадала на 4,8% от общей площади, а территория России – около 0,5%. Каждый пятый житель республики пострадал от радиоактивного загрязнения, а из хозяйственного оборота было выведено 20% земель. Учитывая масштабность и тяжесть последствий катастрофы на ЧАЭС, в июле 1990 г. Верховный Совет Беларуси объявил территорию республики зоной экологического бедствия.

Основной проблемой для Беларуси являются изотопы цезия-137 и трансурановые элементы. Кроме проблем загрязнения окружающей среды и роста уровня заболеваемости раком щитовидной железы, онкологических заболеваний и генетических отклонений, авария на Чернобыльской АЭС также повлекла за собой огромные затраты. Нужно было построить новое жилье для переселенцев и обеспечить им оказание медицинской помощи. Из сельскохозяйственного оборота было выведено 2,64 тыс. км² сельхозугодий. Ликвидировано 52 колхоза и совхоза, закрыты 9 заводов. Значительно уменьшены размеры пользования лесными, минерально-сырьевыми и другими ресурсами. В зоне загрязнения оказались 132 месторождения различных видов минерально-сырьевых ресурсов. Также большой урон нанесен лесному хозяйству. Ущерб, нанесенный республике чернобыльской катастрофой в расчете на 30-летний период ее преодоления, оценивается в 235 млрд. долларов США, что равно 32 бюджетам республики 1985 г. Безусловно, авария на ЧАЭС затронула все сферы жизни общества и повлекла за собой огромные потери не только бюджетные, но и человеческие. Задача нашего поколения – не допустить повторения таких катастроф в будущем.

Список литературы

1. Авария на Чернобыльской АЭС [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Авария_на_Чернобыльской-АЭС. – Дата доступа: 24.02.2020.
2. Новый безопасный конфайнмент АЭС [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Новый_безопасный_конфайнмент-АЭС. – Дата доступа: 24.02.2020.

*Борисевич Полина Владимировна, студентка 1 курса факультета информационных технологий и управления БГУИР, polina.borisevich.86@gmail.com.
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by*

КОЛЛАБОРАЦИОНИЗМ НА ТЕРРИТОРИИ БЕЛАРУСИ В ГОДЫ ВЕЛИКОЙ ОТЕЧЕСТВЕННОЙ ВОЙНЫ

Доклад посвящен характеристике деятельности ряда белорусских коллаборационистских организаций.

Активное участие в осуществлении планов нацистов принимали белорусские коллаборационисты, которые с разрешения нацистов и под их руководством создали ряд организаций. Дадим их краткую характеристику. Белорусская народная самопомощь (БНС). Создана 22 октября 1941 г. Руководители – И. Ермаченко, В. Ивановский, Ю. Соболевский. Руководящим органом была Центральная рада (Централь), состоявшая из 12 членов во главе с И. Ермаченко [1]. Занималась сбором средств и оказанием материальной помощи пострадавшим от войны немцам. В июне 1943 г. она была реорганизована в Белорусскую самопомощь (БСП) – организацию, занимавшуюся исключительно благотворительностью. С 1 марта 1944 г. БСП была подчинена Белорусской центральной раде. Белорусский корпус самообороны (БКС) – военизированное формирование. Создан в июне 1942 г. Белорусской народной самопомощью. Главный комендант – И. Ермаченко. Основной задачей была помощь немецкой и местной полиции в борьбе с партизанами. Было создано 20 батальонов, которые немцы так и не решились вооружить. Весной 1943 г. БКС был распущен. В июне 1943 г. в Минске была сформирована Беларуская рада даверу (БРД) – совещательный орган при генеральном комиссаре В. Кубе. Председателем стал В. Ивановский. В декабре 1943 г. БРД была преобразована в Белорусскую центральную раду. Белорусская краевая оборона (БКО) – военизированное формирование. Создана в начале марта 1944 г. во главе с Ф. Кушелем. Главной задачей БКО была борьба против партизан, Красной Армии и польской Армии Крайовой. Основу БКО составляло принудительно мобилизованное мужское население 1908-1924

г.р. Было мобилизовано около 25 тысяч человек, из которых к середине апреля 1944 г. создано 36 пехотных и 6 саперных батальонов. Пособниками нацистов были три молодёжные структуры: Союз белорусской молодежи (СБМ), Союз борьбы против большевизма (СВПБ) и Союз русской молодежи (СРМ) [1]. Белорусская центральная рада (БЦР) – центральная инстанция белорусской администрации на оккупированной территории. Президентом ее стал Р. Островский. Главными функциями БЦР были руководство школьным делом, культурой, социальной опекой. В ее подчинении находились созданные ранее БНС, СБМ, Белорусское научное общество. Официально БЦР подчинялись формирования Белорусской краевой обороны [2]. Деятельность БЦР прекратила после Второго Всебелорусского конгресса 27 июня 1944 г. Таким образом, структуры белорусского коллаборационизма были созданы национал-радикалами, бежавшими в свое время в Германию, и местными коллаборационистами. Их объединяло одно – ненависть к советскому строю и желание любым способом, даже кровавыми деяниями против белорусского народа, получить политическую власть в «самостоятельной Беларуси».

Список литературы

1. Структуры белорусской коллаборации [Электронный ресурс]. – Режим доступа: <http://archives.gov.by/index.php?id=717212> – Дата доступа: 02. 10. 2019.
2. Соловьев, А. Белорусская Центральная Рада: создание, деятельность и крах / А. Соловьев. – Минск, 1995. [Электронный ресурс] – Режим доступа: <http://maintracker.org/forum/viewtopic.php?> – Дата доступа: 10. 11. 2019.

*Войтешик Анастасия Сергеевна, студентка 1 курса факультета информационных технологий и управления БГУИР, avojtesik1@gmail.com.
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeve@bsuir.by*

ИСТОРИЯ МОЕЙ СЕМЬИ СКВОЗЬ ПРИЗМУ ИСТОРИИ XX ВЕКА

В тексте рассказывается об основателях семьи автора.

Основателями нашего рода мы считаем Сыся Михаила Евсеевича и Сысой (Чирун) Ольгу Адамовну. Мои прадед и прабабушка родились под Глуском. Родителями Михаила были Сысой Евсей и Юлия. Евсей воевал и в Первую, и во Вторую мировую войну. Юлия Сысой в годы Великой Отечественной войны находилась на оккупированной территории и скрывалась в подвале еврейскую семью. Сысой М.Е., мой прадед, принимал участие в Керченской операции. Плав средство его команды было подорвано немцами. Его вынесло на берег, и, когда он очнулся, обнаружил, что полностью седой. Воевал на Кавказе. Был награжден медалями: «За оборону Кавказа» и «За боевые заслуги». Войну закончил в Чехии. До 1961 г. служил в армии. Вышел в запас в звании майора. Работал в Могилевском облисполкоме начальником военного хранилища. Родителями прабабушки Ольги были Чирун Адам Фёдорович и Чирун Дарья Саввишна. До Первой мировой у них родилось двое сыновей – Гаврила (1912 г.) и Василий (1914 г.). В 1915 г. Адама призвали в армию, и тогда же он попал в плен. После освобождения вернулся на родину, переехал с семьёй на хутор, основал хозяйство по образцу фермеров. В 1919 г. у него родился сын Иван, а в 1924 г. – моя прабабушка Ольга Адамовна Чирун. В 1933 г. семья была раскулачена. Старший сын – Гаврила войну встретил в звании младшего лейтенанта 239-ого стрелкового полка 27-й стрелковой дивизии 3-й армии Западного фронта. 22 июня 1941 г., его часть находилась возле границы и была обстреляна немецкой артиллерией. Он пытался прорваться к основным силам. Был контужен, попал в лагерь под Белостоком. Всю войну провел в разных концлагерях, в том числе и Дахау. Официально чис-

лился пропавшим без вести [1]. Освободили лагерь американские войска. В СССР он попал в фильтрационный лагерь. В ходе расследования за Гаврилу поручился советский офицер, которому он помог бежать из концлагеря в Польшу. Иван Адамович прошел всю войну. Он начал службу в звании сержанта госбезопасности и закончил капитаном СМЕРШа. Имел награды: медаль «За боевые Заслуги» и два ордена Красной Звезды. После войны являлся первым секретарём Осиповичского района, директором школы. В Осиповичах одна из школ носит его имя. Ольга Адамовна Чирун всю войну провела на оккупированной территории, была связной в партизанском отряде. Сысой Михаил Евсеевич и Чирун Ольга Адамовна заключили брак в 1946 г. У них родились три дочери. В 1949 году моя бабушка Людмила. О судьбах членов семьи со стороны дедушки информации меньше. Отец моего деда – Воробьёв Матвей Данилович (1919 г. р.) участвовал в «зимней войне» с Финляндией. В начале Великой Отечественной войны оказался в окружении, успешно вышел из него. Попал в диверсионную школу. В 1942 г. был заброшен на оккупированную территорию Беларуси. Партизанил. Участвовал во взятии Кёнигсберга, военной операции в Маньчжурии, где воевал с японскими войсками. Своей жизнью мы обязаны нашим дедам и прадедам. Я горжусь своими предками. Это люди, которые храбро сражались за свое Отечество. Они прошли через ужас и кошмар военных лет, выстояли и победили.

Список литературы

1. Центральный архив Министерства обороны РФ. – Фонд 33, номер описи 11459, номер дела 364. – л. 103.

Воробьёва Анна Денисовна, студентка 1 курса факультета информационных технологий и управления БГУИР, vorobyovaad29@gmail.com
Научный руководитель: Сугако Наталья Анатольевна, кандидат исторических наук, доцент кафедры гуманитарных дисциплин БГУИР, Sugako@bsuir.by

САЛОМЕЯ РЕГИНА РУСЕЦКАЯ: ЖИЗНЕННЫЙ ПУТЬ И ДЕЯТЕЛЬНОСТЬ

Доклад посвящен одной из интереснейших женщин XVIII в. – Саломее Регине Русецкой, первой практикующей женщине-врачу в Речи Посполитой.

XVIII столетие вошло в историю как век авантюристов, страшных войн и балов маскарадов, эпидемий и научных открытий, вольности и нравов просветительства. Пожалуй, ярче всех в истории Беларуси воплотила дух этой эпохи судьба девочки из семьи новогрудского мещанина Ефима Русецкого – Саломеи Регины. Саломея Регина не имела ни большого капитала, ни особых привилегий, но смогла посетить множество стран Европы и Азии. Она побывала в столице Османской империи Константинополе, столице Священной Римской империи германской нации Вене, в городах Видин, Вильно, Санкт-Петербург, Нарва, Ревель (Таллин), Рига, Каменец-Подольский. И везде она прославилась. Саломею-Регину с полным правом можно назвать писательницей. Ее перу принадлежала рукописная книга-дневник с длинным заглавием: «Echo na swiat podane czyli procedura podrozy...» («Эхо, представленное миру, или описание путешествий и приключений моей жизни, в честь и хвалу господу богу и святой троицы единого и святейшей матери Христа, господина моего и всех святых, написала Саломея-Регина де Пильштейн, медицины доктор и окулистка, в 1760 г. в Стамбуле»). В Беларуси книга вышла в 1993 г. в переводе Николая Хаустовича под названием «Авантюры майго жыцця». Однако прославилась Саломея Регина Русецкая как одна из первых практикующих женщин-врачей Речи Посполитой. «Доктор медицины и окулистки», как она себя называла, систематизировала знания в области медицины и фармакологии, гигиены и физического воспитания. Саломея прочла многоме-

дицинских справочников, освоила практику выписки рецептов на латыни, выучила турецкий язык, овладела мастерством лечения глазных болезней и добилась значительных успехов в офтальмологии. Она работала в гареме султана Османской империи, а также при российском и австрийском императорских дворах. Саломею отличала смелость – она не боялась лечить запущенные случаи заболеваний, хотя знала заранее, что при неудачном исходе может погибнуть сама. Она постоянно стремилась повышать свои знания: набиралась медицинских знаний еще в белорусской деревне, затем училась у своего первого мужа, у «вавилонского» лекаря, мальтийского врача и, вероятно, всюду, где бывала. Даты жизни Саломеи Регины Русецкой (Гальпир, Пильштыновой) были установлены лишь из ее косвенных замечаний в своем дневнике. Она родилась в 1718 г. Однако в 1760 г. следы авантюристки затерялись, и дальнейшая судьба талантливого врача, одаренной писательницы и бесстрашной путешественницы осталась неизвестной. Возможно, с ней приключилась очередная авантюра, записки о которой пока не найдены.

Список литературы

1. Грицкевич, В. П. Путешествия наших земляков. Из истории страноведения Белоруссии / В. П. Грицкевич. – Минск: Наука и техника, 1968. – С. 36–59.
2. Грыцкевіч В. Адысея навагрудскай лекаркі: Саломея Русецкая / В. Грыцкевіч. – Мінск: Навука і тэхніка, 1990. – 53 с.
3. Рублевская, Л. И. Рыцари и дамы Беларуси: исторические очерки / Л. И. Рублевская. – Минск: Мастацкая літаратура, 2013. – С. 116–121.

*Горнак Диана Владимировна, студентка 1 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, gornak.dianka@gmail.com.
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин Белорусского государственного университета информатики и радиоэлектроники, кандидат исторических наук, доцент, Mikalayeva@bsuir.by*

МОЯ МАЛАЯ РОДИНА – Г. БЕРЕЗОВКА

Доклад посвящен истории градообразующего предприятия г. Березовка – стекольного завода «Неман».

Берёзовка – город, история которого тесно связана с историей стекольного завода «Неман». В 1883 г. помещик З. Ленский получил разрешение на строительство в своем имении Заенчицы стекольного завода. В 1885 г. пивные бутылки на нем производили 28 человек. В 1891 г. З. Ленский сдал «Старую Гуту» в аренду компании «Столле и Краевский», основателями которой были главный инженер Дядьковского завода Вильгельм Краевский и руководитель художественной мастерской Юлиус Столле. Компаньоны одновременно начали строительство собственного стекольного предприятия. В 1896 г. они завершили строительство «Новой Гуты» в четырех километрах от «Старой Гуты». Годом позже на заводе З. Ленского (он же «Неман А») над производством лампового стекла, предметов роскоши, столовой и аптекарской посуды трудились 164 человека, а на соседнем заводе «Неман Б», выпускающем бутылки и посуду, работали 129 человек. Совместно с партнерами владелец соседнего имения Островня Андрей Квецинский построил третью гуту. В 1899 г. З. Ленский, а в 1900 г. А. Квецинский и компания продали заводы Ю. Столле и В. Краевскому. «Столле и Краевский» закрыла «Старую Гуту», а название «Неман А» получил бывший завод А. Квецинского, была проведена газификация обоих предприятий. В 1905 г. В. Краевский утонул в реке Неман. Его доля в фирме перешла к вдове, которая уступила ее Ю. Столле. Он совместно с управляющим Германом Шаллем продолжил развитие стекольного производства. В год заводы Ю. Столле выпускали 1,2 млн штук сервизов, рюмок, бокалов, фужеров, блюд и пр., свыше 75 тыс. штук предметов роскоши, 400 тыс. штук стеклянных ламп и до 6 тыс. ящиков оконного листового стекла.

Первая мировая война негативно отразилась на деятельности предприятий. В 1915 г. заводы в Березовке были сожжены во время отступления российских войск. Далее следовали вывод кайзеровских войск, Гражданская и польско-советская война. В результате Рижского мирного договора 1921 г. заводы в Берёзовке остались на территории Польши. Благодаря усилиям Ю. Столле и его сыновей удалось восстановить производство. Этому помог госзаказ на производство изоляторов. В 1926 г. восстановили довоенную численность рабочих. Ю. Столле не стало в 1927 г. (72 года). Руководство производством перешло в руки его сыновей: Бронислава и Феликса. В 1930-е гг. компания братьев Столле стала лидером стекольной отрасли Польши. Предприятию удалось преодолеть мировой кризис 1920 – 1930-х гг. Владельцы вложили огромные средства в благоустройство Берёзовки. Они построили заводскую контору, дом опеки матери и ребенка с детским садиком, магазин, помещение пожарной части, госпиталь, костел и семилетнюю школу с библиотекой. Во время Великой Отечественной войны предприятие работало на нужды Рейха, но в 1943 г. его сожгли советские партизаны. После освобождения Беларуси завод был перестроен и возобновил производство в 1945 г. как один из трех крупнейших хрустальных заводов СССР.

Список литературы

1. Память. Лида и Лидский район: Исторически – документальные хроники городов и районов Беларуси / Редкол.: В. Г. Баранов, Г. К. Киселев, М. У. Малец, Б. В. Ульянка; худ. Э. Э. Жакевич. – Минск: Беларусь, 2004. – 566 с.; илл.

Железовский Владислав Юрьевич, студент 1 курса факультета информационных технологий и управления БГУИР, zhele.vlad@mail.ru.
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by

ОБРАЗ ИТ-СПЕЦИАЛИСТА В ПРЕДСТАВЛЕНИЯХ СТУДЕНТОВ

В работе представлены результаты пилотажного исследования стереотипов, которые сложились об ИТ-специалистах. На основании опыта исследователей сформулированы описательные гипотезы об образе специалистов данной отрасли. Профессиональная социализация будущих специалистов и начало карьеры в сфере информационных технологий находится под влиянием этих стереотипов, что оказывает влияние на их поведение и в сфере образования и в трудовой сфере.

За относительно короткое время своего существования информационная сфера буквально обросла стереотипами. Причем оценки этой сферы, преобладающие в общественном сознании, устоялись скорее стихийно, и оценить в какой мере они представляют реальную ситуацию сложно. Предвзятое отношение к данной профессиональной сфере определяет ряд поведенческих практик абитуриентов ВУЗов, и молодых специалистов.

В данном исследовании была предпринята попытка собрать мнения и взгляды о такой социальной группе, как ИТ-специалисты. Целью исследования стала попытка определения основных стереотипов, которые складываются об ИТ-специалистах. Всего было опрошено 412 респондентов, в числе которых 292(70.9%) респондента, профессия которых связана с ИТ-сферой и 120 (29.1%) респондентов, профессия которых не связана с этой сферой. В данной работе предложен общий анализ ответов, рассмотрение распределений ответов по указанным группам – задача последующего анализа. Респонденты опрашивались через интернет с платформы Google Forms.

В первую очередь были исследованы стереотипы об общих контурах профессии. Большинство респондентов склоняются к тому, что «программист» и «айтишник» являются словами-синонимами. Из общего количества опрошенных 242 (58.7%) респондента выбрали вариант ответа «Да», 140 (34%) респондентов выбрали вариант ответа «Нет», 22(5.3%) респондента выбрали вариант ответа «Не знаю». Профессионал ИТ по мнению отвечавших, обладает собственной спецификой. На вопрос об ассоциациях, связанных со специалистами ИТ-сферы, наибольшую популярность получил ответ: «постоянно чему-то учатся, ходят на дополнительные тренинги, и т.п.» 235 (57%) голосов. Вторым по популярности оказался ответ «Проявляют интерес ко всему новому», 171(41,5%) ответов. Также с небольшим отставанием среди самых популярных ответов можно выделить ответы «Знают цену времени» и «Очень логичные люди», которые набрали 160 (38,8%) и 140 (34%) голосов соответственно. Респондентам необходимо было выбрать не более трех ответов.

С точки зрения опрошенных профессия, связанная с информационной сферой, отнюдь не предполагает особых навыков работы с техническими средствами и аппаратами. 85.2% респондентов не согласны с высказыванием: «Все программисты хорошо разбираются в технике и способны починить всё». Только 11.2% согласились с этим мнением.

«Программист - это мужская профессия?». Из 412 опрошенных 160 (38.8%) респондентов выбрали вариант ответа «Нет, абсолютно не согласен (не согласна)», и 131 (31.8%) респондентов согласились с мнением «Скорее не согласен». Остальные придерживаются точки зрения определяющей программирование как мужскую профессию. Учитывая неполное согласие трети респондентов с тем, что в сфере ИТ нет гендерных стереотипов, можно предположить, что программирование интерпретируется преимущественно как мужское занятие.

Четвертой проверяемой гипотезой, определяющей специфику профессиональной группы ИТ специалистов, стало отношение к эмиграции. Большая часть (51.2%) опрошенных склоняются к тому, что эмиграционный потенциал не связан с профессией. При этом, довольно большая часть респондентов (38.1%) считают все же, что ИТ-специалисты наиболее подвержены идеям эмиграции. И лишь малая часть (8.5%) считает, что эмигрировать ИТ-специалисты желают в меньшей мере, чем представители других профессий. Такая статистика позволяет предполагать, что гипотеза о существовании стереотипа: «желание людей идти в ИТ-сферу связано с желанием эмигрировать» не подтвердилась.

Вторым блоком гипотез стал ряд предположений о положении группы специалистов ИТ сферы на рынке труда. Большая часть респондентов даёт оптимистичный прогноз для специалистов данной сферы. Ответ «безусловно, востребованность специалистов этой сферы будет расти», дали 64,8% опрошенных. Часть ответов (33%) была отдана варианту «Думаю, что востребованность останется в той же степени, что и сейчас».

Наиболее важным преимуществом на рынке труда в ИТ-сфере большая часть респондентов (44.2%) считают опыт. Также довольно большая группа (29.1%) считает, что у специали-

стов данной области существуют особые способности, которые делают их более конкурентноспособными по сравнению с их коллегами и 17% считают, что специальное образование также может быть преимуществом. Эта ситуация ставит в заведомо невыгодные условия студентов дневного отделения. Принимая представление о том, что главным требованием при трудоустройстве является наличие опыта, они встают перед дилеммой: прилежное обучение или необходимость подработки. .

Из всей совокупности опрошенных 194 (47.1%) респондента считают, что зарплаты в сфере ИТ значительно выше, чем средние, и лишь 3.9% опрошенных оценили зарплаты как такие же, как в среднем в экономике. За вариант ответа «Зарплаты ниже, чем в среднем в экономике» проголосовал 1 человек. Большинство опрошенных 285 человек (69.2%) считают такой уровень заработной платы в ИТ-сфере оправдан. Сомневаются в этом положении 23.3% опрошенных, и 4.6% считают этот уровень зарплат не оправданным.

Таким образом, специалист ИТ сферы, по мнению опрошенных это скорее всего программист, чаще мужчина, не обладающий особыми навыками работы с техническими средствами и аппаратами. «Айтишник» постоянно чему-то учит-

ся, ходит на дополнительные тренинги. Полученные результаты опровергают предположение о том, что получение специальности в сфере информационных технологий предполагает склонности к эмиграции. ИТ сфера весьма привлекательна для абитуриентов, в связи с тем, что в представлениях она определяется как сфера высоких заработных плат. Более того, уровень зарплат здесь интерпретируется как оправданный. Преимуществом на рынке труда, которым может обладать выпускник ВУЗ, являются особые его способности. Достаточно редким является мнение, о том, что только специальное образование также может облегчить получение места работы. Это особым образом влияет на образовательные стратегии студентов. Студенты видят сочетание работы и учебы как необходимость. В этой ситуации достижение успеха в образовательной и профессиональной сферах весьма сложно, и можно прогнозировать, что некоторые другие области самореализации студентов могут быть депривированы.

Список литературы

1. Аверьянов Л. Я. Социология: искусство задавать вопросы. / Л. Я. Аверьянов – М., 1998. –189 с.
2. Килин П. М. Статистические методы обработки данных: учебное пособие / П. М. Килин, Н. И. Чекарева. – Тюмень: ТюмГНГУ, 2013. – 128 с.

Жук Екатерина Владимировна, студентка 2 курса факультета информационных технологий и управления БГУИР, kate.manager19@gmail.com.

Гулида Марта Андреевна, студентка 3 курса Факультета компьютерных систем и сетей БГУИР, gulidamarta@gmail.com.

Иванова Анна Сергеевна, студентка 2 курса факультета информационных технологий и управления БГУИР.

Бондарева Татьяна Олеговна, студентка 3 курса Факультета компьютерных систем и сетей БГУИР.

Научный руководитель: Пацеева Анастасия Георгиевна, кандидат социологических наук, доцент кафедры гуманитарных дисциплин БГУИР, anastasiapaceeva@yandex.ru.

ВАЦЛАЎ ЛАСТОЎСКИ: ЖЫЦЁВЫ ШЛЯХ І ДЗЕЙНАСЦЬ

Даклад прысвечаны характарыстыцы жыцця і дзейнасці Вацлава Ластоўскага, беларускага грамадскага і палітычнага дзеяча, пісьменніка, гісторыка, філалага, літаратуразнаўца, этнографа, акадэміка Беларускай акадэміі навук

У самым пачатку XX ст. на беларускім літаратурным небасхіле ўзыхляла новая яркая зорка – Вацлаў Ластоўскі. На працягу амаль чвэрці стагоддзя імя яго не сыходзіла са старонак перыядычнага друку. У больш чым 300 публікацыях В. Ластоўскі паўставаў як выдатны пісьменнік, крытык, публіцыст, гісторык, мовазнаўца. Пазней ён выявіў сябе як паэт, філалаг, этнограф, выдавец, перакладчык, бібліёграф. У гісторыі Беларусі В. Ластоўскі застаўся і як актыўны грамадскі і дзяржаўны дзеяч. Ён з’яўляўся прэм’ер-міністрам (1919-1923 гг.) абвешчанай у 1918 г. Беларускай Народнай Рэспублікі, быў адным з першых дырэктараў Беларускага дзяржаўнага музея, правадзейным членам Інстытута Беларускай Культуры (1927 г.) і адным з першых акадэмікаў Беларускай акадэміі навук (1928 г.), працаваў яе сакратаром (1928-1929 гг.).

Восенню 1883 г., 27 кастрычніка паводле старога стылю, а паводле новага – 8 лістапада, у засценку Калеснікава, на Дзісеншчыне, у сям’і Юстына і Ганны Ластоўскіх нарадзіўся сын Вацлаў. Першую навуку чытацка-складаць хлопчык лёгка адолеў дома, і бацькі аддалі сына ў бліжэйшае мястэчка Стары Пагост, дзе месціліся належныя ўстановы. У ліку іх значылася парафіяльная школа. Пасля яе заканчэння ў 1894 г. здольны падлетак спрабаваў, па няпэўных звестках, атрымаць адукацыю ў Дзісенскай павятовай вучэльні.

Юнацтва В. Ластоўскага праглядаецца пункцірна і непазлядоўна: хлопчык на паслугах у адной з віленскіх вінных крамак; канцылярыст казёнай установы ў Шаўлях... Але яго вабіла навука. Юнак дабраўся да Санкт-Пецярбурга – найбуйнейшага культурнага асяродка Расійскай імперыі пачатку XX ст., дзе ўладкаваўся бібліятэкарам прыватнай студэнцкай бібліятэкі, адначасова зазіраючы ў аўдыторыі ўніверсітэта паслухаць лекцыі славуных прафесараў. Не маючы грошай, каб плаціць за ўніверсітэцкую адукацыю, на лекцыі наведваўся «зайцам» (1904-1905 гг.). Пэўнымі застаюцца два факты: у 1902 г. юнак стаў членам Польскай сацыялістычнай партыі Літвы і Беларусі, а ў 1903 г. узяў шлюб з Марыяй Іваноўскай (Іванаўскайтэ, 1872-1957), пазней вядомай пад агульным са старэйшай сястрой Зосяй псеўданімам летувіскай пісьменніцай Лаздзіну Пяледай. Нейкі час маладыя жылі ў мястэчку Парагай (поўнач Летувы). Пажыў Вацлаў і ў Рызе. Там у 1906 г. спрабаваў здаць экзамены на атэстат сталасці, аднакняўдала. Да-

лей працягваў паглыбляць веды нястомнай самаадукацыяй. У 1906 – 1908 гг. В. Ластоўскі быў членам Беларускай сацыялістычнай грамады. З 1902 г. В. Ластоўскі пачаў занатоўваць «цікавыя» словы. Напачатку запісваў іх для сябе самога, бо вельмі ж «асабіста цікава было слова само па сабе». У сакавіку 1909 г. В. Ластоўскі прыехаў у Вільню, куды быў запрошаны на пасаду сакратара рэдакцыі «Нашай Нівы», на старонках якой выйшла і першая публікацыя В. Ластоўскага пад псеўданімам «Власт». 19 лістапада 1909 г. «Наша Ніва» надрукавала яго апавяданне-абразок «Нарадзіны», а ў наступным, 48-м нумары апавяданне «Зайчык». «Зайчык» стаў не толькі дэбютам В. Ластоўскага-празаіка, але і адным з пачаткаў беларускай дзіцячай літаратуры XX ст. Маленькаму чытачу пазней будуць прысвечаны асобныя казкі пісьменніка «Князь Барыс і чорт», «Дзед і ўнук», «Певень і гуся», «Вуж куртаты і мужык багаты» і цэлыя чытанкі, складзеныя ім падчас кіраўніцтва Беларускай выдавецкай таварыствам у Вільні (Кніжыца для беларускіх дзіцячых дзяцей; Rodnyje Ziarniaty: knižyca dla školnaha čytannia; абедзве Вільня, 1916). Рост Власта ішоў шпарка. 1 студзеня 1910 г. сваім навагоднім нумарам «Наша Ніва» распачала друкаваць яго «Кароткую гісторыю Беларусі», з выхадам якой беларуская літаратура нашаніўскай пары набыла надзейныя арыенціры, каб упэўнена крочыць у будучыню. У ліпені 1914 г. пачалася Першая сусветная вайна. Недзе ў гэты час драматычныя змены адбыліся і ў асабістым лёсе В. Ластоўскага: фактычна распаўся шлюб Вацлава і Марыі. Хоць бацькоўскі абавязак і чулае сэрца па-ранейшаму прыводзілі Вацлава да дзвюх мілых шчабятух – Ганкі і малодшанькай Стаські. А пакуль у «Нашай Ніве» рэй вёў новы рэдактар І. Луцэвіч (Я. Купала). В. Ластоўскі ж становіцца фактычным уладальнікам «Беларускай кнігарні», кантралюючы выданне школьных падручнікаў і чытанак. Ды ён не толькі кантраляваў, але і сам удзельнічаў у іх стварэнні: склаў «кніжку для школьнага чытання» «Родныя зярняты», «першую пасля лемантара чытанку» «Незабудка». Знаходзячыся ў эміграцыі ў 1920-1923 гг. В. Ластоўскі наведваў з дыпламатычнымі місіямі Бельгію, Германію, Ватыкан, Італію, Чэхаславакію, Францыю, Швейцарыю. На кожнай урачыстай аўдыенцыі, міжнароднай канферэнцыі, палітычнай нарадзе ён

выступаў як беларус, ад імя Беларусі і дзеля будучыні Беларусі. Пакінуўшы ў 1923 г. пост прэм'ер-міністра БНР В. Ластоўскі ўзяўся за выданне «месячніка літаратуры, культуры і грамадскага жыцця» пад назвай «Крывіч». У Коўне ён выдаў капітальны фаліант «Гісторыі беларускай (крыўскай) кнігі» (1926), з якім прыехаў у Менск на Акадэмічную канферэнцыю па рэформе беларускага правапісу. Тут канчаткова выспеў намер пераехаць у БССР, што ён і ажыццявіў ў сакавіку 1927 г. У Менску В. Ластоўскаму прапанавалі пасаду дырэктара Беларускага дзяржаўнага музея і загадчыка кафедры этнаграфіі пры Інстытуце беларускай культуры. Вучоны арганізаваў пошукавыя экспедыцыі. Падчас адной з іх, у Полацк, быў выратаваны крыж Еўфрасінні Полацкай. У сталым узросце В. Ластоўскі апроч іпастасі паэта і перакладчыка раскрыўся і як мемуарыст. Ягонныя каларытныя ўспаміны пра маленства і вучобу ярка малююць вобраз беларускага мястэчка канца XIX ст., уваскрашаюць шматлікія звычаі, абрады, якія цяпер не захаваліся («Старабеларуская вясковая школа», 1923; «Успаміны з нядаўняй мінуўшчыны», 1925). Незабыўныя згадкі пра Я. Купалу і М. Багдановіча ўзнаўлялі легендарную атмасферу, напоўненую духам пазітывізму, у якой працаваў калектыў «Нашай Нівы». 21 ліпеня 1930 г. В. Ластоўскі быў арыштаваны ДПУ БССР па справе «Саюза вызвалення Беларусі». Пастановай Калегіі АДПУ СССР ад 10 красавіка 1931 г. высланы на 5 гадоў

у Саратаў. Працаваў загадчыкам аддзела рэдкіх рукапісаў бібліятэкі Саратаўскага ўніверсітэта. Паўторна арыштаваны 20 жніўня 1937 г. упраўленнем НКВС Саратаўскай вобласці, В. Ластоўскі, праз 5 месяцаў катаванняў, 23 студзеня 1938 г. як «агент польскай разведкі і ўдзельнік нацыяналі-фашысцкай арганізацыі» рашэннем Ваеннай Калегіі Вярхоўнага Суда СССР быў асуджаны да вышэйшай меры пакарання. 23 студзеня 1938 г. прысуд прывялі да выканання ў Саратаве. Па першым прысудзе рэабілітаваны 10 чэрвеня 1988; па другім — 16 верасня 1958. Адноўлены ў званні акадэміка АНБ у 1990.

Спісок літаратуры

1. Янушкевіч, Я. Неадменны сакратар Адраджэння: Вацлаў Ластоўскі / Я. Янушкевіч. – Мінск: Навука і тэхніка, 1995. – 68 с.; іл.
2. Некрашэвіч, С. [Рэцэнзія] / С. Некрашэвіч // Асвета. – 1926. – № 5. – С. 168. Рэц. на кн.: Ластоўскі, В.Ю. Гісторыя беларускай (крыўскай) кнігі: спроба паясніцельнай кнігапісі ад канца X да пачатку XIX стагоддзья / В. Ю. Ластоўскі. – Коўна: Друк. Сакалоўскага і Лана, 1926. – 776 с.
3. Власт (Ластоўскі, В. Ю.). Кароткая гісторыя Беларусі [1129–1905 гг.]: 3 40 рыс. / Власт; Аўт. паслясл. А. Грыцкевіч. – Мінск: Універсітэцкае, 1992. – 126 с.:іл. – Факс. выд.: Вільня: Друкарня Марціна Кухты, 1910.
4. Барысенка, В. У. Творчасць В. Ластоўскага ў ідэйна-мастацкім кантэксце беларускай літаратуры пачатку XX ст.: аўтарэферат дысертацыі на атрыманне вучонай ступені канд. філал. навук : 10.01.01/ В.У. Барысенка; БДУ. – Мінск, 2000. – 21 с.

Кірылушкін Аляксандр Генадзевіч, студэнт 1 курса факультэта інфармацыйных тэхналогій і кіравання БДУІР, Aleksandrkirilushkin@gmail.com
Мінеев Уладіслаў Аляксандравіч, студэнт 1 курса факультэта інфармацыйных тэхналогій і кіравання БДУІР, vladislav.mineev.07@mail.ru
Навуковы кіраўнік: Мікалаева Людміла Віктараўна, загадчык кафедры гуманітарных дысцыплін БДУІР, кандыдат гістарычных навук, дацэнт, Mikalayeva@bsuir.by

«РЫСКУН» И «ШПЕГ» В ИСТОРИИ РАЗВЕДЫВАТЕЛЬНОЙ СЛУЖБЫ ВКЛ

В докладе представлена информация о малоизвестных страницах истории спецслужб ВКЛ.

История спецслужб Великого Княжества Литовского (далее – ВКЛ) является малоизученной. Потому можно составить примерную картину работы разведки и контрразведки на территории ВКЛ. Одним из основных источников информации были послы, которые помимо основной деятельности – представителя своего государства, успевали следить за обстановкой в стране, в которой находились. По возвращении на родину послы докладывали правителю результаты их основной деятельности и составляли отчеты. Информация могла передаваться через служащих рангом поменьше и гонцов. Контрмеры против такой деятельности послов обычно представляли собой их поселение подальше от городов и сел. Выделяемая государством, принимающим послов, охрана отвечала за их защиту и обеспечение, а также за контроль отсутствия контактов посла и местного населения. Не менее важным источником информации были купцы. Они были мобильнее послов и обладали немалым количеством связей, как того требовало ведение торговой деятельности. Хотя купцам также старались ограничить маршруты передвижения, мобильность их сохранялась, и порой торговцам удавалось приехать и уехать без уведомления принимающей стороны. Третьим немаловажным источником были информаторы в лице служащих и жителей других государств. Обычно таким способом передавалась информация между дружественными государствами, тем не менее, она была не менее полезной, так как удавалось узнать что-нибудь о третьей, не совсем дружественной стороне. Обычные перебежчики и сбежавшие из тыла врага пленные также были источниками информации. Помимо лиц, сбор информации для которых был второстепенным видом деятельности, существовали так же «шпег»

и «рыскуны», сбор сведений для которых являлся основным видом деятельности. Конкретных определений этих терминов, к сожалению, нет, и некоторые исследователи считают их тождественными. Мало информации и о том, кем по социальному положению были рекрутируемые, и какая была оплата труда таких разведчиков. Тем не менее, ориентируясь на редкие документы, эти термины можно разграничить. «Шпег» отвечал перед воеводой, гетманом или непосредственно великим князем, занимался больше разведкой на вражеской территории, собирая информацию как по общей обстановке в государстве, так и по различным укреплениям и другим материальным объектам. Иногда такие кадры набирались из числа граждан государства, в котором нужно было проводить разведку. «Рыскун», в свою очередь, подчинялся волостному наместнику, и скорее представлял собой контрразведывательного агента, в обязанности которого входило отлавливать и допрашивать различного рода перебежчиков, пленных и нарушителей границ. Действовали они на территории своей волости или, если того требовали обстоятельства, на территории соседней. Таким образом, на территории ВКЛ и соседних государств в позднем средневековье возросла роль разведывательных служб и сбора информации, потому государства старались использовать любую возможность для получения данных об обстановке на территории союзника или противника.

Список литературы

1. Bardach, J. Ryskun i szpieg: Z dziejów służby wywiadowczej w Wielkim Księstwie Litewskim XVI stulecia // Bardach J. Studia z ustroju i prawa Wielkiego Księstwa Litewskiego XIV–XVII w./ J. Bardach. – Warszawa: PWN, 1970. – S. 391 – 418.

*Ягода Владислав Валентинович, студент 1 курса факультета информационных технологий и управления БГУИР, sgttomatocool@gmail.com.
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by*

ГІСТОРЫЯ ПЧАЛЫ Ў ГІСТОРЫІ МАЁЙ КРАІНЫ

Школьныя падручнікі па гісторыі вельмі скупа і неапраўдана мала даюць звестак аб бортніцтве. У гэтым я бачу праблему і асноўны матыў выбару тэмы даследавання.

Аб'ектам даследавання былі адабраныя і прааналізаваныя тэксты вучэбнай літаратуры па гісторыі, даведачнай літаратуры, было выдзелена 40 дакументаў, якія адносяцца да розных перыядаў айчыннай гісторыі, адабраныя неабходныя ілюстрацыі да тэксту.

Прадметам даследавання з'яўляліся гістарычныя аспекты пчалярства і бортніцтва. Метады даследавання: збор патрэбнай інфармацыі шляхам падбору неабходных гістарычных дакументаў, аналітычны, апісальны, параўнальны метады. Гіпотэза: калі мёд і воск доўгі час (некалькі стагоддзяў) быў стабільным экспартным таварам, то гэта стала магчымым толькі дзякуючы бортніцтву, як форме культурнага пчалярства, а не простаму збору мёду дзікіх пчол.

Мэтай даследавання з'яўляецца спроба паказаць, што бортніцтва было даволі развітай формай пчалярства, самастойнай галіной гаспадарання, з экспартна-арыентаванай прадукцыяй, з дасканалым прававым забеспячэннем. Задачы даследавання: выявіць, што пчалярства было адным са старажытнейшых заняткаў чалавека; вызначыць асаблівасці пераходу да новага этапу бортніцтва як культурнага пчалярства; раскрыць на аснове этнаграфічных матэрыялаў тэхніку і тэхналогію бортніцтва і ўзровень яе развіцця на беларускіх землях; устанавіць ступень нарматыўна-прававога рэгулявання пчалярства ў XVI-XVII стагоддзях. Тэма гістарычнага агляду пчалярства, бортніцтва ў Беларусі распрацавана недастаткова. Найбольш цікавыя і карысныя звесткі па тэме даследавання знойдзены ў зборніках гістарычных дакументаў.

Такім чынам, найбольш поўным і адпаведным азначэннем бортніцтва будзе наступнае: "Бортніцтва – гэта першая форма культурнага пчалярства, пры якім пчол гадавалі, развадзілі і трымалі ў штучных дуплах". Пчала

пакінула значны след у гісторыі маёй краіны, след, які вельмі цяжка пераацаніць: у старажытных летапісах, у старажытных прававых дакументах, у міфалогіі, геральдыцы, тапаніміцы, у прозвішчах людзей, у звычаях і традыцыях людзей, у занятках людзей. Пчалярства – адно са старажытнейшых заняткаў чалавека. У эпоху ўсіх вядомых народаў старажытнасці сустракаюцца звесткі аб пчолах і атрымліваемых ад іх прадуктаў – мёдзе і воску. Гісторыя пчалярства такая ж сталая, як і гісторыя чалавецтва. Бортніцтва не правільна ўяўляць як прымітыўнае збіральніцтва мёду дзікіх пчол без якіх-нібудзь прыстасаванняў і прыладаў. Тэхнічна і тэхналагічна бортніцтва было складаней, чым паляванне. Апякунства бортніцтва нормамі права на працягу некалькіх стагоддзяў сведчыць аб вялікім значэнні гэтага віду гаспадарчай дзейнасці. Такім чынам, мёд і воск доўгі час быў стабільным экспартным таварам, гэта стала магчымым дзякуючы бортніцтву, як форме культурнага пчалярства, а не простаму збору мёду дзікіх пчол.

Спісок літаратуры

1. Белоруссия в эпоху феодализма. Т.1, с древнейших времен до середины XVII века, – Минск: Академия наук БССР, 1959.
2. Белоруссия в эпоху феодализма. Т. 3, 1772 – 1860. – Минск: Академия наук БССР, 1961.
3. Хрестоматия по истории Беларуси в двух частях. – Минск: Издательский центр БГУ, 2008. – Ч. 1. С древнейших времен до 1917 года.
4. Этнаграфія Беларусі. Энцыклапедыя. – Минск: БелСЭ імя П. Броўкі, 1989.
5. Бабіч А. Ю. Пчалярства як культурная форма ляскарыстання беларусаў у XX – пачатку XXI стагоддзя. [Электронны рэсурс]. – Режим доступа: <https://cyberleninka.ru/article/n/pchalyarstva-yak-kulturnaya-forma-lesakarystannya-belarusa-u-xx-pachatku-xxi-st>.

Липинская Ольга Дмитриевна, студентка 1 курса экономического факультета БГУИР.
Научный руководитель: Куракевич Наталья Ивановна, доцент кафедры гуманитарных дисциплин БГУИР, кандидат исторических наук, kurakevich@bsuir.by.

ГЕРАІЧНАЯ АБАРОНА БРЭСЦКАЙ КРЭПАСЦІ ЛЕТАМ 1941 Г.

Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі г. Мінск, Рэспубліка Беларусь

У ліку месцаў, дзе адпор ворагу быў аказаны ўжо першы дзень Вялікай Айчыннай вайны, была Брэсцкая крэпасць. Да 22 чэрвеня 1941 г. у крэпасці дыслакавалася 8 стралковых батальёнаў, дзін разведвальны, адзін артылерыйскі полк і два артылерыйскія дывізіёны (СТА і СПА), некаторыя спецыяльныя часткі стралковых палкоў і падраздзяленні карпусных частак, зборы прыпісанага складу 6-й Арлоўскай і 42-й стралковай дывізіі 28 стралковага корпуса 4-й арміі, падраздзяленні 17-га Чырвоназнамёнага Брэсцкага пагранічнага атрада, 3-га асобнага інжынернага палка, частка 132-го батальёна канвойных войскаў НКВС, штабы частак (штабы дывізіі і 28-га стралковага корпуса размяшчаліся ў Брэсце), усяго каля 9 тысяч чалавек, не лічачы сем'яў вайскоўцаў (300 сем'яў). З нямецкага боку штурм крэпасці быў даручаны 45-й пяхотнай дывізіі (каля 17 тысяч чалавек) 12-га армеемскага корпуса 2-й армейскай групы пры падтрымцы танкаў, артылерыі і авіяцыі. Крэпасць належала захапіць да 12 гадзін 22 чэрвеня 1941 г. У 4 гадзіны 15 хвілін 22 чэрвеня 1941 г. пачаўся артылерыйскі абстрэл крэпасці. У выніку былі знішчаны склады, вадаправод, перарвана сувязь, нанесены буйныя страты гарнізону. У 4 гадзіны 45 хвілін пачаўся штурм. Нечаканасць атакі прывяла да таго, што адзінага скаардынаванага супраціўлення гарнізон аказаць не змог і быў раздзелены на некалькі частак. Моцнае супраціўленне праціўнік сустраў на Валынскім і асабліва Кобрынскім умацаванні, дзе справа дайшла да штыкавых атак. Да 7 гадзін 42-я і 6-я стралковыя дывізіі пакінулі крэпасць і г. Брэст, аднак мноству вайскоўцаў гэтых дывізіі так і не ўдалося выйсці з крэпасці. Менавіта яны працягвалі змагацца. Па

ацэнках даследчыкаў, з крэпасці выйшла каля 6 тысяч чалавек, каля 9 тысяч заставалася ў ёй. Да вечара 24 чэрвеня праціўнік авалодаў Валынскім і Цярэспальскім умацаваннямі, а рэшткі гарнізона апошняга ноччу пераправіліся ў Цытадэль. Так абарона сканцэнтравалася ў Кобрынскім умацаванні і Цытадэлі. На Кобрынскім умацаванні да гэтага часу ўсе абаронцы (каля 400 чалавек пад камандаваннем маёра П.М. Гаўрылава) сканцэнтраваліся ва Усходнім форце. Штодзённа абаронцам крэпасці даводзілася адбіваць 7-8 атак, прычым прымяняліся агняміты. 26 чэрвеня паў апошні ачаг абароны Цытадэлі каля Трохарачных варот, 29 чэрвеня – Усходні форт. Арганізаваная абарона крэпасці на гэтым скончылася. Заставаліся толькі ізаляваныя групы і адзіночныя байцы. Адзін з надпісаў на сценах крэпасці сцвярджае: «Я умираю, но не сдаюсь. Прощай, Родина. 20/VII-41». 23 ліпеня 1941 г., г.зн. на 32-гі дзень вайны, у палон быў ўзяты маёр П.М. Гаўрылаў, які камандаваў абаронай Усходняга форта, па афіцыйных дадзеных – апошні абаронца Брэсцкай крэпасці. Такім чынам, падзвіг абаронцаў Брэсцкай крэпасці летам 1941 г. стаў яркім прыкладам патрыятызму і бясконцай адданасці Радзіме. Уклад абаронцаў крэпасці быў ацэнены і палітычным кіраўніцтвам краіны. 8 мая 1965 г. Брэсцкай крэпасці было прысвоена званне крэпасць-герой з уручэннем ордэна Леніна і медалём «Залатая Зорка».

Спісок літаратуры

1. Оборона Брестской крепости [Электронный ресурс]. – Режим доступа: <https://3mv.ru/publ/srazhenija/oborona-brestskoj-kreposti/19-1-0-12137>. – Дата доступа: 19.03.2020.

Ярошык Алясей Аляксандравіч, студэнт 1 курса факультэта інфармацыйных тэхналогій і кіравання БДУІР Mr.alex365@mail.ru
Навуковы кіраўнік: Мікалаева Людміла Віктараўна, загадчык кафедры гуманітарных дысцыплін БДУІР, кандыдат гістарычных навук, дацэнт, Mikalayeva@bsuir.by

БЕЛОРУСЫ ВО ФРАНЦУЗСКОМ ДВИЖЕНИИ СОПРОТИВЛЕНИЯ. ИСТОРИЯ КОЖЕМЯКИНА Ф. Ф.

В тексте содержится рассказ подвиге прадедушки автора Ф.Ф. Кожемякина.

Франция и Англия объявили войну Германии 3 сентября 1939 г. после ее нападения на Польшу, но военных действий не вели до весны 1940 г. Этот период получил название «странной войны». Военные действия начались только 10 мая 1940 г., а уже 22 июня 1940 г. Франция заключила перемирие с Германией. Уже летом 1940 г. во Франции возникло движение Сопротивления. Однако первоначально оно не было массовым и слишком активным. За пределами Франции движение Сопротивления при поддержке английского правительства возглавил генерал Ш. де Голль. Более значимой в это время была позиция коммунистической партии (далее – ФКП). Под руководством ФКП с лета 1940 г. на территории Франции стали формироваться военные отряды движения Сопротивления. FTP – les francs-tireurs-partisans («Французские стрелки партизаны») – крупнейшая военная организация движения Сопротивления во Франции, созданная в Бретани по инициативе ФКП в начале 1942 г. До войны Фёдор Кожемякин работал в школе учителем истории в деревне Лешня Копыльского района. В конце 1939 г. его призвали в армию. Он получил звание младшего политрука, закончив курсы в Башкирии. Последнее письмо Фёдора родные получили в июне 1941 г. Дома его ждали жена и маленькая дочка. Зимой 1943 г. немцы пригнали в бретонский порт Lorient группу советских военнопленных разбирать завалы местной электростанции. Весной 1943 г. при помощи французских подпольщиков из рабочего лагеря сбежали пять узников во главе с лейтенантом Ф. Кожемякиным. В итоге беглецы оказались на небольшой станции Gare de Saint-Nicolas-des-Eaux. Французы

отвели беглецов на ферму, где военнопленные провели следующие 14 месяцев (5 августа беглецы присоединились к 1-му батальону FTP). Долгий карантин пятерых советских солдат объяснялся просто – им не доверяли. В Бретани в немецких войсках было немало частей сформированных из русских добровольцев. Французы их называли «gusses blancs» – белые русские или казаки. Красными русскими местные называли советских военнопленных, которые бежали из лагерей и присоединились к партизанам. После того, как Ф. Кожемякин и его товарищи доказали свою ненависть к нацистам, их зачислили в роту французских партизан (маки). 6 июня 1944 г. в Нормандии высадились войска союзников. Район Лорьяна предстояло освободить партизанам. Группа, в которую входили бывшие военнопленные, сражалась в районах Hennebont и Lorient во французской Бретани. Ф. Кожемякин погиб 8 сентября 1944 г. под Lorient. Он похоронен в городке Lochrist во французской Бретани. О судьбе Ф. Кожемякина родные в Беларуси узнали только в 1970 г. В газете «Советская Белоруссия» была опубликована фотография и рассказ о белорусах, сражавшихся за свободу Франции в рядах французского Сопротивления.

Список литературы

1. История семьи Кожемякиных [Электронный ресурс]. – Режим доступа: <https://lavina.info/история-семьи-кожемякиных-e592c59691d0>. – Дата доступа: 23.02.2020.
2. Обратный отсчёт. Они сражались за Францию. – Режим доступа: <https://my.mail.ru/mail/lucab66/video/1381/3704.html>. – Дата доступа: 23.02.2020.
3. Франция помнит белорусских героев. – Режим доступа: <https://www.sb.by/articles/frantsiya-pomnit-belorusskikh-geroev.html>. – Дата доступа: 23.02.2020.

Никитина Алина Юрьевна, студентка 1 курса факультета информационных технологий и управления БГУИР, alinanikitina2015@mail.ru
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by

РИЖСКИЙ МИРНЫЙ ДОГОВОР

Доклад посвящен рассмотрению условий и последствий заключения Рижского мирного договора.

17 сентября 2019 г. исполнилось 80 лет epochальному событию – воссоединению Западной Беларуси с Белорусской Советской Социалистической Республикой. Что привело к тому, что часть Беларуси оказалась в составе Польши? Рассмотрим данные обстоятельства. Советско-польская война 1919 – 1920 гг. завершилась 18 марта 1921 г. подписанием в Риге мирного договора между РСФСР и Украинской ССР с одной стороны и Польшей с другой. Ему предшествовали мирные переговоры, на которых польская делегация потребовала установить границу по линии Дисна – Вилейка – западнее Минска – Лунинец – Сарны – Ровно – по реке Збруч; восточнее этой линии должна была быть нейтральная полоса шириной 30–40 километров. Советская делегация, после некоторых протестов, приняла эти условия. Рижский мирный договор вступил в силу после обмена ратификационными грамотами в Минске 30 апреля 1921 г. По его условиям граница Польши проходила значительно восточнее «линии Керзона», установленной Верховным Советом Антанты как восточная граница Польши, захватывая около трех тысяч квадратных километров в Полесье и на берегу Западной Двины с населением почти 14 миллионов человек (в большинстве, белорусов и украинцев). За вклад в экономику Российской империи Польша потребовала 300 миллионов рублей золотом и две тысячи паровозов. Советская сторона согласилась на 30 миллионов плюс 300 паровозов европейской колеи, 435 пассажирских и 8100 товарных вагонов. Стороны взаимно гарантировали невмешательство во внутренние дела друг друга и обязались не разрешать пребывания на своей территории организаций и сил, враждебных другой стороне. Польша должна была предоставить украинцам и белорусам языковые

и культурные права. Подлежали возвращению все польские капиталы и вклады в государственных, национализированных или ликвидированных банках, от ответственности по долгам Российской империи Польша освобождалась. Стороны договорились в течение полутора месяцев начать торговые переговоры, а дипломатические отношения установить немедленно. Передача основной части культурных ценностей завершилась соглашением от 16 ноября 1927 г. Проблемы с реализацией Рижского договора возникли почти сразу после его вступления в силу. Советско-польские торговые переговоры начались только в марте 1922 г. Однако торговый договор был подписан только 19 февраля 1939 г. Только в феврале 1934 г. диппредставительства сторон стали полноценными посольствами. Демаркация границы завершилась только в ноябре 1922 г., а полностью работа пограничной комиссии – в августе 1924 г. Репатриация военнопленных затянулась до осени 1923 г. Вопреки статье договора о предоставлении культурных прав белорусам и украинцам Польша еще зимой 1920 г. начала проводить политику колонизации новых окраин. В 1924 г. СССР заявил, что «преследования национальных меньшинств приняли массовый и систематический характер». Варшава отклонила эти претензии, объявив их вмешательством во внутренние дела. Таким образом, в 1939 г. была достигнута историческая справедливость и обе части Беларуси соединились в едином национальном государстве.

Список литературы

1. Рижский мирный договор 1921 [Электронный ресурс]. – Режим доступа: <https://ria.ru/20160318/1390906311.html>. – Дата доступа: 18.12. 2019.

Подлешук Матвей Сергеевич, студент 1 курса факультета информационных технологий и управления БГУИР, podleshuk.bsuir@ya.ru
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by

ФЕМИНИЗМ В БЕЛОРУССКОМ ОБЩЕСТВЕ: ИСТОРИЧЕСКАЯ РЕТРОСПЕКТИВА

Доклад посвящен рассмотрению вопросов равноправия женщин на белорусских землях и кратким биографиям некоторых выдающихся женщин.

Концепция женского равноправия была впервые сформирована и обоснована в конце XVI в., когда женщинам предоставлялась сравнительно большая свобода выбора, им прививался интерес к общественным делам. Статут ВКЛ 1588 г. провозгласил приоритет защиты интересов каждого свободного человека. С этого времени достоинство женщин подлежало судебной защите. Первые законы ВКЛ предусматривали равенство мужчин и женщин в аспекте прав на имущество и наследство, устанавливали наказание за насилие над женщиной. Но даже после этого смыслом жизни женщины являлось рождение ребенка и поддержание семейного очага. Никто и подумать не мог, что женщина может быть способна на большее.

В истории Беларуси было немало женщин, чья сила воли, добродушие и стремление добиться чего-то большего помогли сломать стереотипы, сделали вклад в развитие общества и изменили положение женщин в Беларуси.

Евфросиния Полоцкая – первая белорусская просветительница, одна из первых женщин на Руси, канонизированных в святые. В храмовом скриптории Полоцкого Софийского собора она переписывала книги. Монастыри, основанные ею, стали центрами образования в Полоцком княжестве. По заказу Евфросинии был создан крест с реликвиями, выполненный Лазарем Богшей. Также просветительница приобрела редкую и ценную икону Богородицы Эфесской.

Франциска Урсула Радзивилл – драматург, первая женщина-писатель в Польше и Беларуси. Княгиня обновила и оснастила замковую типографию в Несвиже, привела в порядок и увеличила запущенную замковую библиотеку, много книг заказала за границей. Княгиня напи-

сала множество стихов и пьес. Именно Франциска Урсула к числу развлечений во дворце добавила придворный театр. Она сама писала сценарии для постановок, зачастую выбирая поучительные сюжеты. Франциска Урсула оказала огромное влияние на развитие белорусского театра. Благодаря ей первые театральные постановки на белорусской земле были поставлены в 1746 г., раньше, чем в России и Польше. Саломея Русецкая – первая женщина-врач. Когда девушке было 14 лет, ее выдали замуж за врача-окулиста, с которым она вскоре уехала в Стамбул. Саломея сперва лишь помогала мужу, а потом и сама стала практикующим врачом. Свою жизнь, невероятные истории, которые с ней приключились, она описала в романе «Авантюры моей жизни». Эмилия Плятер – графиня и революционерка, которая стала известна как белорусская Жанна д'Арк. Жизнь Эмилии Плятер стала легендой, а ее имя – символом самоотверженной любви к Отечеству. В 1830-х годах 25-летняя девушка сама организовала партизанский отряд и повела его в бой. Девушка успела получить даже звание капитана, но неожиданно заболела и умерла. Помимо революционной деятельности она всегда увлекалась народным творчеством. История этих женщин сподвигла других на проявление себя, достижение, казалось бы, невыполнимых целей. Благодаря им мы сейчас можем довольствоваться равноправием. Женщины, конечно, слабый пол, но они способны на многое.

Список литературы

1. 9 известных женщин Беларуси [Электронный ресурс]. – Режим доступа: <https://adukar.by/news/izvestnye-zhenshchiny-belarusi-kratkie-biografii>. – Дата доступа: 20.03.2020.

Потапова Софья Александровна, студентка 1 курса факультета информационных технологий и управления БГУИР, ptv-sofia@yandex.by
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, Mikalayeva@bsuir.by

ВОСПРИЯТИЕ ПРОБЛЕМЫ СПРАВЕДЛИВОСТИ ГЛАЗАМИ СТУДЕНТОВ.

В работе представлены результаты пилотажного исследования восприятия проблемы справедливости во мнениях студентов. Несмотря на философский смысл этой проблемы, практическое её разрешение во многом зависит от того, какова специфика представлений о ней, и какие пути решения сложных этических ситуаций молодёжь считает приемлемыми.

Проблема справедливости рассматривалась философами с древнейших времен. Парадокс заключается в том, что справедливость для одного человека — это вопиющая несправедливость для другого. Предлагаемая работа стала попыткой разведывательного исследования представлений о справедливости молодых людей. В качестве выборочной совокупности выступили студенты 1-4 курсов ГГУ им. Ф. Скорины, БГУИР, БГЭУ. Было опрошено 115 человек. В ходе исследования респондентам было предложено сформулировать общее представление о справедливости. Чаще всего студенты понимают социальную справедливость как наличие равных условий для реализаций индивидуальных способностей 59%. Несколько реже справедливость связывается с равенством всех граждан перед законом 39%. Возможность воздействовать на выработку и принятие решений по важнейшим вопросам с понятием справедливости связывается ещё реже 27.8%. Около 20% респондентов считают, что социальная справедливость недостижима. Очевидно, убеждение в том, что справедливость недостижима связана с тем, что более половины опрошенных (68.7%) заявляют, что иногда сталкиваются с ущемлением своих прав без законных оснований. А шестая часть опрошенных (12.2%) практически ежедневно сталкиваются с дискриминацией. Наиболее часто встречающимися формами дискриминации являются: дискриминация по полу, дискриминация по возрасту, дискриминация из-за физических/психических особенностей. Пятая часть опрошенных студентов (22%) респондентов от-

носятся относят себя к какому-либо социальному меньшинству, 39% из них, считают, что отношение общества к данной социальной группе отрицательное. В отношении других социальных групп чаще всего по мнению студентов сталкиваются с несправедливостью малоимущие (44% ответов), инвалиды и больные (47%). Более 65% опрошенных считают, что общество устроено несправедливо. Практически половина ответивших (53%) считают, что попытка установления справедливости в социальных отношениях должна начинаться с установки: «Самим быть более честными, справедливыми, соблюдать законы». В качестве заключения можно отметить, две особенности представлений о справедливости, выявленных в ходе данного исследования. Во-первых, в самой большой степени студенты склонны связывать предвзятое отношение к людям с материальным статусом этих людей. Во-вторых, студенты не демонстрируют склонности разрушать установившиеся представления о справедливости путем революционных действий. Как и многие века, люди считают, что общество устроено несправедливо, задумываются о путях решения данной проблемы. Возможно, изменяются пути решения этих проблем.

Список используемых источников

1. Аверьянов Л. Я. Социология: искусство задавать вопросы. / Л. Я. Аверьянов – М., 1998. – 189 с.
2. Пионткевич Л. Ю. Проблема справедливости в социальной философии / Л. Ю. Пионткевич // Журнал «Философия и общество» – М., 2009. – №2(54) – с.100 –109

Толкачева Татьяна Алексеевна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники.
Пузиков Никита Александрович, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, nikitnikpls@gmail.com.
Научный руководитель: Пацеева Анастасия Георгиевна, доцент кафедры гуманитарных дисциплин, кандидат социологических наук, anastasiapaceeva@yandex.ru.

ЭВОЛЮЦИЯ ФОРМЫ ПРАВЛЕНИЯ В РЕСПУБЛИКЕ БЕЛАРУСЬ

В докладе сделана попытка обосновать наличие в Республике Беларусь именно смешанной республики

Республика Беларусь является смешанной республикой и с момента провозглашения своего суверенитета в 1991 г. апробировала все три разновидности республиканской формы правления. С августа 1991 г. по 15 марта 1994 г. она являлась парламентарной республикой. Ее законодательным органом являлся Верховный Совет, который был избран прямым народным голосованием. Коллективным главой государства Республики Беларусь в тот период являлся Президиум Верховного Совета, который избирался из числа депутатов. Верховный Совет утверждал Правительство – Совет Министров Республики Беларусь во главе с его Председателем. С 15 марта 1994 г. по 24 ноября 1996 г. Беларусь являлась президентской республикой. Согласно Конституции в редакции 1994 г. Президент Республики Беларусь являлся главой государства и исполнительной власти. Он избирался непосредственно народом и сам создавал Кабинет Министров как свой рабочий орган. Парламент – Верховный Совет – не вправе был отправить в отставку Кабинет Министров, а Президент – распустить Парламент. С 24 ноября 1996 г. Беларусь по форме правления является смешанной, или полупрезидентской республикой. Конституционная реформа 1996 г. выразилась в установлении сбалансированных полномочий ветвей власти, возложении на Президента страны функций Главы государства, учреждении двухпалатного Парламента, повышении роли Правительства – Совета Министров в системе ветвей власти. Совет Министров подотчетен Президенту и ответствен перед Национальным собранием. Власть Президента существенно ограничена со стороны парламента и судебной власти, что выражается в следующем:

– законодательная власть принадлежит только парламенту, и он может не только отклонить любой предложенный Президентом законопроект, но и преодолеть президентское вето квалифицированным большинством голосов (2/3 состава парламента);

– Президент не имеет право роспуска парламента за исключением случаев оговоренных в Конституции, где сказано, что Президент Республики Беларусь вправе досрочно прекратить полномочия Палаты представителей при отказе в доверии к Правительству, выражении вотума недоверия Правительству либо двукратном отказе в даче согласия на назначение Премьер-министра. Он вправе досрочно прекратить полномочия Палаты представителей либо Совета Республики на основании заключения Конституционного Суда в случае систематического или грубого нарушения палатами Парламента Конституции Республики Беларусь. В свою очередь, Палата представителей вправе выразить вотум недоверия Правительству. Парламент имеет право на импичмент, т.е. отстранение Президента от власти через суд в случае нарушения им Конституции и законов страны. Таким образом, власть Президента в Республике Беларусь ограничивается решениями Национального собрания, а Правительство имеет двойное подчинение перед Президентом и Парламентом, что являясь главным признаком смешанной республики.

Список литературы

1. Беларусь – смешанная республика [Электронный ресурс]. – Режим доступа: <https://infopedia.su/5x1806.html>. – Дата доступа: 26.03.2020.

Радион София Александровна, студентка 1 курса факультета информационных технологий и управления БГУИР, philosophysar@gmail.com
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by

БОРЬБА СОВЕТСКОГО НАРОДА С ФАШИСТСКИМИ ЗАХВАТЧИКАМИ НА ТЕРРИТОРИИ СВИСЛОЧСКОГО РАЙОНА

В тезисах рассматривается боевая деятельность партизан и подпольщиков Свислочского района в годы Великой Отечественной войны.

К началу сентября 1941 г. территория Беларуси была оккупирована немецкими войсками. Был установлен жесткий оккупационный режим. Уже на второй день войны начала активно развиваться антифашистская деятельность. Жителями деревень Собольки и Кукличы, бывшими членами КПЗБ был создан антифашистский комитет, руководство которого доверили В.В. Янушко. Также была создана партизанская группа из 10 человек во главе с Н.Н. Болтриком. К осени 1941 г. группа уже насчитывала 25 человек. Первой крупной диверсионной операцией группы является, подрыв железнодорожного состава с живой силой и техникой врага в канун Октябрьских праздников 1941 г. на участке дороги Свислочь – Гайновка. Вторым организатором партизанского движения был Н.Н. Русак. Он сражался в диверсионной группе «Стойкие», которая действовала на территории Волковысского, Свислочского и бывшего Порозовского районов. 12 февраля 1944 г. группа «Стойкие» на маленьком хуторе по пути к станции Андреевичи была окружена немецкими войсками. Партизанам пришлось вступить в открытый бой. В тот день погибли Николай Никитович Русак, Николай Иванов, Иван Жаворонок, Григорий Манжура, Владимир Шелест, Иван Порозовнюк. Особую активность партизаны проявили в период рельсовой войны. Бригада имени Чапаева смогли уничтожить 46 составов и 20 мостов. С первых дней оккупации начали появляться подпольные организации. В начале 1942 года в Задворенцах была создана антифашистская группа, которую возглавил Василий Ан-

тонович Усевич. В состав организации в основном входила молодёжь. Группа собирала оружие и прятала его, уничтожала телефонную связь. В скором времени они смогли установить связь с партизанами. Они передавали собранное оружие, а также участвовали в операциях. На базе антифашистской группы создаётся подпольная комсомольская организация. Секретарём её был избран 18-летний Володя Усевич. Комсомольские организации участвовали в подрыве железки и в спуске вражеских эшелонов. В период «рельсовая война», задворенские подпольщики приняли в ней самое активное участие. В тоже время подобная группа появилась и в д. Доброволя. Однако им не удалось развернуть активную подпольную деятельность. Патриоты были схвачены и в 1941 г. расстреляны.

1 сентября 1943 г. в партизанской бригаде «Советская Белоруссия» приступил к работе Порозовский подпольный райком ЛКСМБ. Секретарь - Андрей Фёдорович Шиманович, заместитель - Иван Егорович Аксёнов. Осенью стали появляться комсомольские ячейки в деревнях. Выполнить то или иное задание поручалось комсомольцам одной-двух ячеек.

23 октября 1943 года создаётся Свислочский подпольный райком КП(б)Б. Секретарём райкома партии был утверждён Александр Гаврилович Шламанов. 5 ноября 1943 г. приступил к работе Свислочский райком комсомола. Широкомасштабная борьба с захватчиками, которая развернулась на оккупированных врагом территориях СССР, в том числе и Беларуси, явилась одним из факторов победы СССР в войне с фашистской Германией и ее союзниками.

*Радион София Александровна, студентка 1 курса факультета информационных технологий и управления БГУИР, philosophysar@gmail.com
Научный руководитель: Мякинська Анна Владимировна, преподаватель кафедры гуманитарных дисциплин БГУИР mav@bsuir.by*

ПАМЯТНИКИ ЗАРУБИНЕЦКОЙ АРХЕОЛОГИЧЕСКОЙ КУЛЬТУРЫ НА ТЕРРИТОРИИ БЕЛАРУСИ

Доклад посвящен рассмотрению вопросов происхождения и истории зарубинецкой культуры, носители которой проживали на территории Беларуси в железном веке.

Археологи, ведя в 1899 г. раскопки около села Зарубинцы, нашли кремнированные останки и выделили признаки культуры, названной по местоположению зарубинецкой. Одни ученые считают ее праславянской, другие германской. Датируются находки периодом с II века до н.э. по, предположительно, II век н.э. На данный момент существует много версий относительно генезиса зарубинецкой культуры. По современным историческим данным, она сложилась как синтез нескольких культур: скифов-пахарей, живших в лесной зоне, милоградской, жившей довольно рассеянно на этой территории и пришедших под влиянием экспансии кельтов с Повисления племен культуры подклевешевых погребений и поморской культуры. Зарубинцы были в основном земледельцами. Пахали деревянными орудиями с помощью лошадей или волов. Сеяли ячмень, просо, пшеницу мягких сортов и карликовую. Дополняли пищу репа, лен и конопля. В северных областях зарубинецкая культура характеризовалась подсечным земледелием. Зарубинцы разводили коров, коз, овец, свиней, лошадей, реже волов. Кости собак, найденные на поселениях, позволяют предположить их помощь в охоте на пушного зверя. Охота велась на лося, оленя, кабана, зубра, реже – медведя и косулю. Рыболовство и собирательство не имели больших объемов, хотя использовались как подсобные промыслы. Железной рудой обеспечивали болотистые участки. Качество железа было плохое, кричное. Позже специализация привела к возникновению поселков типа Лютежского. Там руду обогащали и выплавляли в горнах. Особенностью керамики стали чернолощенные керамические изделия. Много было бытовых сосудов типа корчаг. Часто встречаются глиняные сковородки.

Зарубинецкая культура называется культурой урновых погребений. Поле, где неглубоко были зарыты останки, соседствовало с поселением. Все могильники бескурганые. Овальные ямы заполнены очищенными от обрядового сожжения костями, обычно помещенными в урну. Но встречаются и труположения. Инвентарь, найденный при погребениях, составляет керамическая посуда, предметы личного потребления и реже – серпы и топоры, глиняные пряслица. В разных регионах поселения имели неодинаковые жилища. От полуземлянок, которые были заглублены на метр, до наземных построек с заглубленным на полметра полом с плетеными и обмазанными глиной стенами. Очаг служил обогревом и давал возможность готовить еду. Раскопки поселения Хотомель многое открыли в вопросах быта зарубинцев. Были раскопаны городище, примыкающее к нему селище и могильник. По сути, это был дружинный поселок, укрепленный искусственными валами и обрамленный заболоченными лугами и рекой. Налицо возникновение классовой структуры с жилищными дружинниками и трудящимися селища. Таким образом, зарубинецкая археологическая культура являлась переходной от времени скифов до славянской киевской культуры.

Список литературы

1. Зарубинецкая культура и поздnezарубинецкие достопримечательности [Электронный ресурс]. – Режим доступа: <https://studbooks.net/11529/kulturologiya> – Дата доступа: 09.03.2020. item Зарубинецкая культура: происхождение, история, интересные факты [Электронный ресурс]. – Режим доступа: <https://fb.ru/article/310841/zarubineckaya-kultura-proishozhdenie-istoriya-i-interesnye-faktyi>. – Дата доступа: 09.03.2020.

Тавлуй Екатерина Владимировна, студентка 1 курса факультета информационных технологий и управления БГУИР, kate.tvl@mail.ru
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeve@bsuir.by

БОРЬБА С ПРОСТИТУЦИЕЙ В ГОРОДАХ БЕЛАРУСИ В XIX – НАЧАЛЕ XX В

В тексте рассматриваются отношения властей Российской империи к проблеме проституции и меры борьбы с ней в городах Беларуси.

Проституция на территории Российской империи была официально запрещена, но продолжала существовать. Правительство, смирившись с тем, что запретить это явление невозможно, решило взять его под контроль. Таким образом, в конце первой половины XIX века вышел ряд постановлений, регламентирующих проституцию. По новым правилам, все женщины, занимающиеся проституцией, а также дома свиданий строго контролировались полицией. Но четких правил контроля прописано не было, поэтому этот вопрос зависел только от личного решения чиновника. Налоги с проституток и домов терпимости не взимались – они были признаны «недостойными правительства». Куда интереснее была организация борделей и домов свиданий. Для их открытия требовалось специальное разрешение, а хозяйка обязательно должна была проживать в заведении. Кроме того, до 1903 г. им нельзя было выходить замуж. Портрет типичной хозяйки дома свиданий в Беларуси – еврейка 35-45 лет мещанского сословия. Бордели в среднем существовали от одного до 10 лет. Надзор за проституцией в различных городах Беларуси был организован по-разному. В Витебске был учреждён врачебно-полицейский комитет при врачебном отделении, как и во многих городах Российской империи. В Пинске, Могилёве, Лиде также существовали врачебно-полицейские комитеты, но не в полном составе. В Минске и Бобруйске надзором за проституцией занимались городские санитарные комитеты. При некоторых санитарных комитетах состояли специальные агенты-сыщики, которые выслеживали женщин, занимающихся проституцией тайно.

По официальным сведениям, в 1889 году в Минской, Гродненской, Витебской и Могилёвской губернии 624 женщины занимались проституцией. Но куртизанок было гораздо больше. Проститутки искали клиентов и оказывали свои услуги практически везде: в трактирах, районах квартир и борделей, банях. Большая часть проституток принадлежала к мещанскому сословию, по вероисповеданию они делились на три практические равные части с небольшим преимуществом в сторону иудаизма. Практически все проститутки были безграмотные и незамужние. Таким образом, проституция в Беларуси в XIX веке была похожа на проституцию во всей Российской империи. Права проституток ущемлялись законодательством. Методы контроля не были эффективными. Минск на фоне других городов Российской империи занимал одну из самых прогрессивных позиций по надзору за проституцией.

Список литературы

1. Воронич, Т. В. Административно-полицейское регулирование проституции в городах Беларуси: вторая половина XIX – начало XX в./ Т. В. Воронич// Журнал Белорусского государственного университета. История. – 2019. – №1. – С. 46 –56.
2. Воронич, Т. В. Город и проституция: из истории повседневности Гродно (вторая половина XIX – начало XX вв.)/ Т. В. Воронич. – Минск: Издательство «Зміцер Колас», 2012. – С. 260 –290.
3. Воронич, Т. В. Царские власти в борьбе за моральность минчан: вторая половина XIX – начало XX вв./ Т. В. Воронич. – Минск: Беларуская навука, 2012. – С. 305 –322.

Федорович Дарья Андреевна, студентка 1 курса факультета информационных технологий и управления БГУИР, lvwvndv@gmail.com
Хусаинов Денис Олегович, студент 1 курса факультета информационных технологий и управления БГУИР, khusainov02@inkloud.com
Научный руководитель: Николаева Людмила Викторовна, заведующий кафедрой гуманитарных дисциплин БГУИР, кандидат исторических наук, доцент, Mikalayeva@bsuir.by

ДИСКРИМИНАЦИЯ ПО ГЕНДЕРНОМУ ПРИЗНАКУ В ИТ-ИНДУСТРИИ

Материал посвящён проблеме гендерного неравенства в ИТ-индустрии.

В 2017 году компания Google уволила своего сотрудника за скандальную статью, в которой он ставил под сомнение усилия компании в области разнообразия и утверждал, что небольшое число женщин на технических должностях является результатом биологических различий, а не дискриминации. В то же время выходит множество опросов и исследований, в которых утверждается о том, что женщины в технологических компаниях подвергаются дискриминации. Например, в 2019 году приложение «Blind» выпускает отчет о сексизме в техноиндустрии. Опрос показывал, что 37% подверглись сексизму или стали его свидетелями на работе. Респондентами были сотрудники различных компаний: Microsoft, Intel, Adobe и т. д. Для того, чтобы шире взглянуть на проблему, привлечем статистические данные, представленные Amazon, Google и т.д. Согласно их отчетам, доля женщин на «инженерных» рабочих местах находится в пределах 20-25%. Вместе с тем данные о количестве девушек-студенток, обучающихся по ИТ-специальностям опровергают тезис о дискриминации: если в начале 90-х гг. XX в. доля студенток равнялась 30%, то в наши дни – чуть менее 20% всех студентов. Очевидно, что во многих случаях академические интересы мужчин и женщин значительно различаются, что отражается в долях в высших учебных заведениях. Некоторые из таких различий развиваются и изменяются со временем, а другие остаются стабильными с течением времени, включая преобладание женщин в медицинских профессиях, государственном управлении, образовании, психологии, коммуникациях и языках. Рассматривая причины этой тенденции, обратимся к исследованиям скандинавских стран, известных как «Скандинавский парадокс»: страны

с более высоким уровнем гендерного равенства, как правило, имеют худший гендерный баланс в таких областях как точные науки и инженерия. Причинами можно назвать разную степень заинтересованности в зависимости от гендера. Похожие выводы делают в статье «Недостаток способностей, а выбора» 2013 года, однако можно предположить, что женщины не их математические способности, а их личный выбор. Исходя из рассмотренных данных можно сделать вывод, что так такового кризиса гендерного неравенства по отношению к женскому полу в ИТ-индустрии не наблюдается. Причиной процентного гендерного неравенства по большей части является лишь то, что девушкам, в связи со множеством факторов, свойственно делать выбор в пользу нетехнических профессий. Поэтому в ближайшие несколько лет нельзя ожидать резкого роста доли женщин в крупных технологических компаниях, но при этом и нельзя сказать, что это является острой социокультурной проблемой. Стоит отметить, что проблемы, касающиеся неравенства, являются постоянно меняющимися и подлежащими обсуждению, что оставляет открытым возможности для дискуссий и новых движений.

Список литературы

1. National Center for Education Statistics [Electronic resource] – Mode of access: <https://nces.ed.gov/> – Date of access: 19.03.2020.
2. Brunila, K. The famous nordic gender equality and what's nordic about it: Gender equality in finnish and swedish education / K. Brunila // Nordic Studies in Education. – 2013. – Vol. 33(4) – P. 300-313.
3. Stoet, G., Geary, D. The Gender –Equality Paradox in Science, Technology, Engineering, and Mathematics Education / G. Stoet, D. Geary // Psychological Science. – 2018. – Vol. 29(4). – P. 581 –593.

Шведов Андрей Робертович, студент 2 курса факультета информационных технологий и управления БГУИР.

Научный руководитель: Киселёв Александр Александрович, кандидат исторических наук, доцент кафедры гуманитарных дисциплин БГУИР kiselev@bsuir.by

Содержание

Секция "Интеллектуальные информационные технологии" 5

А. В. Бобков

Динамическая генерация графического представления элементов диаграмм с использованием семантических сетей 6

Д. С. Бруцкий

Алгоритмы поиска полностью представленных фрагментов знаний для модели унифицированного семантического представления знаний 7

М. С. Веренич

Методика оценки пользовательского интерфейса по отдельным направлениям 8

А. Е. Винокур

Способ взаимодействия субъектов экономики внутри цифровой финансовой экосистемы на основе платежных систем 9

А. В. Демешко

Возможности информационных технологий в сфере предоставления информации о транспортных средствах 10

Е. А. Дюбина

Модель автоматизированной верификации компонентов машин обработки знаний 11

И. С. Ермаченок

Структурная архитектура приложений для ориентации на местности 12

А. А. Жук

Разделение данных, как универсальный метод оптимизации приложений, работающих с большими данными 13

К. П. Корольков

Автоматизированная система проведения футбольных турниров для мобильных платформ 14

П. Н. Лось

Использование экспертных систем для анализа и оценки информационной безопасности 15

П. Н. Лось

Построение модели базы знаний для экспертной системы аудита безопасности 16

А. Э. Мелкумов

Веб-система информационного обеспечения предприятия 17

А. А. Пинчук

Средства прогнозирования загруженности парковок 18

М. О. Стельмачёнок

Способы сбора данных пользователей соцсетей 19

В. А. Тарасенко

Проектирование библиотеки визуализации графовых структур на основе требований пользователя 20

В. В. Трунц

Средства автоматизации разработки многократно используемых компонентов решателей задач 21

А. С. Хоменко

Анализ показателей применяемых для оценки алгоритмов сегментации изображений 22

В. Б. Шабатько

Исследование параметров интернет-магазина, влияющих на его эффективность 23

И. М. Шаплыко	
Требования к модели обучаемого в адаптивных обучающих системах	24
П. В. Юхневич	
Использование алгоритмов машинного обучения для преобразования поискового запроса на естественном языке	25
И. Д. Якимцов	
Средства автоматической обработки текстовых обращений в службу поддержки	26
Секция "Системы управления"	27
Я. С. Архипенко, А. Е. Парамонова	
Применение кодов «Data Matrix» для маркировки продукции. Система контроля качества печати	28
А. В. Бортник	
Алгоритмизация информационной системы с различной архитектурой	30
А. И. Беленькая	
Методы построения информационных диалоговых систем	32
О. А. Буяльский	
Распознавание образов с использованием сиамской нейронной сети	33
В. О. Быков, Я. Ю. Томашевич	
Система наведения лазерного луча	34
К. М. Локтевич	
Совместимые системы документооборота	36
А. А. Малашенков	
Zigbee/IEEE 802.15.4 как вариант сетевого шлюза датчика-индикатора состояния электрической сети и потребителя электрической энергии	38
А. Ю. Петрович	
Алгоритм автоматического определения темпа речи	40
И. А. Романюк	
Системы управления энергопотреблением жилых домов	41
С. В. Снисаренко	
Динамическое планирование траектории коллаборативного робота.....	42
Д. А. Струц	
Имитационное моделирование системы управления группой лифтов.....	44
Е. В. Тарасюк	
Системы видеонаблюдения с применением тепловизионных технологий	45
Д. С. Трофимов	
Протокол DHCP для автоматической настройки сетевых устройств	47
Секция "Автоматизированные системы обработки информации"	49
А. В. Азаренко	
Методы статического анализа программного кода	50
В. В. Азарко	
Сравнение алгоритмов векторизация текста в системах классификации	52
Т. А. Арутюнова	
Методы для распознавания жестов	53

Р. И. Базыльчик	
Обзор и анализ автоматизированных систем учета материальных ценностей учреждения высшего образования	54
В. И. Бартош	
Методы распознавания типов статей	55
Т. С. Боброва, В. И. Ярмолик	
Применение акселерометрического метода для ранней диагностики заболеваний центральной нервной системы	56
М. В. Бойко	
Достаточность 10 принципов OWASP для написания безопасного кода	58
А. В. Бортник	
Алгоритмизация информационной системы с различной архитектурой	61
А. Н. Василевский	
Визуализация данных в виртуальной реальности	63
Е. С. Высоцкий	
Веб-технологии для Решения логистических задач железнодорожных грузоперевозок	64
Е. А. Воробьев	
Распределенный сервис сертификации на блокчейн платформе Ethereum	65
И. П. Геврасёва	
Подходы к построению хранилищ данных	67
Е. С. Долгая	
сар hana как платформа для построения корпоративного хранилища данных	68
Р. С. Герилевич	
Алгоритм автоматизированной системы управления медицинским персоналом для учреждений здравоохранения	69
И. А. Гончарик	
Обработка результатов автоматизированного тестирования	71
О. Д. Гобрин	
Сравнительный обзор технологий анализа данных в параллельных системах	73
Р. Ю. Готовко	
Применение оптического распознавания символов в документообороте	75
Р. Р. Глушень	
Исследование частых ошибок при хранении данных в мобильных приложениях	76
А. В. Гурман	
Анонимность ради безопасности при использовании анонимайзеров	77
С. В. Дробышев	
Исследование проблем и методов управления процессами лесовосстановления	78
К. Р. Емчик	
Алгоритм автоматизации составления заказа	79
И. Г. Завистович, Д. Н. Чернаштан	
Технология генеративных нейронных сетей в преобразовании изображений	80
А. Ю. Зенович-Лешкевич-Ольпинский	
Веб-приложение видеомониторинга	81
В. А. Зинченко	
Методы и алгоритмы формирования персонала компании	82

С. Н. Ивановский, Г. С. Климов Компьютерное моделирование оптимизационных задач	83
Я. Г. Карчевская Web-приложение туристического агентства	84
Е. Г. Клебан Исследование методов обработки изображений на мобильных устройствах	86
А. Г. Климович Методология двумерной визуализации расписания учебного процесса	87
М. С. Короткевич Алгоритмы управления бизнес-процессами в биллинговых системах телекоммуникационных компаний	88
В. С. Кобяк Использование нейронной сети для обучения игрового персонажа	89
Е. В. Кривальцевич Неинвазивные нейрокомпьютерные системы как технология реабилитации	90
А. С. Кривда Метод принятия решений на основе техники интерполяции и кластеризации	91
В. А. Кулыба Обзор Deepfake технологий и решений в области компьютерного зрения	92
Е. М. Ланин Система оперативного планирования медицинского центра	94
А. И. Лапо, В. И. Ярмолик, М. С. Навресь Математическое моделирование лазерной размерной обработки	95
Д. О. Логвинов Мониторинг животных	97
А. Е. Любанец Оптимизация микросервисной архитектуры проложения	98
А. О. Мехедко Методика и средства автоматизированного тестирования интерфейса программного приложения	99
Д. А. Минько Алгоритм улучшения качества медиафайлов	100
А. В. Осецкая Определение совокупности метрик для расчета эффективности работы программных продуктов	101
А. С. Пархамович Использование технологий искусственного интеллекта для голосового взаимодействия с клиентами в контакт-центрах	102
В. С. Потапчик Автоматизированное тестирование игр	103
В. С. Родионов Алгоритмы распознавания образов на основе нейронной сети	104
Е. А. Романович Модификации алгоритма RAFT в распределенных системах	105
О. В. Савик Алгоритмы для анализа данных днк	107

Р. Я. Слука	
Методы оценки гибкости архитектуры приложений	108
А. В. Сочивко	
Кросс-валидация финансовых моделей	109
Д. Н. Ставер	
Принятие решений в задачах логического вывода	111
Staradubets Andrei	
Comparative analysis of machine and deep learning algorithms in semantic analysis	112
М. Д. Тараскевич	
Динамическое перераспределение задач в системах взаимодействующих сервисов	113
В. В. Царева	
Анализ методов нахождения скрытого представления медиафайлов	114
А. И. Циманович	
Проблемы современных веб-приложений на мобильных устройствах	115
А. В. Чопик	
Современное состояние методов обнаружения и распознавания лиц	117
Е. М. Шаюнов	
Системный анализ серверов для выявления аномальных и аварийных состояний	118
Д. А. Шакунов	
Инструментальные методы анализа климатических изменений	119
В. С. Шекунов	
Применение метода роя частиц для калибровки высокоскоростных АЦП	120
А. И. Шостыр	
Анализ эффективности инвестиционной деятельности	121
Э. А. Шихмурадов	
Разработка системы обработки задач организации	122
О. И. Юхневич	
Алгоритмы управления распределёнными программно-определяемыми сетями	123
А. В. Якубович	
Системный анализ городской среды посредством мониторинга социальной сети Twitter	125
Секция "Вычислительные методы и программирование"	127
Ю. Д. Аксак, Е. В. Змитрович, Я. Г. Гриневич	
Полезные плагины для Unity	128
П. В. Борисевич, Д. В. Коршикова	
Ошибки определения целевой аудитории в играх	129
Д. Е. Вельков	
Оценка рисков в бизнесе	130
А. В. Волощик, Я. Г. Гриневич	
Оптимизация кода JIT-компилятором	132
Е. А. Власова, Д. В. Выдрук	
Особенности создания игровых моделей	133
В. В. Гавриков, Е. А. Пукало, А. К. Филипчик, Д. В. Коршикова	
Взаимосвязь игровой индустрии со сферами нашей жизни	134

А. Н. Дюсов, Е. Н. Синяк, Д. В. Коршикова Методы моделирования	135
Е. В. Доброгост, А. А. Жуковец, Я. Г. Гриневич Практическая реализация генерации подземелий на основе процедурно сгенерированных узлов графа	136
К. Н. Курко, В. Д. Кресс Разработка основных механик мобильной стратегии реального времени на Unity	137
К. Н. Курко Разработка мобильной игры «Мемогу»	138
Н. Ю. Кушлеев, Н. М. Киселёва Окружение в играх	139
Е. А. Метла, П. В. Чирак, Т. В. Олецкая, Е. А. Ризмакова Использование игр в качестве интерактивного привлечения абитуриентов	140
Я. Д. Михайлов М. П. Анищенко Шлем дополненной виртуальности	141
Д. С. Панкевич Self-modifying code	142
А. С. Позняк, Д. Д. Кайдалова Теория вероятностей в азартных играх. Разработка симулятора парадокса Монти Холла	144
В. О. Печурихин, Я. Г. Гриневич Использование Unity при создании игры в жанре пошаговой стратегии	145
В. В. Плотников, В. Д. Кресс Шифрование Хаффманом	146
Т. А. Рак Реалистичный рендеринг на основе метода трассировки лучей	147
О. Д. Филистович, А. Ю. Рогач Теория вероятности в контексте основ игрового баланса. Вероятность и случайность	149
Ю. В. Сильванович Кривая 4-го порядка лемниската Бернулли	150
А. В. Стаховский Особенности динамической симуляции огня	151
А. Е. Тымуль, В. В. Мельникова Жанровая классификация видеоигр	152
О. Д. Филистович, К. Н. Курко Разработка мобильной игры с использованием механики 2048 «Малышка, тебе крышка»	153
Я. И. Фролов, М. В. Лемеза Генерация паролей в банковском деле	154
А. Э. Хобта, Е. В. Доброгост, Я. Г. Гриневич Разработка инди-игры на движке Unity 3D	158
Д. С. Шек, Д. С. Папакуль Анализ способов оптимизации отрисовки кадров в реальном времени	159
П. И. Жулковский А. Е. Ярошевский Реализация алгоритмов Паласиоса в аркадном проекте	160
Секция "Электрические цепи"	161

А. Д. Бондарчук, Д. С. Жуковский Анализ соединительных UTP-кабелей	162
Д. В. Дегтярик системы впрыска топлива бензиновых двигателей	163
А. А. Карташов Оптимизация энергопотребления микроконтроллеров на основе risc-v архитектуры	164
Д. А. Кенть Рассматривается архитектура и проблемы проектирования систем на кристалле	165
В. А. Куль Система оплаты проезда в наземном общественном транспорте банковскими бесконтактными картами	166
В. А. Островский Мобильное приложение для управления умным домом на основе платформы Android	167
А. В. Полулех Автоматизированная система пожаротушения склада	168
А. О. Савинков Система анализа выбросов отработавших газов в автомобиле	169
О. В. Семченко Микропроцессорная система охранной сигнализации	170
А. А. Серый Автоматизированная система трёхзонного климат-контроля грузового тягача	171
Е. Ю. Страковский Цифровой тахометр для легкового автомобиля	172
И. В. Шаблинский Стабилизатор тока вольтметра универсального класса точности 0,01	173
А. А. Шайпак Автоматизированная система контроля микроклимата в тепличном комбинате	174
Секция "Гуманитарные дисциплины"	175
А. А. Автухович Зимняя война: новые подходы	176
Н. И. Акулов Оршанская битва 1514 г. как пример военного искусства ВКЛ	178
Д. М. Баранов Социально-экономическое положение Западной Беларуси в составе Польши	179
П. В. Борисевич Влияние аварии на черновольской АЭС на социально-экономическую ситуацию в Беларуси (1986 – 1990-е гг.)	180
А. С. Войтешик Коллаборационизм на территории Беларуси в годы Великой Отечественной войны	181
А. Д. Воробьёва История моей семьи сквозь призму истории XX века	182
Д. В. Горнак Саломея Регина Русецкая: жизненный путь и деятельность	183

В. Ю. Железовский	
Моя малая родина – г. Березовка	184
Е. В. Жук М. А. Гулида А. С. Иванова Т. О. Бондарева	
Образ IT-специалиста в представлениях студентов	185
А. Г. Кірылушкін У. А. Мінееў	
Вацлаў Ластоўскі: жыццёвы шлях і дзейнасць	187
В. В. Ягода	
«Рыскун» и «Шпег» в истории разведывательной службы ВКЛ	189
О. Д. Липинская	
Гісторыя пчалы ў гісторыі маёй краіны	190
А. А. Ярошык	
Гераічная абарона Брэсцкай крэпасці летам 1941 г.	191
А. Ю. Никитина	
Белорусы во французском движении сопротивления. История Кожемякина Ф. Ф.	192
М. С. Подлешук	
Рижский мирный договор	193
С. А. Потапова	
Феминизм в белорусском обществе: историческая ретроспектива	194
Н. А. Пузиков Т. А. Толкачёва	
Восприятие проблемы справедливости глазами студентов.	195
С. А. Радион	
Эволюция формы правления в Республике Беларусь	196
С. А. Радион	
Борьба советского народа с фашистскими захватчиками на территории Свислочского района .	197
Е. В. Тавлуй	
Памятники зарубинецкой археологической культуры на территории Беларуси	198
Д. А. Федорович Д. О. Хусаинов	
Борьба с проституцией в городах Беларуси в XIX – начале XX в	199
А. Р. Шведов	
Дискриминация по гендерному признаку в IT-индустрии	200

Научное издание

**56-я НАУЧНАЯ КОНФЕРЕНЦИЯ АСПИРАНТОВ, МАГИСТРАНТОВ
И СТУДЕНТОВ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»**

(МИНСК, 21–24 АПРЕЛЯ 2020 ГОДА)

МАТЕРИАЛЫ КОНФЕРЕНЦИИ

по направлению 2:

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И УПРАВЛЕНИЕ

В авторской редакции
Ответственный за выпуск *А. Б. Гуринович*
Компьютерная верстка *Я. Г. Гриневич, А. Ф. Трофимович*
Дизайн обложки *А. А. Навроцкий*

Подписано в печать 25.05.2020. Формат 60×84 ¹/₈. Бумага офсетная.
Гарнитура «Computer Modern». Отпечатано на ризографе. Усл. печ. л. 24,65.
Уч.-изд. л. 22,7. Тираж 25 экз. Заказ 79.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014, №2/113 от 07.04.2014,
№3/615 от 07.04.2014. ЛП №02330/264 от 14.04.2014.
220013, Минск, П. Бровки, 6