

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

Институт информационных технологий

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

(Минск, 21 апреля 2018 года)

**ТЕЗИСЫ ДОКЛАДОВ
по направлению 8:**

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Минск БГУИР 2018

УДК 004(082)
ББК 32.971
П99

Редакционная коллегия:
В. И. Пачинин, А. А. Охрименко

П99 **54-я** научная конференция аспирантов, магистрантов и студентов учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» : тез. докл. по направлению 8: Информационные системы и технологии / редкол. : В. И. Пачинин, А. А. Охрименко. – Минск : БГУИР, 2018. – 115 с. : ил.
ISBN 978-985-543-409-3.

Сборник включает тезисы докладов 54-й научной конференции аспирантов, магистрантов и студентов БГУИР, которые посвящены предметам, изучаемым студентами, а также результатам их научных исследований и разработок. Все тезисы в сборнике представлены в редакции авторов.

Адресуется студентам БГУИР всех специальностей, преподавателям и широкому кругу читателей.

УДК 004(082)
ББК 32.971

ISBN 978-985-543-409-3

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2018

ОГЛАВЛЕНИЕ

ФИЗИКО-МАТЕМАТИЧЕСКИЕ И ЭКОНОМИЧЕСКИЕ ДИСЦИПЛИНЫ	5
<i>Буянов А.В.</i> АВТОМАТИЗАЦИЯ РАБОТЫ ОТДЕЛА ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ ..	5
<i>Герасюк В.А.</i> ВЛИЯНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ НА АДАПТАЦИЮ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ В СОВРЕМЕННОМ ОБЩЕСТВЕ	6
ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ	7
<i>Кабанович А.С.</i> В ГЛАВНОМ РАСЧЕТНОМ ИНФОРМАЦИОННОМ ЦЕНТРЕ БЕЛОРУССКОЙ ЖЕЛЕЗНОЙ ДОРОГИ ..	7
<i>Камагаева А.В.</i> МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ФИЛЬТРАЦИИ АНОМАЛЬНЫХ ЖИДКОСТЕЙ	8
<i>Мазур А.Д.</i> НАЧАЛЬНАЯ СИНХРОНИЗАЦИЯ ПОМЕХОУСТОЙЧИВОГО КОДА С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ РАДЕМАХЕРА	9
<i>Мартыничук М.Н.</i> ФИНАНСОВЫЕ ЗАТРАТЫ БЕЛОРУССКИХ БАНКОВ НА ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ..	10
<i>Мельченко А.О.</i> ЭФФЕКТ КАЗИМИРА И ЕГО ПРИМЕНЕНИЕ	11
<i>Предкель Д.А.</i> КОДИРОВАНИЕ СЕГМЕНТИРОВАННОГО ОБЪЕКТА	13
<i>Пунцель Е.В.</i> АКТУАЛЬНОСТЬ ЭЛЕКТРОННЫХ БИБЛИОТЕК В УЧРЕЖДЕНИЯХ СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ	14
<i>Росляков А.П., Цацура К.М.</i> ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И НЕЙРОННЫЕ СЕТИ	15
<i>Хадкевич О.В.</i> ДЕКОДИРОВАНИЕ ОРТОГОНАЛЬНОГО НЕЛИНЕЙНОГО КОДА	17
<i>Шлома К.Н.</i> ЗАЩИТА ИНФОРМАЦИИ НА ОСНОВЕ НИЗКОСКОРОСТНОГО КОДИРОВАНИЯ	18
ТЕХНИЧЕСКИЕ И ОБЩЕСТВЕННЫЕ ДИСЦИПЛИНЫ	20
<i>Алешко Н.С., Анкуда Д.И.</i> ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ УСТРОЙСТВ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ПИЛОТАМИ ВОЕННОЙ АВИАЦИИ	20
<i>Бакунович Д.В.</i> УГРОЗЫ ДЛЯ УСТРОЙСТВ, РАБОТАЮЩИХ ПОД УПРАВЛЕНИЕМ ОС ANDROID	22
<i>Беликов А.С., Агапкин Л.М., Чучвал А.Ю., Мирончик А.Н.</i> ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В ОБРАЗОВАНИИ ..	23
<i>Булахов Е.В.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ РАСПОЗНАВАНИЯ ЛИЦ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ	24
<i>Бурак Д.И.</i> СОВРЕМЕННЫЕ WEB-ПРИЛОЖЕНИЯ	26
<i>Бурак Д.Б.</i> ИНФОРМАЦИОННОЕ МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ОРГАНИЗАЦИЙ И СЛУЖБ ЖИЛИЩНО-КОММУНАЛЬНОГО ХОЗЯЙСТВА	27
<i>Венско А.В.</i> МЕТОД ИЗОФОТОМЕТРИИ ФУНКЦИИ РАССЕЙЯНИЯ ТОЧКИ	28
<i>Вербило А.И.</i> ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ПРИЛОЖЕНИЙ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ	29
<i>Воробей К.П.</i> ПРОГРАММНОЕ СРЕДСТВО НА ПЛАТФОРМЕ ANDROID ДЛЯ РЕГЛАМЕНТИРОВАНИЯ ПОСТУПАЮЩИХ ВЫЗОВОВ	30
<i>Германенко И.И., Бахур Н.И., Моженкова Е.В.</i> МОДЕРНИЗАЦИЯ ПРОХОДНОЙ КАМЕРЫ ТЕПЛА И ХОЛОДА	30
<i>Гимик В.О.</i> СИСТЕМА ИДЕНТИФИКАЦИИ FACE ID	33
<i>Гольдюк В.Н.</i> ОБРАЗ СОВРЕМЕННОГО РАЗРАБОТЧИКА ДЛЯ БИЗНЕСА	34
<i>Гончаров Я.И.</i> МОБИЛЬНОЕ ПРИЛОЖЕНИЕ «СИСТЕМА ИНФОРМИРОВАНИЯ В ЭКСТРЕННЫХ СИТУАЦИЯХ» ..	35
<i>Гулис Н.В.</i> ПАТТЕРН УПРАВЛЕНИЯ СОСТОЯНИЕМ НА ПРИМЕРЕ VUEX	36
<i>Гулис Н.В.</i> VUE.JS – АНАЛИЗ ФРЕЙМВОРКА	38
<i>Дедович Д.К.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ДЕЛЕГИРОВАНИЯ ДОСТУПА К ВЕБ-РЕСУРСАМ НА ПЛАТФОРМЕ ANDROID	39
<i>Денис Н.А.</i> ОБРАБОТКА И УЛУЧШЕНИЕ КАЧЕСТВА ИЗОБРАЖЕНИЯ С ПОМОЩЬЮ ПРЕОБРАЗОВАНИЙ ФУРЬЕ ..	40
<i>Жигуновский А.В.</i> КРОССБРАУЗЕРНЫЙ ВЕБ-СЕРВИС ДЛЯ РЕАЛИЗАЦИИ АВТОМОБИЛЬНЫХ ЗАПЧАСТЕЙ	42
<i>Житников С.В.</i> МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ КЛИЕНТОВ КОМПАНИИ «ЕДА ДАРОМ» НА ПЛАТФОРМЕ ios ..	43
<i>Заяц И.Л., Лазаренко Р.А.</i> РЕКЛАМА, КАК УГРОЗА ИНФОРМАЦИОННО-ПСИХОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ ЛИЧНОСТИ	45
<i>Казанок Д.Ю., Новохрестова А.О.</i> ЗАЩИТА ОТ АВТОМАТИЧЕСКОГО ТРАФИКА	46
<i>Кармызов А.С.</i> ВИРТУАЛИЗАЦИЯ СЕРВЕРОВ С ПОМОЩЬЮ WEBSHERE	48
<i>Кармызов А.С.</i> ПРИМЕНЕНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ В СИСТЕМАХ УМНОГО ДОМА	49
<i>Карнацкий П.Н.</i> СИСТЕМА УПРАВЛЕНИЯ ЭЛЕКТРОДВИГАТЕЛЕМ С РЕКУПЕРАЦИЕЙ	51
<i>Кожарнович В.В.</i> СИСТЕМА УПРАВЛЕНИЯ АКТИВНЫМ АНТИКРЫЛОМ СПОРТИВНОГО АВТОМОБИЛЯ	52
<i>Козырь И.Ю.</i> ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ГЕНЕРАЦИИ ТЕСТ-КЕЙСОВ МЕТОДОМ PAIRWISE	54
<i>Крень Н.С., Бакунович Д.С.</i> РИСКИ ИСПОЛЬЗОВАНИЯ ОБЛАЧНЫХ ХРАНИЛИЩ	55
<i>Кузнецов В.А., Иванов А.Ю., Матусевич А.В., Высоких В.А.</i> ПРОБЛЕМЫ БЕЗОПАСНОСТИ СОВРЕМЕННЫХ SMS	56
<i>Кузьмицкий В.И., Биркос В.А.</i> ДИАГНОСТИЧЕСКАЯ СИСТЕМА ЭЛЕКТРОНИКИ АВТОМОБИЛЯ НА ПРЕДМЕТ ОШИБОК. ИЗМЕНЕНИЕ ПАРАМЕТРОВ ВПРЫСКА И ЗАЖИГАНИЯ ТОПЛИВА	58
<i>Ламчановский А.Г.</i> АЛГОРИТМ РАБОТЫ СИСТЕМЫ УЧЕТА ЗАДОЛЖЕННОСТИ ПО БАНКОВСКИМ КРЕДИТНЫМ КАРТАМ	59
<i>Литовко Д.А.</i> РОБОТЫ В CRM БИТРИКС24. REST ДЛЯ CRM-ТРИГГЕРОВ	60
<i>Лупач В.Д.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОБУЧЕНИЯ ОСНОВАМ КРИПТОГРАФИИ	62
<i>Мультап Р.И., Мазур А.Д.</i> ОСОБЕННОСТИ КЭШИРОВАНИЯ КОМПОНЕНТОВ В UNITY	63
<i>Мультап Р.И., Мазур А.Д.</i> ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ ТЕКСТУР НА ОСНОВЕ АЛГОРИТМА DIAMOND-SQUARE ..	66
<i>Малашенко А.Ю.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ВЕДЕНИЯ УЧЕТА ГРУЗОПЕРЕВОЗОК	68
<i>Мисюра А.Ф.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ УПРАВЛЕНИЯ ЗАДАЧАМИ И ОТСЛЕЖИВАНИЯ ОШИБОК ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	69
<i>Наливко В.Н.</i> ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ФОРМИРОВАНИЯ ЗАКАЗОВ И ПРОДАЖИ СТРОИТЕЛЬНЫХ МАТЕРИАЛОВ	70
<i>Нестерович Р.В.</i> КОРПОРАТИВНЫЕ ПОРТАЛЫ НА ПРЕДПРИЯТИИ	72
<i>Пагин С.В.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ УПРАВЛЕНИЯ ЭЛЕКТРОННЫМИ АКТИВАМИ	73

Паиковский Н.О. ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ И ОЦЕНКИ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ	74
Петрович А.С. РАЗВЕРТЫВАНИЕ МИКРОСЕРВИСОВ	75
Петрович А.С., Думиков А.А. ПОСТРОЕНИЕ ЗАКРЫТОГО УЧАСТКА ИНФРАСТРУКТУРЫ С ЦЕЛЬЮ ОБЕСПЕЧЕНИЯ ПОЛЬЗОВАТЕЛЕЙ БЕЗОПАСНОЙ ТЕСТОВОЙ ВИРТУАЛЬНОЙ СРЕДОЙ	77
Поплёвка В.И. МОДЕЛИ И АЛГОРИТМЫ ОБЕСПЕЧЕНИЯ НЕОБХОДИМЫМИ РЕСУРСАМИ ДЛЯ ЖИЗНЕДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА НА ОСНОВЕ ТЕОРИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА	78
Прибыткин М.А. МИКРОПРОЦЕССОРНОЕ УСТРОЙСТВО МОНИТОРИНГА ПЕРЕМЕЩЕНИЯ АВТОМОБИЛЯ	80
Прузан А.Н. О КОМПЛЕКСНОСТИ НАУЧНЫХ ИССЛЕДОВАНИЙ В ОБЛАСТИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ	81
Пуйдак В.А. КОМПЛЕКСНАЯ ЗАЩИТА ПЕРСОНАЛЬНЫХ ДАННЫХ НА БАЗЕ ПО NET STUDIO	82
Путицкий А.И. ИГРОВОЕ ANDROID-ПРИЛОЖЕНИЕ «KNASAGI: THE LONE WANDERER»	85
Радионов М.Г. ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПРОЕКТИРОВАНИЯ САДОВЫХ ДОМОВ	86
Русак Д.С. СПРАВОЧНО-ИНФОРМАЦИОННАЯ СИСТЕМА ФИНАНСОВОГО ПОРТАЛА MYFIN.BY	87
Рыбаков Е.А. ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ONLINE БРОНИРОВАНИЯ И ТРАНСФЕРА АВТОМОБИЛЕЙ	88
Сеглюк И.А., Камоцкий Р.Г. АНАЛИТИЧЕСКАЯ СИСТЕМА ПРОГНОЗИРОВАНИЯ ОШИБОК	89
Сергиеня П.В. BLOCKCHAIN	90
Ситник М.Ю. СОСТОЯНИЕ ЗАЩИТЫ ПЕРСОНИФИЦИРОВАННЫХ МЕДИЦИНСКИХ ДАННЫХ В БЕЛАРУСИ В 2018 ГОДУ	92
Соколов В.А. Скуратович Е.С. МНОГОПОТОЧНОСТЬ В UNITY СРЕДСТВАМИ РЕАКТИВНЫХ РАСШИРЕНИЙ	94
Соломенников А.С. ПРОГРАММНОЕ СРЕДСТВО ДЛЯ КАДРОВОГО УЧЕТА В ПРОЕКТЕ «ЭЛЕКТРОННАЯ ШКОЛА»	96
Сорока А.А. ПРОГРАММНОЕ СРЕДСТВО «ИНТЕРНЕТ-КИНОТЕАТР» НА ОСНОВЕ ПИРИНГОВЫХ СЕТЕЙ	96
Фиалко Е.А. СИСТЕМА УПРАВЛЕНИЯ КЛИМАТИЧЕСКИМИ ПАРАМЕТРАМИ СУШИЛЬНОЙ КАМЕРЫ	97
Халецкий Д.М. КВАНТОВЫЙ КОМПЬЮТЕР – НОВОЕ ПОКОЛЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ	98
Халецкий Д.М. ТЕХНОЛОГИИ 3D ПЕЧАТИ. 3D ПРИНТЕРЫ И СФЕРЫ ИХ ПРИМЕНЕНИЯ	100
Харлов А.А. ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА РАБОЧЕГО ВРЕМЕНИ	101
Хваль И.Н. РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА АНАЛИЗА ЛИЦЕНЗИОННЫХ НАРУШЕНИЙ В РАЗРАБАТЫВАЕМЫХ ПРОДУКТАХ	103
Хведченя А.В. СИСТЕМА ИЗМЕРЕНИЯ ДАВЛЕНИЯ В ШИНАХ АВТОМОБИЛЯ	104
Черник Д.В. АВТОРИЗАЦИЯ НА ОСНОВЕ ПОЛИТИКИ ASP.NET CORE	105
Черный М.С. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ АДМИНИСТРАТИВНОЙ ПАНЕЛЬЮ MAGENTO ..	107
Чернявский С.В. РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ВЗАИМООТНОШЕНИЯМИ С КЛИЕНТАМИ	108
Чернянин Р.И. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ПЕРСОНАЛЬНЫМИ ФИНАНСАМИ	109
Шилкин А.О. ПРОГРАММНОЕ СРЕДСТВО ПО ОРГАНИЗАЦИИ ТРАНСЛЯЦИИ ПОТОКОВОГО ВИДЕО ПОСРЕДСТВОМ WOWZA STREAMING ENGINE	110
Шпилевский В.В. АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННОГО ПРОЦЕССА ИЗГОТОВЛЕНИЯ ЭЛЕКТРОННЫХ ИЗДЕЛИЙ	112
Ярмолович О.М., Зайкина И.С. АВТОМАТИЗИРОВАННАЯ СИСТЕМА АНАЛИЗА И УЧЁТА ИЗМЕНЕНИЙ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ	113
Яровенко К.А. ПРОВЕРКА ГИПОТЕЗЫ ДЛЯ ВЫБОРА ТЕМПЕРАТУРЫ ТЕРМОВЫДЕРЖКИ КОМПЛЕКТУЮЩИХ ЭВМ	114

ФИЗИКО-МАТЕМАТИЧЕСКИЕ И ЭКОНОМИЧЕСКИЕ ДИСЦИПЛИНЫ

АВТОМАТИЗАЦИЯ РАБОТЫ ОТДЕЛА ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Буянов А.В.

Майсеня Л.И. – зав. кафедрой ФМД, докт. пед. наук, профессор

В статье рассматривается результат внедрения разработанной системы обработки электронных заявок в деятельность отдела технической поддержки учреждения образования «Республиканский институт профессионального образования», филиал «Индустриально-педагогический колледж». Описана логика работы приложения, а также задачи, которые данная система реализует.

Функции отдела технической поддержки учреждения образования «Республиканский институт профессионального образования», филиал «Индустриально-педагогический колледж», заключаются в информационно и аппаратно-программном обеспечении персонала и учащихся учреждения, что включает в себя:

- обеспечение бесперебойного функционирования компьютерной и мультимедийной техники колледжа;
- установку и настройку специального программного обеспечения, а также разработку новых программных продуктов и последующую их поддержку;
- администрирование сайта учреждения образования;
- консультирование сотрудников и учащихся колледжа по вопросам информационных технологий;
- разработку дизайнерских шаблонов учреждения для презентационного материала;
- осуществление покупки/ремонта компьютерного и мультимедийного оборудования колледжа;
- обеспечение функционирования печатного оборудования учреждения.

Все работы должны быть документально оформлены, что в нашем случае представляется в виде заполнения журнала заявок на проведение тех или иных работ. На оформление и ведение журнала уходит некоторая часть рабочего времени, что сказывается на количественном и качественном выполнении заявок сотрудников колледжа.

С целью автоматизации данного бизнес-процесса, разработан и внедрен программный продукт для оформления и обработки заявок сотрудников колледжа, а также для создания ежемесячной отчетной документации отдела технической поддержки. Для системы заявок выделен отдельный адрес, который был выдан всем сотрудникам учреждения, а также проведена ознакомительная лекция с целью обучить сотрудников пользоваться данной системой. При переходе в данную систему, сотрудник попадает на разработанную форму авторизации (рисунок 1).

После успешной авторизации пользователь видит лишь две кнопки: одна для оформления заявки, а вторая для контроля своих еще не выполненных заявок (пример оформления новой заявки представлен на рисунке 2).

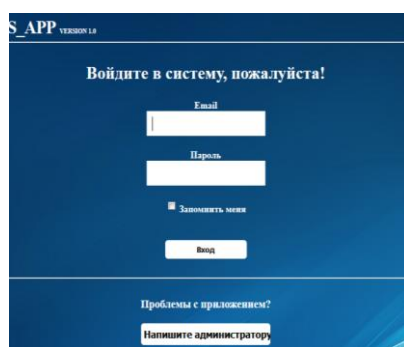


Рисунок 1 – Окно авторизации системы заявок

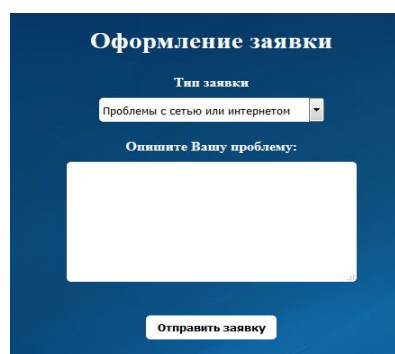


Рисунок 2 – Окно оформления новой заявки пользователем.

При оформлении заявки пользователю стоит лишь указать тип проблемы (выбор осуществляется из всевозможных проблем в колледже), а также оставить поясняющую информацию для сотрудника отдела техподдержки, чтобы он смог максимально быстро и без дополнительных вопросов выполнить свою работу. Данный алгоритм исключает излишние посещения сотрудниками колледжа отдела технической поддержки, что позволяет экономить время, которое можно потратить на выполнение необходимой работы. В свою очередь сотрудник отдела технической поддержки видит в системе список всех поступивших заявок, а также получает возможность просмотра состояния любой заявки (была ли она уже взята в работу другим сотрудником технической поддержки).

В конце каждого месяца система выводит отчет о проделанной работе отдела, а также информацию о времени, затраченном на каждую заявку от момента ее регистрации до момента ее закрытия. Данные отчеты хранятся в системе и могут быть распечатаны на бумажный носитель при необходимости, что заменяет прежний журнал регистрации заявок.

Данный программный продукт является инструментом не только для эффективной организации работы отдела технической поддержки колледжа, но и аналитическим инструментом, позволяющим вносить изменения в процесс планирования рабочего времени отдела и в распределение обязанностей между сотрудниками отдела технической поддержки.

ВЛИЯНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ НА АДАПТАЦИЮ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ В СОВРЕМЕННОМ ОБЩЕСТВЕ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Герасюк В.А

*Майсеня Л.И. – зав. кафедрой ФМД, докт. пед. наук, профессор,
Моженкова Е.В. – ст. преподаватель каф ИСиТ, м.т.н.*

В статье рассматривается разработанное программное средство чтения электронных книг для людей с ограниченными возможностями на основе устройств под управлением операционной системой iOS. Программное средство также может быть использовано в учебных учреждениях для обучения чтению учащихся с дислексией.

Получение информации для людей с ограниченными возможностями здоровья является одним из основных и неотъемлемых условий их успешной социализации, обеспечения их полноценного участия в жизни современного общества, эффективной самореализации в различных видах профессиональной и социальной деятельности. Новое поколение смартфонов позволило человеку значительно улучшить процесс создания и потребления информационных ресурсов. Современная электронная техника позволяет незрячим опосредованно получать информацию о явлениях окружающей среды, предметах и их свойствах, ориентироваться в пространстве, читать плоскочечатную литературу, распознавать цвета и определять источник света. Информационные технологии помогают также быстрее адаптироваться и развиваться людям со снижением слуха, нарушением устной речи. Они компенсируют несформированность высших зрительных, пространственных, временных, оптико-пространственных, зрительно-моторных функций.

Информационные технологии являются важными инструментами интеграции и взаимодействия лиц с нарушениями зрения или дислексией – нарушением способности к овладению навыком чтения и письма при сохранении общей способности к обучению. Исследование Департамента по делам детей, школ и семей Великобритании в рамках проекта «Пилотный проект доступных ресурсов» оценило эффективность применения компьютерных технологий в процессе обучения школьников [1]. Ноутбуки со специализированным программным обеспечением для доступа к электронным ресурсам и MP3-плееру были предоставлены 40 ученикам в возрасте 11-14 лет, и 10 педагогам-специалистам для преобразования учебников и вспомогательных материалов в формат MS Word с целью дальнейшего использования учениками с нарушениями зрения или письма. Технологии применялись в учебном заведении на протяжении года. Оценка результатов была проведена Э.А. Драффаном из Саутгемптонского университета. Для оценки использовались вопросыники, личные интервью с персоналом и учениками, а также данные в режиме онлайн, полученные из информации, собранной в ходе проекта. Результаты опроса были следующие: 71% учащихся с дислексией показали улучшение навыков чтения, 84% учеников с ослабленным зрением продемонстрировали повышение уверенности, а 58% показали улучшение в выполнении домашних заданий.

В результате упомянутого исследования выявлены основные возможности программных средств, позволяющие повысить навыки чтения и письма людей с нарушением зрения или дислексией:

- изменение размера шрифта;
- изменение цвета фона;
- изменение масштаба;
- преобразование текста в речь;
- выделение текста при прочтении;
- использование и настройка функций доступности, встроенных в операционные системы.

Концепция использования электронных файлов и технологий доступа более характерна для поддержки людей с ослабленным зрением, хотя многие из них по-прежнему используют реальные книги.

Люди с ограниченными возможностями выигрывают от использования персональных устройств и соответствующих электронных учебных материалов. Они вносят значительный вклад в развитие письма и чтения, обучения, в целом уверенности в себе и самоуважения, независимости, развитие жизненных навыков.

Следует отметить, что имеется существенное различие между электронными файлами, создаваемыми издателями для печати (PDF), и теми, которые требуются в структурированном файле, который может быть легко преобразован в другие альтернативные форматы. Образовательные продукты имеют сложные и привлекательные макеты, в то время как технологии доступа и преобразования требуют более линейного представления содержимого. Исходя из этого нами было разработано программное средство для чтения

электронных книг людьми с ограниченными возможностями на основе устройств под управлением операционной системой iOS. Программное средство предоставляет пользователям следующие возможности:

- чтение электронных книг и иных документов с аудио воспроизведением и подсветкой текста;
- доступ к списку специализированных онлайн библиотек с соответствующим материалом;
- подписки на новостные рассылки;
- добавление заметок к материалам;
- скачивание дополнительных голосов для воспроизведения книг;
- расширенные настройки отображения текста и воспроизведения аудио.

Благодаря разработанному программному средству люди с ограниченными возможностями могут намного более эффективно и удобнее заниматься процессом чтения. Программное средство также может быть использовано в учебных учреждениях для улучшения и обучения чтению учащихся с дислексией. Мобильность платформы, на которой реализовано программное средство, создает дополнительные возможности его использования.

Список использованных источников

1. DfE Accessible Resources Pilot Project Final Report [Электронный ресурс]. – 2017. – Mode of access: <http://www.altformat.org/mytextbook/index.asp?page=2>. – Дата доступа: 28.03.2018.

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ В ГЛАВНОМ РАСЧЕТНОМ ИНФОРМАЦИОННОМ ЦЕНТРЕ БЕЛОРУССКОЙ ЖЕЛЕЗНОЙ ДОРОГИ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Кабанович А.С.

*Майсеня Л.И. – зав. кафедрой ФМД, докт. пед. наук, профессор
Хмельёва А.В. – канд. техн. наук, доцент*

В статье рассмотрена актуальность внедрения разработанной системы информационного обеспечения повышения квалификации работников предприятия.

Внедрение информационных технологий во все сферы деятельности обусловлено стремительным развитием науки, качественным скачком возможностей человека, постоянно растущими объемами информации, сложностью происходящих процессов и явлений. Управление организацией любого типа, будь то крупное предприятие или мелкая фирма, связано с переработкой потока информации и принятием на ее основе оперативных и перспективных решений. Поэтому автоматизация управленческих работ является основным направлением совершенствования экономики. Применение различных методов, современных средств вычислительной техники и современных технологий связи создает новые возможности для дальнейшего совершенствования управления.

Информатизация является актуальной и для организации деятельности в сфере повышения квалификации на предприятии. Управление обучающим веб-сайтом на предприятии, его информационными потоками, документооборотом и прочими процессами представляет собой сложную систему, мелкие и крупные задачи которой тесно связаны между собой. Управленческие решения принимаются руководством на основе объективных и оперативных данных о состоянии дел в подразделениях, их показателях и данных из внешней среды.

В организационной системе наиболее трудоемкими являются процессы, связанные с обработкой информации – сбор, накопление, преобразование, отображение, хранение, передача и вывод. Ускорить эти процессы и облегчить труд персонала государственной организации позволяет Автоматизированная система управления. Человеко-машинный характер АСУ удобен тем, что благодаря этой системе осуществляется управление не только процессом обучения, но и персоналом, т.к. система создает и поддерживает деловую, творческую обстановку, обеспечивает условия для непрерывного повышения общеобразовательного, профессионального и культурного уровня работающих. В АСУ ведущая роль в выработке и принятии решений закрепляется за руководством. Программно-математические методы и вычислительная техника являются лишь средствами обработки информации и подготовки соответствующих управленческих решений.

Система повышения уровня квалификации работников должна быть гибкой, динамичной, позволять эффективно рассматривать намеченные планы, оперативно освещать все последние новости, учитывать изменения и новые тенденции. Большие возможности для совершенствования управления сайтом предоставляет использование вычислительной техники. Этими установками мы пользовались при разработке сайта для сотрудников главного расчетного информационного центра Белорусской железной дороги.

Разработанная нами система включает в себя множество разделов, в которых содержатся тесты по наиболее востребованным темам, связанным с информационными технологиями, охране труда на предприятии, а также непосредственно со спецификой работы на предприятии. После прохождения теста пользователь получает таблицу с правильными ответами. Система анализирует полученные ответы и предлагает список литературы, который поможет устранить пробелы в знаниях по определенным темам. Пользователь может заполнить также электронную форму для записи на интересующие курсы, тренинги и найти актуальную информацию.

Предложенная система позволяет оптимизировать деятельность в сфере повышения квалификации сотрудников на предприятии. Сотрудники смогут устранить пробелы в знаниях и в последующем выполнять свою работу более качественно и быстро. Разработанная система позволяет снизить затраты на перечисленные проблемы и оптимизировать процесс обучения сотрудников на предприятии.

Список использованных источников:

1. Аудит обучения персонала на предприятии [Электронный ресурс]. – Электронные данные. – Режим доступа: [http:// ppt-online.org/](http://ppt-online.org/). – Дата доступа: 23.02.2018.
2. Шекшня С.В. Управление персоналом современной организации. Учебно – практическое пособие. Издание 4– е, перераб. и доп. – М.: ЗАО « Бизнес – школа «Интел – Синтез», 2000. – 368 с. [Электронный ресурс]. – Электронные данные. – Режим доступа: [http:// hr-portal.ru/](http://hr-portal.ru/). – Дата доступа: 23.02.2018.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ФИЛЬТРАЦИИ АНОМАЛЬНЫХ ЖИДКОСТЕЙ

*Стерлитамакский филиал Башкирского государственного университета
г. Стерлитамак, Россия*

Камагаева А.В.

Хусаинова Г.Я. – доцент каф. ПИиП, к.ф.-м.н., доцент

Данная статья посвящена исследованию полей температуры аномальных жидкостей при прямолинейно-параллельной фильтрации. Получено распределение давления и температуры для линейной фильтрации.

В настоящее время, в связи с интенсификацией многих технологических процессов, потребности науки возросли настолько, что стало актуальным построение моделей, которые уточняли бы те или иные феноменологические законы, и выявляли бы внутренние механизмы и характерные особенности поведения реологически сложных систем [1-3].

С учетом этого Мирзаджанзаде А.Х. предложил феноменологическую теорию фильтрации вязкопластичной жидкости, в основе которой лежит линейная модель вязкопластичной сплошной среды.

При рассмотрении фильтрации аномальных жидкостей существенную роль играет пористая среда. Взаимодействие фильтрующейся жидкости с материалом скелета приводит к изменению реологических параметров жидкости (например, вязкости жидкости) или пористой среды. Из-за вышеперечисленных эффектов появляются фильтрационные аномалии и в этом случае речь идет уже не о фильтрации данной жидкости, а о поведении системы «жидкость - пористая среда».

Ниже рассмотрены поля температуры для прямолинейно-параллельного фильтрационного течения. В полубесконечном пласте, который заполнен нефтью, обладающей предельным градиентом сдвига, в начальный момент времени $t=0$ на границе пласта $x=x_0$ начинает работать закачивающая галерея. Рассматривается случай однофазного течения.

Уравнение энергии имеет вид:

$$c_p \frac{\partial T}{\partial t} - c_l m \eta \frac{\partial P}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}, \quad \left| \frac{\partial P}{\partial x} \right| \leq G_0,$$

$$c_p \frac{\partial T}{\partial t} + c_l \nu \left(\frac{\partial T}{\partial x} + \varepsilon \frac{\partial P}{\partial x} - \eta G_0 \frac{\frac{\partial P}{\partial x}}{\left| \frac{\partial P}{\partial x} \right|} \right) - \eta m c_l \frac{\partial P}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}, \quad \left| \frac{\partial P}{\partial x} \right| > G_0, \quad (1)$$

$$t > 0; x > x_0, T|_{t=0} = 0, \quad T|_{x=x_0} = T_0(t).$$

По заданному дебиту распределение давления находится из уравнения:

$$\nu = -\frac{k}{\mu} \left(\frac{\partial P}{\partial x} - G_0 \frac{\frac{\partial P}{\partial x}}{\left| \frac{\partial P}{\partial x} \right|} \right) = \frac{Q}{S}. \quad (2)$$

Считаем известным давление на забое

$$P(x_0) = P_0. \quad (3)$$

Из системы уравнений найдем поля давления и температуры, которые помимо чисто физического способствуют решению многих геолого-геофизических и технологических задач на стадии поисков.

Список использованных источников.

1. Нигматуллин, Р.И. Динамика многофазных сред. Ч. 1–2./ Р.И. Нигматуллин – М.: Наука, – 1987. – 359 с.
2. Бондарев Э.А., Красовицкий Б.А. Температурный режим нефтяных и газовых скважин./ Э.А., Бондарев, Б.А. Красовицкий. – Новосибирск: Наука. – 1976. – 88 с.
3. Хусаинова Г.Я. Исследование температурных полей при фильтрации аномальных жидкостей. Автореферат диссертации на соискание ученой степени кандидата физико-математических наук. / Г.Я. Хусаинова. – Уфа. – 1998. – 14 с.
4. Хусаинов, И.Г. Тепловые процессы при акустическом воздействии на насыщенную жидкостью пористую среду/ И.Г. Хусаинов // Вестник Башкирского университета. 2013. Т. 18. № 2. С. 350-353.

НАЧАЛЬНАЯ СИНХРОНИЗАЦИЯ ПОМЕХОУСТОЙЧИВОГО КОДА С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ РАДЕМАХЕРА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Мазур А.Д.

Митюхин А.И. – доцент каф. ФМД

В работе рассматривается математический алгоритм с использованием дискретных функций Радемахера. Метод позволяет для определенных применений сократить время на установление начальной синхронизации и уменьшить сложность технической реализации устройства синхронизации.

Качественные характеристики современной инфокоммуникационной системы с кодовым разделением во многом определяются надежностью и эффективностью процедуры синхронизации. Решение этой задачи значительно усложняется, когда основные параметры $[n, k, d]$ - кодов над полем $GF(q)$ должны обеспечивать высокую помехоустойчивость и малую вероятность неправильной синхронизации. Специальные системы, например, космические, военные, где применяются низкоскоростные $[n, k, d]$ -коды со значениями $n > 10^5, k \ll n, d \sim \frac{n}{2}$ должны обеспечивать малое время вхождения в синхронизм с точностью до периода n . В случае большой области n неопределенности задержек сигнала и низких отношениях сигнал/шум на входе приемника известный алгоритм шагового поиска приводит к значительным временным затратам и существенному увеличению объема оборудования. Асимптотическая сложность этого алгоритма $O(n^2)$. Для каналов с гауссовским шумом оптимальное решение задачи основывается на корреляционном подходе. Определение начальной фазы входного кодового слова сводится к нахождению максимальной коррелированности между входной последовательностью $x^j(i), i = 1, 2, \dots, n$ и множеством $\{x^M(i), M = 1, 2, \dots, q^k\}$ задержанных опорных сигналов. Точная синхронизация фазы входной последовательности с опорной определяется по максимальному значению корреляционной функции

$$\rho_{x^j(i), x^u(i)} = \frac{1}{n} \sum_{i=1}^n x^j(i) x^u(i), u \in M. \quad (1)$$

Вопросы временного поиска связаны с выбором типа низкоскоростного кода. Учет структурных особенностей кода может упростить вхождение системы в синхронизм. В [1] для решения задачи декодирования описаны свойства кода построенного с использованием функций Радемахера $rad_l(i), l = 1, 2, \dots, k$. Кодовое слово для информационной последовательности $b = b_1, b_2, \dots, b_k$ определяется мажоритарной операцией

$$x(i) = \text{Maj}[b_1 rad_1(i), b_2 rad_2(i), \dots, b_k rad_k(i)], b_l \in \{1, -1\} \quad (2)$$

над периодическими дискретными функциями Радемахера. Конструкции кода явно отражает кратность периодов функций Радемахера для разных значений $l = 1, 2, \dots, k$. В спектральном образе любого кодового слова (1) ярко выделяются регулярные спектральные компоненты, связанные с периодом каждой функции $rad_l(i)$. Эта особенность спектра, позволяет ускорить процесс поиска, если в качестве опорных сигналов использовать функции Радемахера. Время нахождения начальной фазы слова значительно уменьшается в сравнении с шаговым поиском. Можно показать, что асимптотическая сложность этого алгоритма $O(n)$.

Пример. Пусть необходимо найти начальную фазу входной последовательности $x^j(i)$ кода с параметрами: число информационных символов $k = 3$, блоковая длина $n = 8$, минимальное расстояние $d = 4$. Для вектора $\mathbf{B} = (b_1, b_2, b_3) = (1 \ 1 \ 1)$ дискретные функции Радемахера $\{rad_1(i), rad_2(i), rad_3(i)\}$ имеют вид

$$R = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{pmatrix}.$$

По формуле (2) получена порождающая матрица кода X (представлена в виде всех $M = 8$ кодовых слов).

$$X = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ & & & & \vdots & & & \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{pmatrix}.$$

Пусть на вход устройства синхронизации поступает слово $x(i) = (-1 - 1 - 1 1 1 1 - 1)$. Вычисляя (1) для $x(i)$ и $\{rad_3(i), rad_3(i+1)\}$, т.е. для четных и нечетных точек интервала неопределенности задержек, находим максимальное значение $\rho_{x(i), rad_3(i+1)}(1) = 4$. Очевидно, что область неопределенности задержек уменьшена в два раза. Далее подлежат проверке только нечетные точки $i = 1, 3, 5, 7$ области неопределенности. Полученные коэффициенты корреляция для $x(i)$ и

$$\{rad_2(i+1), rad_2(i+3), rad_2(i+5), rad_2(i+7)\}$$

имеют максимальные значения в точках: $\rho_{x(i), rad_3(i+1)}(1) = 4$ и $\rho_{x(i), rad_3(i+5)}(5) = 4$. Проверая точки 1 и 5, используя функции $rad_1(i+1)$ и $rad_1(i+5)$, находим $\rho_{x(i), rad_1(i+1)}(1) = -4$ $\rho_{x(i), rad_1(i+5)}(1) = 4$. Выбираем за начальную фазу входного кодового слова элемент $x(5)$. Реализуя в генераторе кода задержку, на пять тактовых интервалов, получаем $x^1(i-5) = x(i) = (-1 - 1 - 1 1 1 1 - 1)$ – синхронное состояние в системе. Легко установить, что для ввода систему в синхронизм потребовалось n шагов корреляции.

Список использованных источников.

1. Митюхин А. И. Корреляционные спектры и кодовые расстояния мажоритарных последовательностей/ А.И. Митюхин, П.Н. Якубенко // Доклады БГУИР. – 2015. № 4 (90). – С. 5–9.

ФИНАНСОВЫЕ ЗАТРАТЫ БЕЛОРУССКИХ БАНКОВ НА ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Мартынчук М.Н.

*Пачинин В.И. – зав. кафедрой ИСиТ, к.т.н., доцент
Куликовский Д. В. – ассистент каф. ПЭ*

В работе рассмотрены затраты белорусских банков на внедрение и использование информационных технологий в своей работе.

Интерес к данной отрасли обусловлен, в первую очередь, сравнительно высоким уровнем развития информационных технологий в банках, а также вовлеченностью ИТ в ключевые бизнес-процессы банковского сектора.

В качестве источника информации используются открытые данные – отчетность банков, подготовленная в соответствии с международными стандартами финансовой отчетности (МСФО).

Выделяются четыре составляющие в структуре затрат на ИТ:

- Поступление основных средств, относящихся к ИТ;
- Поступление нематериальных активов (НМА), связанных с программным обеспечением;
- Операционные расходы на информационные технологии и обработку данных;
- Расходы на ИТ-персонал.

В структуре ИТ-бюджета наибольшая часть принадлежит основным средствам. В 2017 г. эта доля выросла по сравнению с 2016 г. (с 31% до 40%) в основном за счет уменьшения долей НМА и операционных расходов, в то время как доля затрат на персонал осталась на том же уровне (около 15%).

Лидерами по затратам на ИТ являются крупные системообразующие банки (ОАО «Белагпропромбанк» 25 млн. бел. руб., ОАО «АСБ Беларусбанк» 22 млн. бел. руб., ОАО «БПС-Сбербанк» 18 млн. бел. руб., «Приорбанк» ОАО 15,1 млн. бел. руб.).

Распределение между статьями расходов у крупнейших банков различалось. Например, «Приорбанк» ОАО имел самые низкие в первой четверке затраты на персонал (4,1 млн. бел. руб.) и самые высокие расходы на лицензии, связанные с приобретением нового ПО (5,9 млн. бел. руб.). ОАО «АСБ Беларусбанк» продемонстрировал самые высокие затраты на персонал (6,2 млн. бел. руб.), которые заметно превосходят показатели остальных системообразующих банков, что тоже объяснимо с учетом размеров этого банка.

Доля ИТ-расходов в доходах крупнейшей банков велика – в среднем они тратят более 10% средств, заработанных от основной деятельности, а у некоторых их них этот показатель превышает отметку 20 % (например, ЗАО «Альфа-Банк» 22% и ОАО «Банк БелВЭБ» 25%).

В целом отношение уровня затрат на ИТ к доходам банков в Беларуси значительно отличается в меньшую сторону от общемировых тенденций (Беларусь по банковскому сектору – 10%, мир – 21%).

Для оценки влияния затрат на ИТ и развитии тем самым бизнеса банков были использованы два относительных показателя: ИТ-расходы на одного сотрудника и чистые доходы на одного сотрудника. Расчеты основаны на открытой информации об общей численности сотрудников банков, размещенной в сети Интернет.

Лидером по данному критерию является «Приорбанк» ОАО, демонстрирующий наиболее высокие затраты на ИТ, но в то же время высокие доходы на одного сотрудника (6,24 тыс. бел. руб. и 5,98 тыс. бел. руб. соответственно). В первой пятёрке также находится ЗАО «МТБанк», который поддерживает на высоком уровне затраты на ИТ, сопоставимом с гораздо более крупными системообразующими банками (3,96 тыс. бел. руб. – затраты, 3,54 тыс. бел. руб. – доходы на одного сотрудника).

Из этого следует, что те банки, которые вкладывают больше средств в ИТ, как правило получают более высокие доходы. Поэтому можно считать, что затраты на ИТ носят инвестиционный характер.

В заключение хотелось бы отметить, что эффективное управление инвестициями в ИТ становятся не только технической, но и бизнес-функцией, которая напрямую связана с результатами работы банка.

Список использованных источников:

1. Официальный сайт Национального банка Республики Беларусь [Электронный ресурс]. – Режим доступа: www.nbrb.by/ – Дата доступа: 15.03.2018.

2. Налоговый кодекс Республики Беларусь. Особенная часть [Электронный ресурс] 29 дек. 2009 г., № 71-3 : принят Палатой представителей 11 дек. 2009 г. : одобр. Советом Республики 18 дек. 2009 г. : с изм. и доп. по состоянию на 31.12.2013 г. // КонсультантПлюс. Респ. Беларусь / ЗАО «КонсультантПлюс». – Минск, 2014.

ЭФФЕКТ КАЗИМИРА И ЕГО ПРИМЕНЕНИЕ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Мельченко А.О.

Тараканов А. Н. – доцент каф. ФМД, канд. физ.-мат. наук, доцент

Рассматривается открытый в 1948 эффект Казимира и его применение в различных областях физики.

Термин «эффект Казимира» представляет общее название для физических явлений, имеющих квантовую природу и обусловленных существованием нулевых колебаний квантованных полей в состоянии вакуума ([1], [2]). Диапазон областей физики, в которых проявляется эффект Казимира, очень широк – от теории межмолекулярных взаимодействий до физики элементарных частиц и космологии.

В 1948 голландские химики Эверт Йоханнес Виллем Вервей и Ян Теодор Герард Овербек из Физической лаборатории Н. В. Филиппса в Эйндховене опубликовали теоретические исследования по электрохимии коллоидных растворов, выполненные в 1940-1944, в которых попытались объяснить стабильность гидрофобных коллоидов и взвесей на основе концепции уравнивания силы взаимного отталкивания двух электрохимических двойных слоёв и сил притяжения Лондона-Ван дер Ваальса. Оказалось, что между их теорией и экспериментом имеется существенное расхождение, которое можно было устранить, предположив, что на больших расстояниях притягивающая сила между двумя атомами уменьшается значительно быстрее, чем

R^{-7} . Овербек затем указал, что по мере того, как расстояние между частицами становится сравнимым с длиной волны, соответствующей атомным частотам, следует ожидать влияние запаздывания на взаимодействие. Работавшие вместе с ними физики Хендрик Казимир (1909-2000) и Дирк Полдер (1919-2001) рассмотрели две академические задачи о взаимодействии нейтрального атома с идеально проводящей плоскостью и взаимодействии между двумя атомами с учётом влияния запаздывания ([3]). Оказалось, что влияние запаздывания приводит к уменьшению энергии взаимодействия на поправочный множитель, который монотонно уменьшается с увеличением расстояния R . На больших расстояниях этот множитель пропорционален R^{-1} , так что энергия взаимодействия атома с плоскостью оказалась равной $-3\hbar c \alpha / 8\pi R^4$, а энергия взаимодействия атома с атомом равной $-23\hbar c \alpha_1 \alpha_2 / 4\pi R^7$, где α , α_1 , α_2 – статические поляризуемости атомов. Вскоре Казимир показал, что эти выражения можно получить из классической электродинамики, рассматривая изменение электромагнитной энергии нулевых колебаний ([4]). Продолжая в том же духе, он получил выражения для поверхностной плотности энергии взаимодействия двух идеально проводящих пластин: $E = -\pi^2 \hbar c / 720a^3$,

где a – расстояние между пластинами. Соответствующая сила $F = -dE / da = -\pi^2 \hbar c / 240a^4$ может быть интерпретирована как отрицательное давление между пластинами, на основании чего Казимир сделал вывод, что «между двумя металлическими пластинами существует сила притяжения, которая не зависит от материала пластин, до тех пор пока расстояние не станет настолько большим, что для длин волн, сравнимых с этим расстоянием, глубина проникновения мала по сравнению с этим расстоянием. Эта сила может быть интерпретирована как нулевое давление (zero point pressure) электромагнитных волн» ([5]).

Силы подобного рода, предсказанные теоретически, были названы силами Казимира-Полдера. Более общо, эффект Казимира - это эффект, заключающийся во взаимном притяжении проводящих незаряженных тел под действием квантовых колебаний (флуктуаций) в вакууме. Чаще всего речь идёт о двух параллельных незаряженных зеркальных поверхностях, размещённых на близком расстоянии, однако эффект Казимира возможен не только между плоскостями, но и с другими, более сложными объектами. Величина этой силы очень мала, однако, зависит от геометрии поверхностей. В случае пластин она всегда действует перпендикулярно их плоскости и равна $F = -8,11 \times 10^{-15} \text{ МэВ Чм Ча}^{-4}$. Вследствие малости этой силы обнаружить её экспериментально чрезвычайно затруднительно.

Первые эксперименты были проведены в 1957 Маркусом Спаарнеем ([6], [7]), который для металлических пластин из хрома и хромовой стали, расстояние между которыми изменялось в пределах $0,3 < a < 2,0$ мкм, получили согласие с предсказанием с погрешностью менее 0,01 мкм. Аналогичные результаты были получены для алюминиевых пластин.

Обеспечить параллельность пластин при субмикронном зазоре чрезвычайно трудно, поэтому большинство экспериментов по исследованию эффекта Казимира проводили, заменяя одну из пластинок сферой. В этом случае сила притяжения обратно пропорциональна кубу расстояния между ней и пластиной. Притяжение между плоской и сферической металлическими поверхностями исследовали при помощи так называемого атомного силового микроскопа. Был учтён вклад электростатических зарядов, неровности поверхностей и прочих мешающих факторов. Удалось также обнаружить предсказанную зависимость величины эффекта от температуры тел. Эксперименты подтвердили теорию с точностью до 1%.

Существование эффекта Казимира можно рассматривать как доказательство нулевых колебаний (или так называемых флуктуаций) вакуума, когда он находится между стенками, или в какой-то замкнутой области. Как известно, Планк в 1911 предложил закон квантования энергии излучения в форме $E_n = (n + 1/2)\hbar\omega$, что позволило Дебаю построить теорию теплоёмкости твёрдых тел при низких температурах. Флуктуации вакуума можно представить в форме колебаний связанных электрон-позитронных пар, порождающих квантованное электромагнитное излучение в форме виртуальных фотонов, причём нулевым колебаниям вакуума соответствуют различные длины волн. Это излучение резонирует с собственными колебаниями пластин, рассматриваемых как резонатор. При резонансе в пространстве между пластинами существуют только те стоячие волны, для которых соблюдается условие резонанса: на расстоянии a между пластинами укладывается целое число n полуволн, т.е. $a = n\lambda / 2$. Максимально возможная длина волны $\lambda = 2a$ будет при $n = 1$. Поэтому в пространстве между пластинками виртуальные фотоны с длинами волн, превышающими $2a$, рождаться не могут. Поэтому плотность энергии нулевых колебаний в зазоре между пластинками меньше, чем снаружи, что и обуславливает притяжение пластинок.

С начала 70-х годов эффект Казимира активно изучался в космологии, где его проявление может быть связано с неевклидовостью топологии пространства-времени. Спектр нулевых колебаний отличается от случая пространства Минковского, что позволяет установить глобальную топологическую структуру Вселенной по результатам локальных измерений. Если средняя плотность материи во Вселенной меньше критического значения 10^{-92} г/см^3 , то пространство нашего мира подобно поверхности гиперболоида вращения, если средняя плотность равна критической, то пространство является плоским. Если же средняя плотность превосходит критическую, то Вселенная замкнута и имеет конечный объём. Однако ситуация оказывается не такой простой, так как средняя плотность материи известна лишь очень приближённо, и её значение ненамного отличается от критического, причём неясно даже, в сторону увеличения или уменьшения. Кроме того, наблюдательные данные о средней плотности всегда относятся к конечному объёму, и поэтому, опираясь только на них, в принципе нельзя сделать вывод о конечности или бесконечности Вселенной. Эффект Казимира позволяет решить этот вопрос. Если мы живём в гиперболическом или плоском мире, то эффекта Казимира нет, а если в сферическом, то он должен проявляться. Соответствующая положительная плотность энергии вакуума очень мала, однако в принципе её можно зафиксировать в локальных измерениях и по их результатам реконструировать структуру Вселенной в целом – в частности, решить проблему конечности – бесконечности.

Эффект Казимира активно использовался в физике сильно взаимодействующих частиц, адронов, представляемых в виде мешка, т. е. области пространства, в которой заперты кварки и глюоны, которые движутся по геодезическим траекториям и не могут вылететь за пределы этой области. В частности, первая идея Казимира в этом плане заключалась в применении эффекта для объяснения загадочной стабильности электрона, представляемого в виде шара, ограниченного проводящей поверхностью сферы ([9]). Однако, казимировская энергия сферы оказалась положительной, что соответствует силам отталкивания, а не притяжения. В случае адронов ситуация оказывается более сложной, так как кварки и глюоны взаимодействуют.

В 80-х годах эффект Казимира активно применяется в суперсимметричных теориях типа Калуцы-Клейна, в которых учёт спектра нулевых колебаний определяет механизм спонтанной компактификации (или, так называемой размерной редукции) дополнительных измерений N -мерного пространства-времени.

В настоящее время эффект Казимира стало возможным наблюдать в связи с развитием прецизионных методов измерений. В частности, в 2011 был обнаружен динамический эффект Казимира, возникающий при ускоренном движении зеркала или изучаемого объекта. В отличие от классической электродинамики, в которой возникает излучение ускоренного электрического заряда, возможно излучение ускоренной нейтральной частицы. В динамическом эффекте Казимира вакуумные флуктуации служат затравкой, приводящей к рождению реальных (а не виртуальных) фотонов, на рождение которых тратится кинетическая энергия зеркала.

После открытия в 1948 эффекта Казимира появилось несколько сот работ, посвящённых различным аспектам применения этого эффекта, причём их количество увеличивалось с каждым годом. Несмотря на 70 лет после открытия эффекта интерес к этому явлению остаётся, у него много различных приложений с интересными перспективами. Не исключено, что он станет новым тестом для проверки предсказаний фундаментальных физических теорий.

Список использованных источников.

1. Мостепаненко, В.М. Эффект Казимира и его приложения./ В.М. Мостепаненко, Н.Я. Трунов // УФН, 1988, 156, Вып. 3, 385-426.
2. Мостепаненко В.М., Эффект Казимира и его приложения/ В.М. Мостепаненко, Н.Я. Трунов – М.: Энергоатомиздат, 1990. – 216 с..
3. Casimir, H.B.G. The Influence of Retardation on the London-van der Waals Force/. H.B.G. Casimir, D..Polder // Phys. Rev., 1948, 73, no. 4, 360-372.
4. Casimir, H.B.G. Sur les forces Van der Waals-London. (Colloque sur la théorie de la liaison chimique/. H.B.G. Casimir //Paris, 12-17 April, 1948). // J. Chim. Phys., 1949, 46, 407-410.
5. Casimir, H.B.G. On the attraction between two perfectly conducting plates/ H.B.G. Casimir // Proc. Kon. Ned. Akad. Wet., 1948, 51, no. 7, 793-795.
6. Sparnaay, M.J. Attractive Forces between Flat Plates/. M.J. Sparnaay // Nature, 1957, 180, no. 4581, 334-335.
7. Sparnaay, M.J. Measurements of attractive forces between flat plates/ M.J.Sparnaay // Physica, 1958, 24, no. 6-10, 751-764.
8. Wilson, C.M. Observation of the Dynamical Casimir Effect in a Superconducting Circuit/. Wilson, C.M. and etc.// Nature, 2011, 479, no. 7373, 376-379.
9. Casimir, H.B.G. Introductory remarks on quantum electrodynamics/ H.B.G. Casimir// Physica, 1953, 19, 846-849.

КОДИРОВАНИЕ СЕГМЕНТИРОВАННОГО ОБЪЕКТА

Институт информационных технологий БГУИР
г. Минск, Республика Беларусь

Предкель Д.А.

Митюхин А. И. – доцент каф. ФМД

Представлен сравнительный анализ кодирования коррелированных данных, полученных на этапе сегментации. Результаты анализа точности восстановления данных позволяют выбрать метод обработки, позволяющий уменьшить вычислительную сложность, упростить техническую реализацию цифрового процессора.

Основным ограничением, которое возникает при необходимости эффективной передачи информации во многих технических приложениях является пропускная способность $C = I(X:Y)_{\max}$ каналов обмена данными. Специальные инфокоммуникационные средства передачи с комплексным представлением информации, с интерактивным взаимодействием и защитой от несанкционированного доступа особенно чувствительны к названному ограничению. Передача различных типов данных по каналам с ограниченной шириной полосы частот (даже по закрытым радиоканалам) сказывается на показателях быстродействия системы. Однако, в некоторых приложениях выдвигаются требования по сжатию данных, например, в отношении 3600:1.

Эффективное кодирование (сжатие) информации основано на выполнении двух основных операций. Первая операция – это декоррелирующее линейное преобразование статистически зависимых данных. Вторая операция – это энтропийное или универсальное кодирование. Цифровой эквивалент аналогового сигнала путем соответствующего линейного ортогонального преобразования приводится к виду, который наиболее удобен с точки зрения сокращения статистической избыточности информации. Для сжатия можно использовать такие линейные преобразования, как дискретное преобразование Адамара (ДПА), дискретное косинусное преобразование (ДКП), дискретное преобразования Хартли (ДПХ), Хаара и ряд других [1]. Прямое двумерное дискретное преобразование исходного сегментированного объекта размером $M \times N$ в матричном обозначении записывается как

$$\hat{G}_{m,n} = W_{m,m} G_{m,n} W_{n,n}^T \quad (1)$$

где W – ядро преобразования. Восстановление данных после сжатия с использованием порогового отбора коэффициентов преобразования происходит по выражению

$$\tilde{G}_{m,n} = W_{m,m}^T \hat{G}_{m,n} W_{n,n}$$

Пример. На рис. 1 показан в виде матрицы G размером 4x4 сегментированный объект. На рис. 2, 3, 4 показаны спектральные образы (1) соответственно ДПХ, ДПА и ДКП.

$$G = \begin{pmatrix} 28 & 27 & 22 & 23 \\ 25 & 22 & 18 & 19 \\ 26 & 23 & 20 & 21 \\ 19 & 21 & 19 & 18 \end{pmatrix}$$

Рисунок 1 - Исходные данные

$$\hat{G}_H = \begin{pmatrix} 351 & 31 & 3 & 7 \\ 17 & 9 & 1 & 5 \\ 29 & 5 & 1 & 5 \\ 3 & -5 & -5 & -9 \end{pmatrix}$$

Рисунок 2 - Коэффициенты ДПХ

$$\hat{G}_{Над} \begin{pmatrix} 351 & 3 & 31 & 7 \\ 29 & 1 & 5 & 5 \\ 17 & 1 & 9 & 5 \\ 3 & -5 & -5 & -9 \end{pmatrix} \dots$$

Рисунок 3 - Коэффициенты ДПА

$$\hat{G}_{DCT} \begin{pmatrix} 87.75 & 7.448 & 1.75 & -2.268 \\ 6.697 & 2.487 & 1.632 & -0.675 \\ 0.75 & -1.633 & -2.25 & -0.678 \\ 5.075 & 0.325 & 0.678 & 0.012 \end{pmatrix}^T$$

Рисунок 4 - Коэффициенты ДКП

Эффективность кодирования (1) оценивалась путем вычисления среднеквадратической ошибки (СКО)

$$e = \sqrt{\frac{1}{NM} \sum_{m=1}^M \sum_{n=1}^N (g(m,n) - \tilde{g}(m,n))^2}$$

для разных значений порога $h \geq \hat{g}_{i,j} \in G$. Результаты представлены в таблице 1.

Таблица 1 - Среднеквадратическая ошибка

K	1,77	2,66	4	5,3	8	16
едпх	0,16	0,50	0,80	1,2	1,7	2,3
едпу	0,16	0,50	0,80	1,2	1,7	2,3
едкп	0,13	0,40	0,57	0,7	1,4	2,3

Коэффициент сжатия определяется отношением длины всех кодовых слов исходного изображения к суммарной длине слов, предназначенных к передаче.

$$K = NMn/Ln = NM/L$$

Как видно, понижение размерности обрабатываемого сегмента с меньшей среднеквадратической ошибкой обеспечивается применением ДКП. Далее, к полученным коэффициентам применяется алгоритм энтропийного или универсального кодирования.

Список использованных источников.

1. Митюхин, А.И. Цифровая обработка речи и анализ изображений/ А.И. Митюхин - Минск. БГУИР, 2016.

АКТУАЛЬНОСТЬ ЭЛЕКТРОННЫХ БИБЛИОТЕК В УЧРЕЖДЕНИЯХ СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Пунцель Е.В.

*Майсеня Л.И. – зав. кафедрой ФМД, докт. пед. наук, профессор,
Буянова С.Г. – преподаватель*

Нарастающая популярность информационных технологий затрагивает и образовательный процесс. В статье рассмотрена актуальность внедрения электронной библиотеки в учреждение среднего специального образования, а также описаны основные функции, которые данная библиотека выполняет.

Большинство компаний, будь то крупная корпорация или фирма с малым бизнесом, используют информационные технологии для автоматизации своих бизнес-процессов, где Internet зачастую выступает в качестве основной технологии, а иногда и единственной. Новые информационные технологии достигли такого развития, что не осталось сфер человеческой жизни, которые не затронула глобальная сеть Internet.

В настоящее время широко применяется концепция электронных библиотек, когда пользователю не нужно выходить из дома чтобы получить необходимый информационный ресурс. Современную библиотеку трудно представить без компьютера и специализированного программного обеспечения, позволяющего эффективно осуществить выбор необходимого издания пользователями. Такие системы позволяют удобно и быстро работать с огромным числом наименований разных изданий.

Следует отметить, что на уровне учреждений среднего специального образования в нашей стране такой ресурс как электронная библиотека используется далеко не в каждом учреждении, поэтому актуальны ее разработка и внедрение. Использование данного ресурса способствует скорейшему и простейшему нахождению

дополнительной литературы учащимся, что, в свою очередь, способствует лучшему усвоению ими учебного материала.

Пример разработанной нами электронной библиотеки можно увидеть в учреждении образования «Республиканский институт профессионального образования», филиал «Индустриально-педагогический колледж», посетив сайт учреждения. Целевая аудитория сайта – учащиеся, обладающие знаниями в IT-технологиях, которые не хотят тратить время на хождение в библиотеку. Сайт представляет собой инструмент, предоставляющий учащимся информацию об имеющихся учебных ресурсах с последующей возможностью скачивания. После перехода по ссылке с сайта колледжа в электронную библиотеку, пользователь попадает на главную страницу сайта, где ему необходимо авторизоваться, если требуется просмотр и скачивание литературных источников. Фрагмент главной страницы представлен на рисунке 1.

Далее пользователь может посетить любой раздел Web-сайта. Например, при наведении на раздел «Категории» предоставляется возможность выбрать необходимое отделение, а затем выбрать специальность (рисунком 2). После этого появится список книг, соответствующий выбранным критериям.

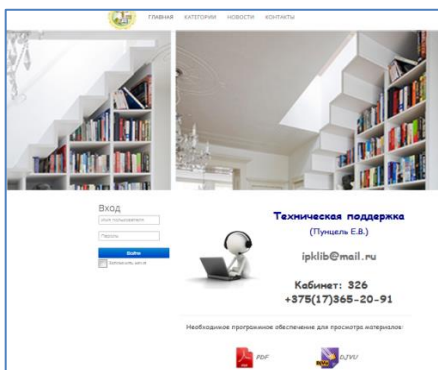


Рисунок 1 – Фрагмент главной страницы электронной библиотеки колледжа



Рисунок 2 – Поиск книги для конкретной специальности нужного отделения

Авторизация на сайте необходима для того, чтобы различать педагогов и учащихся данного учреждения образования, а также с целью получения литературы сотрудниками или учащимися колледжа (без внешних пользователей).

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И НЕЙРОННЫЕ СЕТИ

*Институт информационных технологий БГУИР,
, г. Минск, Республика Беларусь*

Росляков А.П., Цацура К.М.

Ламчановская М. В. – доцент каф. ФМД, канд. физ.-мат. наук, доцент

Нейронные сети – одно из направлений исследований в области искусственного интеллекта. В докладе рассматриваются принципы работы нейронных сетей и сфера их применения.

Искусственный интеллект – способность приспосабливаться к новым ситуациям, способность к обучению и запоминанию на основе опыта, пониманию, применению и использованию полученных знаний для управления окружающей средой. «Интеллект – способность системы создавать в ходе самообучения алгоритмы для решения задач определённого класса сложности и решать эти задачи». Возможности компьютера в плане скорости вычислений оказались намного больше человеческих, и учёные задались вопросом: каковы границы возможностей вычислительной техники и смогут ли машины достигнуть уровня развития человека? В 1950 году английский учёный Алан Тьюринг, написал статью «Может ли машина мыслить?», в которой описывает процесс, с помощью которого можно будет зафиксировать момент, когда вычислительная машина сравняется в плане интеллекта с человеческим. Испытание назвали в честь автора, тест Тьюринга.

В работе Oxford Martin School 2013 года говорилось о том, что 47% всех рабочих мест может быть автоматизировано в течение следующих 20 лет. Основным драйвером этого процесса является применение искусственного интеллекта(ИИ), как более эффективной замены человеку.

Идея нейронных сетей (НС) появились в ходе исследований в области ИИ, а именно как результат попыток повторить способность биологических нервных систем учиться и исправлять ошибки. У нейрона сложная структура, которая состоит дендритов – устройства ввода информации, ядра – основы и аксона – разветвляющийся выходы. Клетки аксоны соединяются с дендритами иных клеток при помощи синапсов. При возбуждении нейрон посылает электрохимический импульс по своему аксону. Импульсы через синапсы достигают остальных нейронов, они в свою очередь могут возбуждаться. Нейрон возбуждается, когда общий уровень импульсов, пришедших в его ядро из дендритов, превышает определенный уровень (порог возбудимости).

Модели НС могут быть как программной, так и аппаратной реализации (рисунок 1). В дальнейшем рассмотрим первый тип. Исключая значительные различия, отдельные типы НС имеют несколько схожие характеристики.

Из рисунка 1 (а) видно, что искусственный нейрон, полностью повторяет живой, он состоит из синапсов, которые связывают входы нейрона с ядром, и самого ядра нейрона, которое выполняет сбор и обработку поступающих импульсов и аксона, который объединяет нейрон с нейронами соседнего слоя. Одна из характеристик синапса – вес, который показывает в какой мере соответствующий вход нейрона имеет влияние на его состояние. Помимо этого, синапс обладает такой характеристикой как синаптическая связь или ее вес, которая по физическому значению равносильна электрической проводимости.

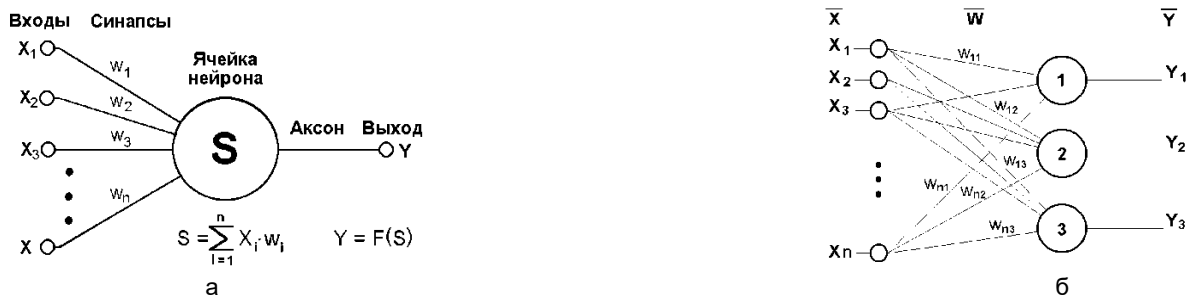


Рисунок 1 – Искусственный нейрон(а), трехнейронный перцептрон (б)

Одной из общих характеристик присущим всем НС, параллельный принцип обработки сигналов. Достигается путем объединения огромного числа нейронов в слои, а также, соединения определенным образом нейронов разных слоев, и нейронов одного слоя между собой. Взаимодействие и обработка всех нейронов происходит послойно.

В качестве примера простейшей НС рассмотрим трехнейронный перцептрон (рис 1, б). На n входов поступают определённые импульсы, проходящие по синапсам на три нейрона, формирующие единственный слой этой НС и выдающие три выходных импульса. Теоретически количество слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется НС. Чем сложнее структура НС, тем масштабнее задачи, она способна решить. Выбор структуры НС осуществляется в соответствии с особенностями и сложностью задачи. В ситуации когда задача не может быть приведена ни к одному из известных типов, разработчику необходимо решать сложную проблему синтеза нового набора конфигураций. При этом он обязательно руководствуется несколькими основополагающими принципами: возможности сети возрастают с увеличением числа ячеек сети, плотности связей между ними и числом выделенных слоев; введение обратных связей наряду с увеличением возможностей сети поднимает вопрос о динамической устойчивости сети; сложность алгоритмов функционирования сети также способствует усилению мощи НС. Вопрос о необходимых и достаточных свойствах сети для решения того или иного рода задач представляет собой целое направление нейрокомпьютерной науки. Так как проблема синтеза НС сильно зависит от решаемой задачи.

Очевидно, что процесс функционирования НС, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, задавшись определенной структурой НС, отвечающей какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов. Этот этап называется обучением НС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время эксплуатации. На этапе обучения кроме параметра качества подбора весов важную роль играет время обучения. Как правило, эти два параметра связаны обратной зависимостью и их приходится выбирать на основе компромисса. Если кратко, то: в простейшей модели машинного обучения необходимую информацию предоставляет сама среда. Обучаемый получает информацию, которую фиксирует и модернизирует в своей базе данных, знания из которой функциональный элемент использует для решения некоторой задачи.

Для моделей, которые созданы на основах человеческого мозга, характерна не слишком большая выразительность, легкое распараллеливание алгоритмов, и связанная с этим высокая производительность параллельно реализованных НС. Такие сети в каком-то смысле копирует человеческий мозг, ведь НС могут работать даже при условии неполной информации. НС на поставленные вопросы способны дать ответ не только "да" или "нет" но и "не знаю точно, но скорее всего да".

Одной из успешных сфер применения НС – задачи распознавания образов, например, сильно зашумленных. Вот некоторые важные примеры задач, в которых были успешно решены нейросетевые методы: распознавание состояния больного, прогнозирование на фондовом рынке, предоставление кредита, системы слежения за состоянием оборудования, управление работой двигателя, военная промышленность, нейросети в искусственном интеллекте. Область применения ИИ: доказательство теорем, игры, распознавание образов, принятие решений, адаптивное программирование, сочинение машинной музыки, обработка данных на естественном языке.

Однако все ли так хорошо в перспективе развития ИИ и НС? Сложно думать о том, чистый как слеза младенец созданный ИИ однажды смогут превратиться в монстров футуристической антиутопии. Илон Маск, Стив Возняк и ещё много других специалистов в области компьютерных наук и ИИ подписали открытое письмо, в котором призывали не использовать ИИ в военных компаниях (например, дроны, которые сами летают и уничтожают цели). Сейчас у людей нет конкретного понимания, как будет работать ИИ, каким образом он будет принимать решения и развиваться. Именно это является камнем преткновения и поэтому умнейшие люди планеты опасаются: что ИИ в какой-то момент станет умнее и хитрее своих создателей, и начнет изменять сам

себя непредсказуемым образом. А об этом люди смогут узнать, только когда станет слишком поздно. Пока, никто не знает, будут ли изменения ИИ и их попытки внедрения в жизнь людей положительными для самих людей или нас ждет неопределенный конец, хотя некоторые возлагают большие надежды.

В заключении отметим, развитие ИИ со временем приобретает все большую популярность, ведь он имеет некоторую привилегию, однако выдающимся умам еще не довелось создать полностью совершенного интеллекта. Ответ на вопрос, какое будущее принесет нам ИИ и нейронные сети, остается открытым, по сей день.

Список использованных источников:

1. Тьюринг, А. М. Вычислительные машины и разум/А. М. Тьюринг //Глаз разума /Д. Хофштадтер, Д. Деннетт. – Самара: Бахрах-М, 2003. – С. 47–59.
2. Компьютер учится и рассуждает (ч. 1) // Компьютер обретает разум = Artificial Intelligence Computer Images / под ред. В. Л. Стефанюка. – Москва: Мир, 1990. – 240 с.
3. Девятков, В.В. Системы искусственного интеллекта / В.В. Девятков. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. – 352 с.
4. AIportal [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.aiportal.ru/articles/neural-networks/neural-networks.html>. Дата доступа 15.03.2018.

ДЕКОДИРОВАНИЕ ОРТОГОНАЛЬНОГО НЕЛИНЕЙНОГО КОДА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Хадкевич О.В.

Митюхин А. И. – доцент каф. ФМД

В работе рассматривается задача декодирования нелинейного помехоустойчивого ортогонального кода, обеспечивающего коррекцию ошибок и защиту информации в основном информационном канале и в канале с подслушиванием.

Одно из применений нелинейных ортогональных помехоустойчивых $[N, M, d]$ -кодов длиной N и кодовым расстоянием d заключается в коррекции ошибок – обнаружении и исправлении t ошибок в канале с аддитивным гауссовским шумом $n(i)$ [1]. Другое применение – это обеспечение определенного уровня защиты информации от несанкционированного доступа. Свойство нелинейности кода позволяет иметь значительно больший ансамбль M кодовых слов над полем Галуа $\{F\}$ в сравнении с линейными кодами, что важно для защиты информации. Рассматривается подход решения задачи декодирования помехоустойчивого кода для системы, в которой предусмотрена защита информации и коррекция ошибок. Предполагается, что преднамеренные генерируемые ошибки в принятом сигнале в канале подслушивания должны мешать правильному декодированию перехватываемой информации.

Пусть слова кода $X = X^1, X^2, \dots, X^s, \dots, X^M$, $X^s = (x(1), \dots, x(N))$ передаются по основному каналу с шумом $n(i)$. Структура кодовых слов $\{X\}$ априори известна на приемной стороне. На выходе канала формируется аддитивный процесс

$$y^s(i) = x^s(i) + n^s(i), i = 1, 2, \dots, N, \quad (1)$$

который можно представить в виде вектора наблюдения $Y^s = (y(1), \dots, y(N))$. Экспериментально вычисляя вероятности $P(Y^s)$ на множестве $\{Y\}$, имея априорные значения вероятностей $\{P(X)\}$ кодовых слов (входа канала) и зная свойства канала – переходные вероятности $P(Y|X^s)$, можно найти вероятность $P(X^s|Y)$ – вероятность получения слова X^s кода X на приемной стороне (выходе канала). В соответствии с теоремой Байеса вероятность получения слова входа основного канала, при условии, что на выходе канала уже получен вектор Y , определяется как

$$P(X^s|Y) = \frac{P(Y|X^s)P(X^s)}{P(Y^s)}. \quad (2)$$

Так как вероятности $P(X^s)$ и $P(Y^s)$ известны, отношение $\frac{P(X^s)}{P(Y^s)}$ равно постоянной величине $K = \frac{P(X^s)}{P(Y^s)}$. Выражение (2) примет вид $P(X^s|Y) = K P(Y|X^s)$. Из последней записи следует, что процедура декодирования по основному каналу заключается в нахождении такого значения номера s кодового слова, при котором значение апостериорной вероятности $P(X^s|Y)$ достигает максимума. Таким образом, декодирование требует вычисления $\max P(Y|X^s)$ – вероятности получения Y при условии, что было передано кодовое слово X^s . Если $Y = X^s$, достигается $\max P(Y|X^s)$ и обеспечивается прием с нулевой вероятностью ошибок. Поскольку из (1) $y^s(i) - x^s(i) = n^s(i)$, функцию $P(Y|X^s)$ отобразим как

$$P(Y|X^s) = P(Y - X^s = \mathbf{n}) = P(\mathbf{n}),$$

где $\{\mathbf{n}\}$ – множество преднамеренных шумовых векторов, $P(\mathbf{n})$ – вероятности векторов шума. Выражение $\{Y - X^s\}$ – это множество $\{d_x\}$ расстояний Хемминга между словом Y на входе декодера и всеми словами кода. С позиции теории кодирования d_x показывает, сколько символов в слове надо исказить, чтобы перевести одно разрешенное для передачи кодовое слово в другое разрешенное. Тогда вычисление функции $\max P(Y|X^s)$ сводится к нахождению опорного кодового вектора X^s ближайшего по расстоянию Хемминга к принятому вектору Y (на выходе канала). Если использовать пространственную интерпретацию кода как множество точек (векторов) N -мерной решетки, то вероятность того, что точка X^s совпадет с точкой Y , увеличивается с уменьшением евклидова расстояния. Степень близости точек X^s и Y в пространстве размерностью N легко вычисляется с помощью скалярного произведения

$$\langle Y|X^s \rangle = \sum_{i=1}^N y(i) x^s(i) \quad (3)$$

векторов, описывающих точки. Эта операция лежит в основе декодирования по основному каналу. Из проведенного анализа следует, что для успешного перехвата информации по каналу подслушивания необходимо иметь априорные знания об основных параметрах кода, в частности, мощности множества $\{X\}$ и переходных вероятностях канала. Задача декодирования еще более усложняется, когда шум используется с целью маскирования информации. Тогда декодирование должно выполняться по вектору наблюдения $Y = X + \mathbf{n}$. Отсутствие точной информации об алгебраической структуре кода X и множестве случайных векторов $\{\mathbf{n}\}$ в канале с подслушиванием не позволяет перехватчику декодировать сигнал Y по алгоритму (3). Решение задачи перехвата на основе многоканальной обработки по (3) и перебором последовательностей X и \mathbf{n} потребует значительных вычислительных, временных и технических ресурсов.

Список использованных источников.

1. Митюхин, А.И. Корреляционные спектры и кодовые расстояния мажоритарных последовательностей/А.И. Митюхин, П.Н. Якубенко// Доклады БГУИР. – 2015. № 4 (90). – С. 5–9.

ЗАЩИТА ИНФОРМАЦИИ НА ОСНОВЕ НИЗКОСКОРОСТНОГО КОДИРОВАНИЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шлома К.Н.

Митюхин А. И. – доцент каф. ФМД

Анализируется один из основных сервисов информационной безопасности – помехоустойчивое кодирование и защита информации от несанкционированного доступа в радиоэлектронных системах. Рассматривается система кодирования информации с использованием множества неприводимых полиномов над полем $GF(2)$ в каналах с гауссовским шумом. Дана оценка возможных временных затрат декодирования при приеме кодированных данных в условиях априорной неопределенности.

Для обеспечения информационной безопасности при передаче данных используются помехоустойчивые коды. Защита информации в специальных радиоэлектронных системах гражданского и военного назначения от воздействия непреднамеренных и преднамеренных помех осуществляется посредством низкоскоростного помехоустойчивого кодирования [1]. Кроме защиты информации от ошибок (коррекции информации) в каналах с шумами, такие коды над конечным полем $GF(q)$ обеспечивают определенную степень безопасности информационных комплексов от случайного и несанкционированного доступа к информации. Решение этой задачи основывается на применении множества изменяющихся во времени $[n, k, d]$ -кодов

$$C = \{C^1, \dots, C^J, \dots, C^L\}, C^j \in C^J, C^j = (c_1, \dots, c_i, \dots, c_n), c_i \in GF(q),$$

где k – размерность j -кода, $n = q^k - 1$ – длина j -кода, d – минимальное расстояние j -кода (характеризует корректирующую способность кода), C^j – кодовая последовательность с символами из поля $GF(q)$, $M = q^k - 1$ – количество слов j -кода (мощность множества), q – простое число (основание кода). Практическая алгебраическая конструкция псевдощумового низкоскоростного $[n, k, d]$ -кода основывается на применении неприводимого над полем $GF(2)$ полинома вида

$$h(x) = 1 + h_1x + \dots + h_{m-1}x^{m-1} + h_mx^m,$$

где коэффициенты $h_i \in \{0, 1\}$, $k = m$. Число возможных различных неприводимых над полем $GF(2)$ полиномов $h(x)$ степени m определяется по формуле

$$L = \frac{\varphi(n)}{m},$$

где $\varphi(n)$ – функция Эйлера. Напомним, если n – простое число, то $\varphi(n) = n - 1$. Для реальных значений степеней m и длин n , т.е. в диапазоне значений $n = 2^7 - 2^{19}$ было рассчитано число полиномов $h(x)$. Соответствующие значения показаны в таблице

Таблица 1. Число неприводимых полиномов над полем $GF(2)$

m=k	7	8	9	10	11	12	13	14	15	16	17	18	19
L	18	16	48	60	176	144	630	756	1800	2048	7710	7776	27954

Пример оценки мощности кода и кодирования источника [1023, 10, 512]-кодом. Степень кодирующего полинома $h(x)$ определяется размерностью кода $k = 10$. Существуют таблицы, с помощью которых можно найти все неприводимые над полем $GF(2)$ полиномы степени $2 \leq m \leq 61$ [2]. Для $m = 10$ имеется 60 полиномов, каждый из которых образует код с числом кодовых слов $M = 2^{10} = 1024$. Суммарное количество слов с законами кодирования $h(x)^j = 1 + \dots + h_9x^9 + h_{10}x^{10}$ равно $M_{\Sigma} = LM = 60 \cdot 1024 = 61440$. Пусть проектируется система с длительностью тактового интервала $\tau = 1$ мкс c^j – последовательности. Период одного кодового слова $T = n\tau = (2^{10} - 1)1 \cdot 10^{-6} c = 1,023$ мс. Время формирования всех M_{Σ} последовательностей составит $T_{\Sigma} = M_{\Sigma}T \cong 61440 \times 1,023 \text{ мс} \cong 63 c$, т.е. около одного часа. В реальной практике стратегия любой защищаемой системы основывается на ограничении времени t передачи информации, когда $t \ll T_{\Sigma}$ и использовании значительного числа равновероятных кодовых последовательностей. Перехватчик, даже имея некоторые априорные знания о методах кодирования, должен затратить большое время, соизмеримое со значением T_{Σ} , на анализ наблюдаемого процесса с целью обнаружения и последующего декодирования сигнала. Задача перехвата усложняется из-за отсутствия априорных данных о структуре и алгоритме использования порождающих полиномов. По этой причине, для обнаружения и декодирования ничего не остается как использовать алгоритмы затратной многоканальной обработки наблюдаемого сигнала, фактически, случайного процесса. В этом случае, оптимальное декодирование на основе стратегии максимального правдоподобия [1], реализуемое в рассматриваемой системе, для перехватывающей стороны становится не возможным. Обработка кодированной информации становится не эффективной, чрезмерно затратной по оборудованию и временному ресурсу.

Список использованных источников

1. Митюхин, А.И. Корреляционные спектры и кодовые расстояния мажоритарных последовательностей/ А.И. Митюхин, П.Н. Якубенко // Доклады БГУИР. – 2015. № 4 (90). – С. 5–9.
2. Ziemer, R. E. Introduction to Digital Communication/R.E. Ziemer, R.L. Peterson // Prentice-Hall, NJ, 2001.

ТЕХНИЧЕСКИЕ И ОБЩЕСТВЕННЫЕ ДИСЦИПЛИНЫ

ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ УСТРОЙСТВ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ПИЛОТАМИ ВОЕННОЙ АВИАЦИИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Алешко Н.С., Анкуда Д.И.

Савенко А.Г. - магистр технических наук, ассистент

На сегодняшний день инструменты дополненной реальности применяются для решения широкого спектра задач, от медицины до компьютерных игр. Однако стоит отметить, что история применения AR начинается в военной авиации.

Первым устройством, использовавшим технологии AR, стал ИЛС (индикатор на лобовом стекле, head-up display (HUD)) британского самолета de Havilland Mosquito. Разработанный в 1942 году, он был призван совместить информацию, поступающую с РЛС с установленным на самолете прицелом, для обеспечения пилота информацией во время выполнения задач ночного истребителя. Совмещение происходило с помощью небольшого дисплея, установленного на одной линии с прицелом пилота [1].



Рисунок 1 – Современный ИЛС, установленный на истребителе F/A-18

Современные ИЛС позволяют демонстрировать пилоту значительно большее количество информации, например, такие показатели, как угол атаки, навигационные отметки при взлете/посадке, силу тяги двигателя и вектор траектории, а в вариантах системы, используемых военными-расстояние до цели, статус вооружения, положение сенсоров наведения, скорость сближения с целью [2]. Пример такого индикатора приведен на рисунке 1. Также у современных ИЛС существует возможность выводить на экран изображение с внешних источников, таких как подвесные контейнеры целеуказания и навигации, а также внешние видеокамеры [2].

Следующим шагом в использовании дополненной реальности в военной авиации стало создание систем нащлемного целеуказания и индикации (НСЦИ, Helmet-mounted display, HMD). Концепция устройства заключалась в выведении дополнительной информации прямо на индикатор, находящийся в шлеме пилота. Первым устройством данного типа стал Super Cockpit, разработанный по заказу ВВС США в 1969 году [3]. Конструктивно оно представляло из себя шлем, в котором совмещались дополненная и виртуальная реальность (более подробная схема изображена на рисунке 2). Основной целью разработки являлось упрощение выполнения задач на малых высотах [3].

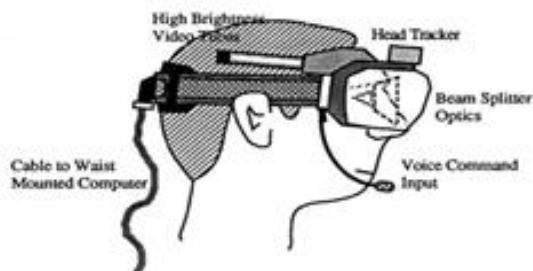


Рисунок 2 – Схема устройства Super Cockpit

В общем случае, нащлемная система целеуказания и индикации состоит из пяти основных частей: собственно, шлема, оптической системы, источника изображения, комплекса электронных систем, а также устройства отслеживания взгляда летчика. Сюда же могут входить и устройства, позволяющие ориентироваться ночью, например, модули ночного видения, которые крепятся к шлему пилота.

В дальнейшем системы НМД совершенствовались добавлением нового функционала, а именно выведения изображения с внешних камер и отображения информации о текущем состоянии летательного аппарата и маршруте полета (Integrated Helmet And Display Sight System (IHADSS), 1985 г. [4]). Очень важным событием стало создание НСЦИ «Щель-ЗУМ». Система позволяла осуществлять наведение ракеты «воздух-воздух» с помощью движений головы, что давало пилоту преимущество в воздушном бою [5]. «Щель-ЗУМ» являлась первой НСЦИ с таким функционалом, второй (и пока единственной, принятой в эксплуатацию) такой системой является Joint Helmet-Mounted Cueing System (JHMCS) (изображенная на рисунке 3), поставленной на вооружение ВВС США в 2003 году [6].



Рисунок 3 – Пилот в шлеме Joint Helmet-Mounted Cueing System

На сегодняшний день самой совершенной НСЦИ является Helmet-Mounted Display System (HMDS), разработанная специально для истребителя 5-го поколения F-35. В данном устройстве реализована интеграция с 6 внешними инфракрасными камерами, установленными на самолете, что позволяет пилоту осуществлять наблюдение буквально «вокруг» всего самолета. В шлеме присутствует возможность наведения всего спектра оружия, применяемого самолетом, а также возможность регулирования выводимой на дисплей НСЦИ информации. К примеру, пилот может выбрать отображаемые в визоре шлема параметры полета или цели [7]. Такой широкий спектр возможностей позволил конструкторам F-35 отказаться от ИЛС вовсе в пользу НСЦИ.

Проводя соответствующую оценку, можно выделить преимущества устройств дополненной реальности применительно к пилотам военной авиации.

Уменьшение времени реакции с помощью выведения и настройки объема необходимой информации с приборов на дисплей ИЛС\визор НСЦИ, что актуально при активном маневрировании, например, в ближнем воздушном бою. Особенно это касается НСЦИ с системой целеуказания – они позволяют использовать вооружение значительно быстрее, упрощая наведение до простого движения головы.

Стоит отметить и упрощение маневрирования летательного аппарата в целом. С помощью ИЛС\НСЦИ пилот может быть визуально проинформирован о том, какое маневрирование является безопасным, а какое – нет. Также стоит отметить возможность ИЛС\НСЦИ выводить пилоту информацию о рекомендуемых параметрах для взлета\посадки (которые считаются самыми опасными моментами в полете), включая скорость и высоту, а также систему «контрольных точек», что значительно упрощает действия пилота.

Однако у устройств дополненной реальности есть также очевидные недостатки.

Характерной особенностью некоторых НСЦИ является то, что конструктивно визор может использоваться лишь одним глазом, второй глаз при этом визором остается незадействованным. Из-за разницы в получаемом зрительной системой человека изображении пилоты могут ощущать дискомфорт, вплоть до сильной головной боли [8]. Также стоит отметить большую массу НСЦИ, что приводит к большой нагрузке на шею носителя и вызывает боли в спине [9].

Значимой проблемой является информационная перегрузка. Несмотря на то, что в современных НСЦИ и ИЛС существует возможность фильтрации выводимой информации, в некоторых ситуациях потоки данных, демонстрируемые пользователю, не могут быть своевременно им обработаны, что увеличивает вероятность совершения пилотом ошибки. Особенно это опасно при выполнении сложных маневров на малой высоте или во время нахождения в зоне боевых действий.

Еще одной общей проблемой устройств дополненной реальности в военной авиации является их сложность и, следовательно, надежность. Большая часть современных НСЦИ испытывает проблемы как на стадии разработки, так и на стадии непосредственной эксплуатации [10]. Эти проблемы являются решаемыми и своевременно исправляются, однако на время внесения исправлений в аппаратную или программную части НСЦИ перестает быть рабочим инструментом.

Сложность AR-устройств также потенциально уменьшает темпы подготовки пилотов. Учитывая все возможности НСЦИ, пилота требуется дополнительно обучать использовать функционал устройства, что занимает достаточно большой промежуток времени ввиду наличия широкого спектра инструментов.

Несмотря на все недостатки, устройства дополненной реальности находят все более широкое применение в военной авиации, что говорит о том, что риск их использования в сравнении с преимуществами все же не является настолько большим. С учетом постоянного совершенствования технологий и развития возможностей AR, вышеперечисленные проблемы могут быть решены в самом скором времени.

Список использованных источников:

1. White I. The History of Air Intercept (AI) Radar and the British Nightfighter /I.White.– Casemate Publishers, 2007.– 326 p.
2. Spitzer, Cary R., ed. Digital Avionics Handbook. Head-Up Displays. – Boca Raton, FL: CRC Press, 2001. – 206 p.
3. Military Applications of Augmented Reality / Mark A. Livingston [and other]. – Washington: Naval Research Laboratory, 2011.

4. USAARL Report No. 88-13 [Электронный ресурс]. – Режим доступа: <http://www.usaarl.army.mil/TechReports/88-13.PDF>. – Дата доступа: 18.03.2018
5. Авиашлемы. Виртуальная реальность в настоящем бою [Электронный ресурс]. – Режим доступа: <https://naked-science.ru/article/tech/aviashlemy-virtualnaya-realnost-v/>. – Дата доступа: 18.03.2018.
6. Joint Helmet Mounted Cueing System [Электронный ресурс]. – Режим доступа: [https://www.rockwellcollins.com/Products and Services/Defense/Avionics/Displays and Controls/Helmet Mounted Displays/ Joint Helme Mounted Cueing System.aspx](https://www.rockwellcollins.com/Products_and_Services/Defense/Avionics/Displays_and_Controls/Helmet_Mounted_Displays/Joint_Helme_Mounted_Cueing_System.aspx). – Дата доступа: 18.03.2018
7. F-35 Gen III Helmet Mounted Display System [Электронный ресурс]. – Режим доступа: [https://www.rockwellcollins.com/Products and Services/ Defense/Avionics/Displays and Controls/Helmet Mounted Displays/F-35 Gen III Helmet Mounted Display System.aspx](https://www.rockwellcollins.com/Products_and_Services/Defense/Avionics/Displays_and_Controls/Helmet_Mounted_Displays/F-35_Gen_III_Helmet_Mounted_Display_System.aspx). – Дата доступа: 18.03.2018
8. Rash C.E. Helmet Mounted Displays: Design Issues for Rotary-wing Aircraft / C. E. Rash. – SPIE, 2001. – 258 p.
9. Newman D.G. Flying Fast Jets: Human Factors and Performance Limitations/D.G. Newman.– CRC Press, 2014.– 184 p.
10. Pentagon: Here are all the problems with the F-35 [Электронный ресурс]. – Режим доступа: <http://www.businessinsider.com/here-are-all-the-problems-with-the-f-35-that-the-pentagon-found-in-a-2014-report-2015-3>. – Дата доступа: 18.03.2018.

УГРОЗЫ ДЛЯ УСТРОЙСТВ, РАБОТАЮЩИХ ПОД УПРАВЛЕНИЕМ ОС ANDROID

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Бакунович Д.В.

Пачинин В.И., зав. кафедрой ИСиТ, к.т.н., доцент

В докладе проведен анализ угроз для устройств, работающих под управлением ОС Android. На основе проведенного анализа рассмотрен ряд примеров устранения возникающих угроз и расширения возможностей программных продуктов.

В настоящее время мобильные устройства активно внедряются во все сферы жизни современного человека. Смартфоны, планшеты, электронные книги – все эти устройства уже стали неотъемлемой частью жизни. Большинство мобильных устройств работают на ОС Android. Однако эта ОС используется практически везде: начиная от домашних бытовых приборов (холодильники, телевизоры и т.д) и заканчивая различными помощниками управляемых голосом, жестами и т.д. Существуют системы “Умный дом” которые управляются при помощи голоса, жестов и приложении.

Такое широкое распространение ОС Android повлияло на очень быстрый рост различных вредоносных программ. Сначала они были весьма безобидными: показывали рекламу, воровали деньги путем рассылки платных смс, блокировали работу устройства (баннеры-вымогатели), но из-за стремительного роста Android, вредоносные программы становились все более изощренными и опасными. К примеру, развитие интеллектуальных помощников, которые могут совершать покупки в интернет магазинах, при помощи вредоносной программы можно похитить денежные средства, делать фото, включать микрофон, а также повлиять на работу всех функций, которые может использовать система. Например, была ситуация, когда система “умный дом” была заряжена трояном-вымогателем. Уязвимость в приложении “умного” термостата позволяла дистанционно поднимать или опускать температуру, а после этого требовала выкуп за возвращение контроля. Данная уязвимость, в случае отсутствия хозяев либо их сна, может привести к таким негативным последствиям, как различные болезни от переохлаждения или даже смерти домашних животных.

Для удаленного управления различными устройствами используются приложения на смартфоне. Чаще всего в таких приложениях отношение к безопасности посредственное и наиболее распространенными уязвимостями являются:

— Нет проверки целостности кода. Данная уязвимость позволяет добавлять свой вредоносный код или целиком менять различные функции в приложении.

— Нет обфускации. Это значит, что при декомпиляции приложения код никак не защищен и злоумышленник может просмотреть как реализовываются те или иные функции, что позволит потенциально добраться до дыр в системе.

— Нет технологии обнаружения root. Права суперпользователя позволят вредоносным программам полностью взять контроль над устройством.

— Нет зашифрованные логины и пароли. Доступ к внутренним файлам приложения позволит получить персональные данные пользователей.

— Нет защиты от троянов, которые помещают свое окно поверх окна приложения. Распространенная уязвимость, которая позволяет обманом вынудить ввести логин и пароль через фишинговое окно.

Например, автомобиль Tesla уже неоднократно дистанционно взламывался с практически полным контролем над мультимедийной системой автомобиля. Но такие взломы обычно строятся на основе компрометации бортового ПО самого автомобиля. Но специалисты решили атаковать официальное Android-приложение Tesla. Пользователь, после установки приложения, должен ввести имя пользователя и пароль, после которого формируется токен OAuth. Исследователи выявили, что токен хранится в незашифрованном виде в директории sandbox. Получив токен уже доступен ряд некоторых действий над автомобилем, однако одного токена недостаточно для запуска двигателя. Для того, чтобы получить полный доступ к автомобилю можно установить вирус с root доступом, который скопирует данные на необходимый адрес. Специалисты пошли

другим путем: изменили код приложения и предложили его скачать за бесплатный ужин в ресторане. Приложение сразу имело root доступ и сразу же отправляло учетные данные владельца авто на свой сервер. После этого, имея доступ к приложению, можно совершать ряд действий: заводить двигатель без ключа, открывать двери, отслеживать местоположение автомобиля и так далее. Для того, чтобы защитить свой автомобиль от взлома необходимо пользоваться официальным приложением производителя, регулярно обновлять ПО и систему, не подключаться к незнакомым wifi сетям и быть предельно осторожными при вводе учетных данных.

Таким образом, безопасность Android-приложений должна обеспечиваться в полном объеме, так как от этого можно потерять не только личные данные, но и получить различные травмы, поскольку приложения применяются уже практически во всех сферах жизни современного человека.

Список использованных источников:

1. Kaspersky.ru [Электронный ресурс] — Режим доступа: <https://www.kaspersky.ru/resource-center/threats/android-mobile-threats>. Дата доступа: 02.03.2018.
2. Першин Александр. Безопасность мобильных технологий в корпоративном секторе /Александр Першин — М: 2015.

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В ОБРАЗОВАНИИ

*Институт информационных технологий БГУИР
г. Минск, Республика Беларусь*

Беликов А.С. Агапкин Л.М, Чучвал А.Ю, Мирончик А.Н.

Бакунова О.М. – ст. преподаватель каф. ИСиТ, м.т.н.

Современное образование уже как множество лет остается неизменным на фоне стремительных изменений в различных сферах деятельности современного общества. Справедливо сказать, что темпы эволюции образования отстают от скорости развития общества. Получения актуального и качественного образования будущими специалистами является одной из главных проблем в обществе. Однако вопросы оценки качества образования остаются почти не исследованными, при этом большинство вузов понимает, что без создания внутренней информационно-аналитической системы проблему качества образования не решить. Эти вопросы рассматриваются в настоящей работе.

Как правило, рост контингента студентов и сотрудников, быстрое развитие вуза, непрерывно увеличивающийся объем информации в различных подразделениях – все эти факторы становятся предпосылками для создания собственной информационной системы в вузе. Также немаловажным условием является наличие общеуниверситетской компьютерной сети и программных систем по сбору информации о преподавателях, студентах, научной деятельности вуза, материально-технической базе, востребованности выпускников на рынке труда.

Однако в наше время уже недостаточно просто создание информационных подсистем по сбору информации, связанной с деятельностью вуза. Появляется необходимость в создании эффективных и производительных средств анализа полученной информации для оценки качества образования. Только так возможно преобразовать информационную систему вуза в информационно-аналитическую систему. При построении таких систем, вуз, с точки зрения проблемы качества образования, получает возможность оценивать: качество преподавательского состава; качество полученных студентами знаний; состояние материально-технической базы вуза; уровень конкурентоспособности специалистов на рынке труда.

Одним из способов решения данной проблемы, является внедрение в образовательный процесс всевозможных технических средств, обучающих систем, использование Internet-обучения.

Разработка обучающих систем в настоящее время очень популярный и интенсивно развивающийся вид научной деятельности, из-за возобновившегося интереса к использованию на практике технологий искусственного интеллекта, а также интенсивного развитие Internet-технологий, которые дают возможность инженерам использовать новые производительные средства разработки, которых не было ранее. Популярность в этой области научных исследований привела к тому, что в настоящее время существует большое количество научных исследований по данной теме, разработаны сотни обучающих систем, реализованы уникальные подходы и методологии.

В настоящее время функционируют множество систем для создания обучающих структур, среди которых лидирующие места занимают искусственные нейронные сети.

Нейронные сети - вычислительные структуры, которые моделируют простые биологические процессы, обычно ассоциируемые с процессами человеческого мозга.

Основные свойства нейронных сетей:

- обучение сети, обобщение;
- параллелизм;
- представление информации в распределенном виде и дальнейшие вычисления;
- адаптивность;
- умеренное энергопотребление;
- контекстуальная обработка информации;
- обработка ошибочных ситуаций.

Искусственные нейронные сети позволяют решить такие проблемы как: классификация образов, кластеризация/категоризация (без учителя), аппроксимация функций, задачи с предсказанием и прогнозом, всевозможные оптимизации,

задачи, связанные с памятью, адресуемой по содержанию, распознавание образов, различного рода управленческие задачи.

Классификация нейронных сетей представлена на рисунке 1.

Для оценки качества образования в ВУЗе предлагается использовать две модели:

– компетентностная модель – для оценки качества знаний выпускников, включает в себя 6 групп компетенций: общие, личные для направления, личные для всех специальностей, дополнительные компетенции, специальные компетенции (практические), специальные компетенции (теоретические),

– карту сбалансированных показателей (BSC) – для оценки качества организации образовательного процесса.

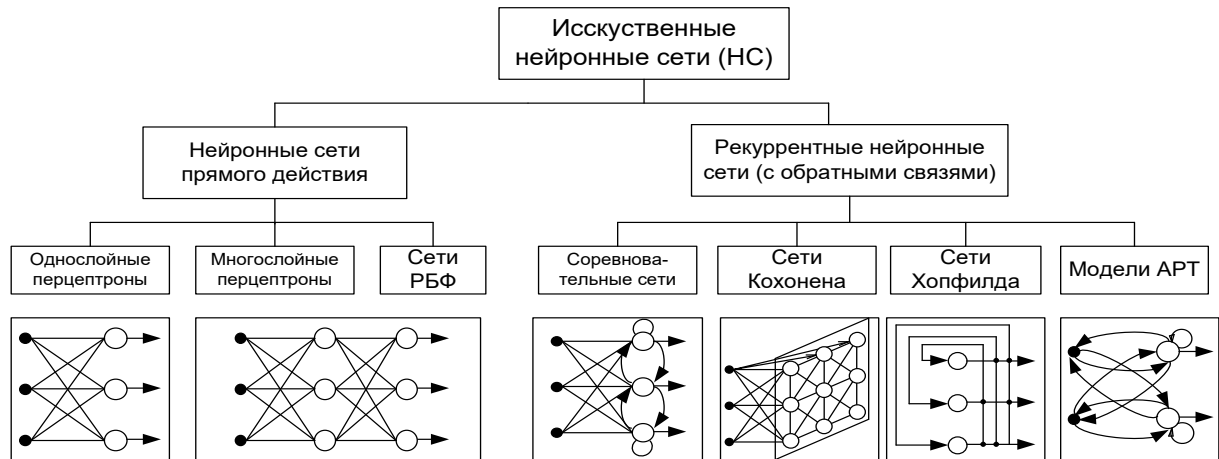


Рисунок 1 – Классификация нейронных сетей

Данные модели позволят проанализировать такие ключевые направления деятельности ВУЗа как:

– ведение учебного процесса (методы и формы обучения, стимулирование и мотивация, цель и результат);

– управление административным и профессорско-преподавательским составами;

– научная, инновационная, финансовая и маркетинговые деятельности;

– обеспечение учебно-методическими материалами;

– социальное и техническое обеспечение.

Таким образом, создание информационно-аналитической системы является актуальной задачей, имеющей важное значение в управлении и развитии высшего учебного заведения. Использование современных подходов в информационных технологиях анализа данных позволит вузу эффективно управлять своей деятельностью. Также стоит отметить, что построение таких систем требует значительных трудозатрат и может быть реализовано только коллективом высококвалифицированных разработчиков при постоянной поддержке со стороны руководства вуза.

Список использованных источников.

1. Хайкин, Саймон. Нейронные сети. Полный курс. / Саймон Хайкин / - Издательский дом "Вильямс", 2006., - 486 - 488 с.
2. Каллан, Роберт. Основные концепции нейронных сетей/ Роберт Каллан / - Издательский дом "Вильямс", 2001., - 928 с.
3. Рассел С. Искусственный интеллект. Современный подход/ С. Рассел, П. Норвиг / – БХВ - Петербург, 2013., - 480 с.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ РАСПОЗНАВАНИЯ ЛИЦ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ

Институт информационных технологий БГУИР
г. Минск, Республика Беларусь

Булахов Е.В.

Шпак И.И. – зав. кафедрой ПЭ, к.т.н., доцент

В докладе рассмотрен комплекс вопросов, связанных с созданием, тестированием и внедрением систем распознавания лиц на основе разработанного программного средства.

Распознавание лиц – практическое приложение теории распознавания образов, в задачу которого входит автоматическая локализация лица на фотографии и, в случае необходимости, идентификация персоны по лицу [1].

Задача идентификации и распознавания лиц – это одна из первых практических задач, которая стимулировала становление и развитие теории распознавания и идентификации объектов.

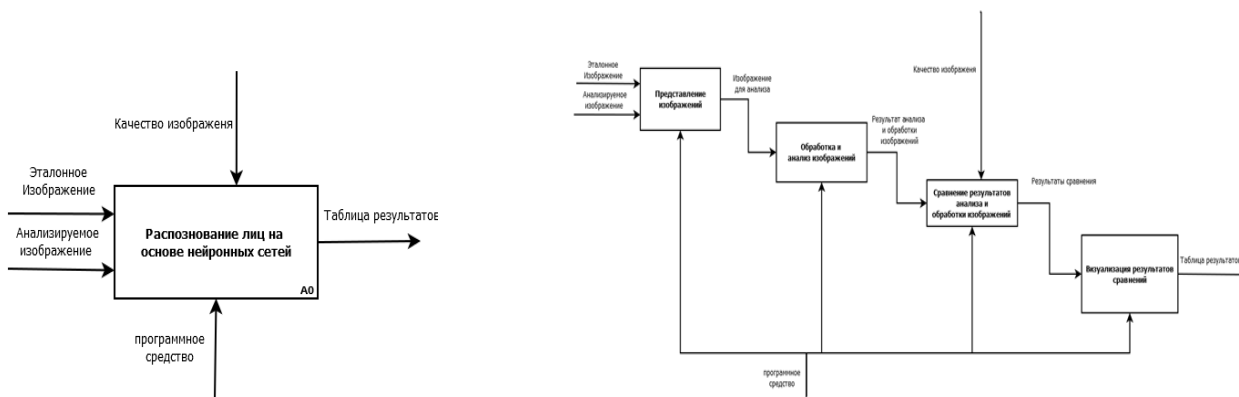
Основными проблемами задачи идентификации и распознавания лиц являются: проблема освещенности, проблема положения головы [2]. И отдельные проблемы, связанные с методом, основанным на нейронной сети, в частности трудности на этапе выбора архитектуры сети (количество нейронов, слоев, характер связей). Проблемы математического характера, связанные с обучением попадание в локальный оптимум, выбор оптимального шага оптимизации, переобучение [3]. Добавление нового эталонного лица в базу данных требует полного переобучения сети на всем имеющемся наборе (достаточно длительная процедура, в зависимости от размера выборки от 1 часа до нескольких дней).

Создаваемое программное средство должно позволять пользователям произвести идентификацию лиц на анализируемом изображении.

Главной особенностью программного средства является набор готовых классов, процедур, функций, структур и констант, предоставляемых программным средством для использования во внешних программных продуктах.

Функционал программного средства распознавания лиц на основе нейронных сетей будет выглядеть так: при запуске программного средства пользователю будет дана возможность загрузить выбранное им эталонное и анализируемое изображение. Далее пользователю необходимо запустить процесс обработки изображения. После запуска процесса обработки программное средство автоматически выделит и сегментирует изображения, полученные сегменты будут обработаны различными видами фильтров. Результат обработки программное средство будет отправлять в нейронную сеть, где и произойдет сам процесс распознавания лиц, с последующим выводом таблицы сравнения на экран.

На рисунке 1 отображена функциональная структура объектов программного средства, производимые ими действия и связи между этими действиями.



Контекстная диаграмма распознавания лиц

Диаграмма декомпозиции Процесса распознавания лиц

Рисунок 1 - Процесс распознавания лиц на основе нейронных сетей.

Одним из моментов разработки стало построение модели нейронной сети. На рисунке 2 представлена разработанная модель нейронной сети.

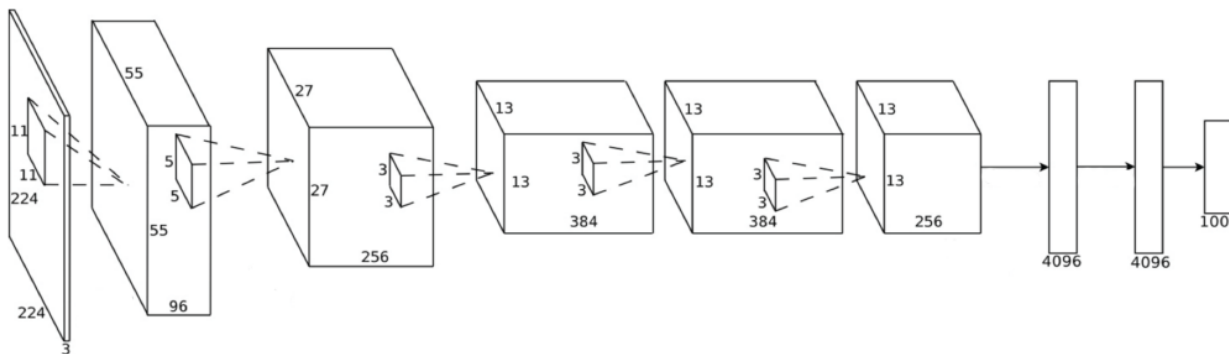


Рисунок 2 - Модель нейронной сети

В докладе процессы разработки программного средства, его тестирования и особенности внедрения рассмотрены более подробно.

Список использованных источников:

1. Хайкин. Neural Networks: A Comprehensive Foundation / Саймон Хайкин. – «Вильямс», 2016. – 1, 56, 77, 1104 с.
2. Goodfellow, Bengio, Courville. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. – MIT Press, 2016. – 552, 773 с.
3. Ciresan, Meier, Masci, Schmidhuber. Multi-column Deep Neural Network for Traffic Sign Classification. Neural Networks / Ciresan, Meier, Masci and Schmidhuber - MIT Press, August 2012, 333 – 338 с.

СОВРЕМЕННЫЕ WEB-ПРИЛОЖЕНИЯ

*Институт информационных технологий БГУИР
г. Минск, Республика Беларусь*

Бурак Д. И.

Бакунова О.М. - ст. преподаватель каф. ИСиТ, м.т.н.

Бакунов А.М. - ст. преподаватель каф. ИСиТ, м.т.н.

Калитеня И.Л. - ассистент каф. ИСиТ, м.т.н.

Хмелевская А.Л. - ассистент каф. ПЭ

В работе рассматриваются различные веб-приложения и средства их использования.

Web-приложение - клиент-серверное приложение, в котором клиент взаимодействует с сервером посредством браузера, а за сервер отвечает — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами.

Для функционирования web-приложения используется HTTP Server. Наиболее популярными являются Apache HTTP Server и nginx.

Nginx – это простой, быстрый и надёжный сервер для статических веб-сайтов и как прокси-сервера перед динамическими сайтами.

Apache – это модульный HTTP сервер для динамических сайтов. Поддерживает различные языки программирования реализована за счёт подключаемых модулей.

В высоко нагруженных серверах используются их комбинация nginx и apache. Nginx используется для отдачи статических данных пользователю (картинки, видео, музыка и тд). Apache используется для динамического содержимого.

Web-приложение разделяется на Front end и back end. Front end – предоставляет графический интерфейс для взаимодействия пользователя с приложением. Back end – отвечает за функционирование бизнес-логики.

Front end – представляет собой интернет страницы разработанные на html, css и javascript. Все взаимодействия пользователем с приложением осуществляется через Front end. Существуют множество методов и принципов построение графической части. Самый современный метод – это перенос формирование конечных страниц с сервера на клиенте. Вследствие возросшей вычислительной мощности компьютеров конечных пользователей, стало возможно быстро формировать динамические страницы на стороне пользователя. Данный подход позволяет разгрузить сервер от формирование динамических страниц. Также он позволяет организовать плавный переходы по сайт без перезагрузки страницы.

Back-end – определяет логику работы сайта, обеспечивает функционирование динамического содержимого сайта. Раньше самым популярным технологией для back-end было PHP. Так-как он позволял быстро и просто создавать динамические сайты. На сегодня он также является популярным технологией, но используется только фреймворки написанные на нем. Такие как Laravel, CodeIgniter, Yii, Symfony. Так же существует фреймворки написанных на других языках - Java Spring, C# ASP.net mvc, python Django.

Список используемых источников

1. Марко Беллиньясо. Разработка Web-приложений в среде ASP.NET 2.0: задача — проект — решение = ASP.NET 2.0 Website Programming: Problem - Design - Solution. — М.: «Диалектика», 2007. — С. 640.
- 2.2) Олишук Андрей Владимирович. Разработка Web-приложений на PHP 5. Профессиональная работа. — М.: «Вильямс», 2006. — С. 352.

ИНФОРМАЦИОННОЕ МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ОРГАНИЗАЦИЙ И СЛУЖБ ЖИЛИЩНО-КОММУНАЛЬНОГО ХОЗЯЙСТВА

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Бурак Д.Б.

Охрименко А.А. – канд. техн. наук, доцент

В докладе рассмотрен комплекс вопросов, связанных с созданием и внедрением современных информационно-коммуникационных технологий и мобильных приложений для нужд организаций и служб ЖКХ различных форм собственности.

Повышение качества услуг, предоставляемых органами государственной власти, органами местного управления и самоуправления гражданам и бизнесу на современном этапе неразрывно связано с совершенствованием и активным применением средств информационно-коммуникационных технологий в рамках модернизации системы государственного управления, создания электронного правительства. Это позволяет сэкономить бюджетные средства, повысить эффективность и управляемость социальных процессов, увеличить скорость принятия решений, а также способствует борьбе с коррупцией [1].

Сегодня у каждого человека всегда при себе есть мобильный телефон, планшет, смартфон и т.п. Банки, инвестиционные и страховые компании, интернет-магазины, сервисы заказа такси и компании общественного транспорта предлагают своим клиентам мобильные приложения, которые позволяют повысить эффективность принятия решений, находить новые каналы коммуникаций. Мобильные приложения значительно упрощают деятельность организациям различной формы собственности. Компании переходят на поддержку клиентов через чаты, получают возможность одновременно вести диалог с несколькими клиентами, эффективно решать вопросы по их обслуживанию, оптимизировать персонал и т.д.

В связи с реформами в системе жилищно-коммунального хозяйства городские власти стремятся внедрять ИКТ и упрощать взаимодействие населения с коммунальными службами посредством разработки и внедрения различных мобильных приложений и сервисов для взаимодействия с организациями ЖКХ, которые должны, например, упростить связь граждан с организациями ЖКХ и мастерами из ЖЭУ [2].

В докладе представлены результаты разработки мобильного приложения по сервису услуг в области ЖКХ, позволяющего существенно упростить и автоматизировать процессы взаимодействия заказчиков и исполнителей услуг ЖКХ, а также оценить эффективность работы сотрудников.

Предлагаемое мобильное приложение отличается удобным интерфейсом, позволяет быстро произвести заказ «в один клик»; производить регистрацию, учет и контроль заявок на ремонт и обслуживание; повысить оперативность исполнения заданий и получить наглядное подтверждение выполненных работ; формировать отчет о выполненных заявках; производить оперативный поиск и предоставлять информацию о состоянии заявок; контролировать выполненные работы; осуществлять печать заявок; сформировать и развивать клиентскую базу; значительно сократить время взаимодействия с клиентами и др.

Вид интерфейса мобильного приложения на разных этапах работы представлен на рисунке 1.

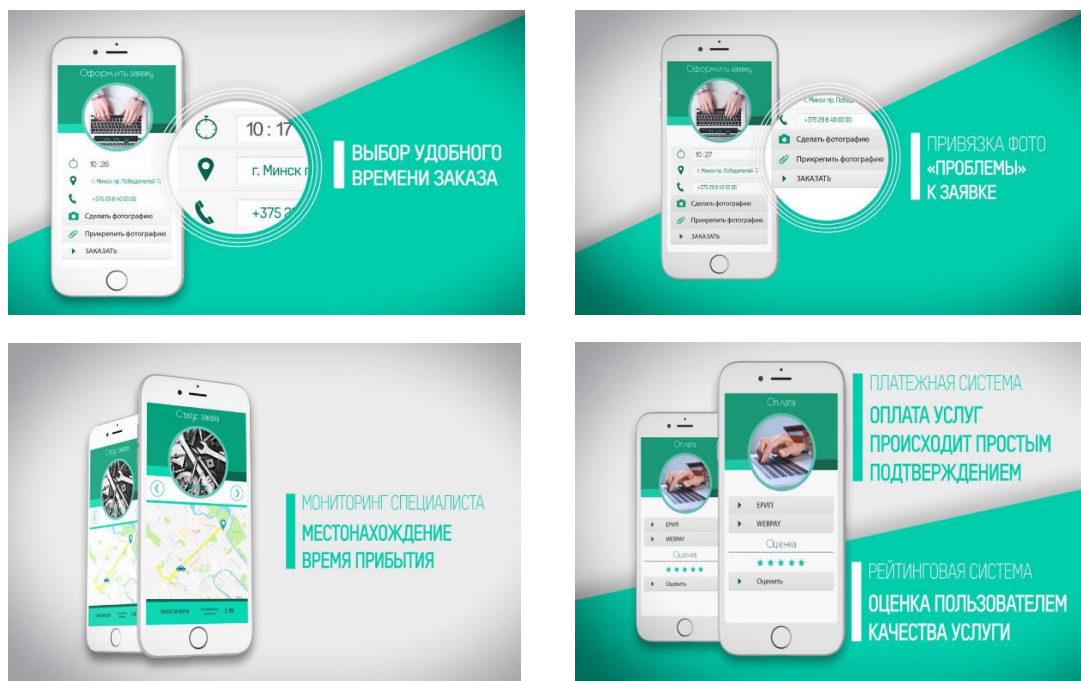


Рисунок 1. Вид интерфейса приложения на различных этапах работы

Существенным преимуществом предлагаемого приложения является его интеграция с автоматизированной информационной системой единого расчетного и информационного пространства (АИС ЕРИП).

Список использованных источников:

1. Охрименко, А.А. Развитие государственных услуг в Беларуси в электронном виде / А.А. Охрименко, И.П. Сидорчук, А.А. Григорьев // Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2015): сб. докл. XIV Междунар. конф., Минск, 19 нояб. 2015 г. / ГНУ ОИПИ НАН Беларуси.- Минск: ОИПИ НАН Беларуси, 2015. - С. 146-152.

2. Приложение «Мобильный мастер», переадресовывающее заявки граждан хаус-мастерам, разработают для ЖКХ Минска // [Электронный ресурс]. – 2018. – Режим доступа: <http://pravo.by/novosti/obshchestvenno-politicheskie-i-v-oblasti-prava/2018/january/27318/> – Дата доступа: 04.04.2018.

МЕТОД ИЗОФОТОМЕТРИИ ФУНКЦИИ РАССЕЯНИЯ ТОЧКИ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Венско А.В.

Пачинин В.И. – зав. кафедрой ИСиТ, к.т.н., доцент

Для устранения недостатков традиционных методов определения ФРТ (функции рассеяния точки), сформированной при работе реальной (изготовленной) оптической системы, созданы методы фотографической изофотометрии. Эти методы позволяют регистрировать перепады освещенности в пятне рассеяния более пяти порядков, строить кривые распределения освещенности в любом сечении, а также топограмму распределения освещенности, вычислять ФКЭ и ФПМ

Метод фотографической изофотометрии основан на получении серии фотоснимков пятна рассеяния с переменным временем экспозиции, то есть получении совокупности фотометрических сечений, соответствующих различным уровням равной освещенности. Каждое фотометрическое сечение формируется как фигура с четким контуром, Линия контура называется изофотой. Таким образом, изофота есть геометрическое место точек имеющих различные пространственные координаты и равное значение уровня относительной освещенности. Формирование изофоты исследуемого оптического изображения осуществляется благодаря применению приемника изображения, обладающего световой характеристикой (или функцией преобразования ФП) типа "импульс" (рис. 1).

Для получения системы изофот (изофотограммы) исследуемого распределения освещенности необходимо использовать ФП типа "гребенка" (по латински - COMB). Эта функция имеет вид ряда равномернорасположенных импульсов (рисунок 1) [1].

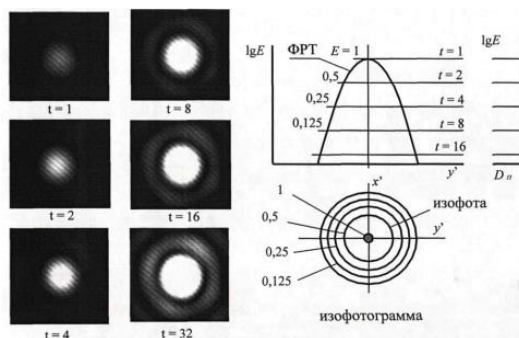


Рисунок 1 – Схема метода изофотометрии с переменным временем накопления и вид изофотограммы

Для получения ФП такого вида выполняется регистрация данного оптического изображения в виде серии последовательных кадров при осуществлении ряда экспозиций различной длительности [1, 5] на приемник изображения с ФП типа "импульс". Удобно этот ряд экспозиций располагать по шкале длительностей, значения которых определяются степенным законом возрастания:

$$t_i = 2^i, \quad (1)$$

где t - относительная продолжительность экспозиции.

Равномерная шкала логарифмов экспозиции позволяет последовательно "наводиться" импульсной характеристической кривой на равномерный ряд фотометрических сечений, расположенных по логарифмическому закону изменения интенсивности.

Таким образом, при использовании приемника изображения с импульсной функцией преобразования,

изменяя время накопления на приемнике изображения, достигаем эффекта перемещения импульсной функции преобразования в новые положения вдоль оси IgE .

Список использованных источников.

1. Родионов. С.А. Основы оптики / С.А. Родионов // СПб.: ИТМО. 2000 г.

ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ПРИЛОЖЕНИЙ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Вербило А.И.

Савенко А.Г. - ассистент каф. ПЭ, м.т.н.

Дополненная реальность использует визуальный ряд реального мира, дополняя его виртуальными предметами и свойствами. В настоящее время эта технология получила большое распространение в самых различных областях применения: от развлечений и мобильных устройств до исследовательского процесса и военной промышленности.

Одно из самых известных устройств – Google Glass. Проект был запущен в 2013 году, в 2015 было объявлено о приостановке производства. В тоже время был анонсирован продукт от Microsoft – HoloLens.

Из менее специализированных устройств существует поддержка данной технологии и на смартфонах. Google и Apple выпустили SDK, позволяющие разработчикам взаимодействовать с камерами смартфонов и обрабатывать изображения внутри существующих систем.

Помимо смартфонов компания Google занимается разработкой различных сервисов. Один из таких сервисов – Google Map. Для создания полной карты мира Google использует снимки со спутников и склеивает их. В дополнение к снимкам спутника Google запускает машины с камерами по миру. И в результате обработки всей информации появляется возможность посмотреть город, улицу и даже дом, но только снаружи, почти в актуальном состоянии [1].

При данном подходе, кроме огромной сложности и объемов расчетов, существуют проблемы связанные со сбором информации и ее актуализации.

Концепцию решения проблемы сбора и актуализации информации можно подсмотреть у сервисов построения дорожных пробок на картах. Базовая логика такого сервиса основана на том, что приложение собирает информацию о том, где, как и с какой скоростью движется пользователь, и на основании этого, можно построить предположения о пробках. Чем больше пользователей, тем более высока точность прогноза о том, что на данном участке находится именно пробка, а не просто остановился случайный водитель [2].

Используя знания о том, как строится карта пробок, возможности дополненной реальности и тот факт, что в теории у каждого пользователя есть устройство, пусть и не специализированное, но с возможностью работать с дополненной реальностью, можно построить систему, которая будет поддерживать существующие карты в актуальном состоянии на основании информации, которые предоставляют сами пользователи системы.

Используя возможности современных технологий, можно создать устройство или улучшить существующее, таким образом, чтобы устройство собирало информацию о местоположении устройства, направления зрения, и снимки перемещения пользователя, об удаленности объектов на снимке. И комбинируя эти данные с полученными данными от других пользователей, сервис обновлял информацию карт.

В общем случае принцип работы сбора информации будет следующий: во время перемещения пользователя по городу устройство собирает такую информацию, как географические координаты, направления взгляда пользователя, снимок местности, и удаленность определенных точек на изображении от пользователя. Собрав всю эту информацию от одного пользователя можно построить не только маршрут пользователя, но и визуализировать посещенную местность, построить приблизительную карту местности, собранная информация об удаленности точек позволяет построить трехмерную проекцию посещенной местности.

В случае с одним пользователем на визуализации могут появляться своеобразные артефакты, которые не являются частью карты, например, люди и машины.

Чем больше пользователей пользуется системой, как и в случае со сбором информации о пробках, тем более точные данные будут находиться в системе. Пешеходы, машины и другие не статические городские объекты на снимках от разных пользователей не будут занимать одно и то же положение и как следствие такие динамические объекты можно считать артефактами и не вносить в объекты карты. Даже в ситуации, когда, например, парковка забита весь рабочий день, и в результате обработки дневного потока информации на карте появятся объекты машин, то вечерние обновления покажут, что машины на парковке объект не статичный, и их можно не учитывать в обработке данных.

К проблемам, которые можно выявить на этапе теории, можно отнести объем данных, в чрезмерно людных местах объем данных будет избыточен, а в недостаточно людных, наоборот, информации о местности может не хватать.

Список использованных источников:

1. How images are collected [Электронный ресурс] – Режим доступа: <https://support.google.com/earth/answer/6327779> - Дата доступа: 12.03.2018.
2. Как работают Яндекс.Пробки [Электронный ресурс] – Режим доступа: <https://yandex.ru/company/technologies/yaprobki>. – Дата доступа: 12.03.2018.

ПРОГРАММНОЕ СРЕДСТВО НА ПЛАТФОРМЕ ANDROID ДЛЯ РЕГЛАМЕНТИРОВАНИЯ ПОСТУПАЮЩИХ ВЫЗОВОВ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Воробей К.П.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Коренская И. Н. – ст. преподаватель каф. ИСиТ*

Объектом исследования является программное обеспечение для мобильной ОС Android выполняющее функцию управления входящими вызовами. Целью проекта является разработка программного обеспечения на платформе Android для регулирования входящих вызовов. Основной эффект – сократить время, затрачиваемое на решение повседневных задач, уменьшая затраты времени на обработку входящих вызовов.

Для решения поставленных целей реализован функционал по обслуживанию поступающих телефонных вызовов и формирования гибкого расписания и правил обслуживания в виде чёрных и белых списков. Качество оценивается проведением тестирования и перехвата исключительных ситуаций, получаемых при выполнении программного средства.

Входная информация по задаче определяется как данные, поступающие на вход задачи и используемые для ее решения. Входной информацией служат первичные данные документов ручного заполнения, информация, хранящаяся в файлах базы данных, входные сигналы от датчиков.

К документам входного заполнения относятся параметры, добавляемые в расписание, как во временное расписание, так и в регулярное, так же шаблоны sms-сообщений, имена и телефоны для черных и белых списков.

В базе данных хранятся данные ручного заполнения в виде шаблонов sms-сообщений имена и телефоны для черных и белых списков, время временного и регулярного расписания.

Программное средство для регламентирования поступающих вызовов разработано в новейшей среде разработки Android Studio. Android Studio – интегрированная среда разработки продуктов Google, с помощью которой разработчикам становятся доступны инструменты для создания приложений на платформе Android. Подсистема хранения данных основана на системе управления базами данных SQLite.

Разработанное программное средство может работать как на планшетах, так и на смартфонах под управлением операционной системы Android.

На устройстве должны быть выполнены следующие условия:

- операционная система Android 4.1 и выше;
- стандартной приложением для обработки вызовов;
- размер экрана не менее 4 дюймов для комфортной работы.

Разработанное программное средство выполняет следующие функции:

- составление регулярного и временного расписания;
- работа с черным и белым списком;
- обработка входящих вызовов по определённым правилам.

Список использованных источников:

1. Эванс, Э. Предметно-ориентированное проектирование/Э. Эванс – И.Д. Вильямс, 2017. – 448 с.
2. Шилдт, Г. Java 8. Руководство для начинающих/Г.Шилдт – И.Д. Вильямс, 2015. – 720 с.

МОДЕРНИЗАЦИЯ ПРОХОДНОЙ КАМЕРЫ ТЕПЛА И ХОЛОДА

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Германенко И.И., Бахур Н.И., Моженкова Е.В.

Зайкина И.С. – ассистент каф. ИСиТ, м.т.н.

Основным направлением по поддержанию стабильности производства, роста объемов выпуска готовой продукции является обеспечение содержания оборудования в надлежащем состоянии, его своевременного технического обслуживания, ремонта и модернизации. Целью модернизации промышленного оборудования является функциональное усовершенствование

и эффективная адаптация к условиям эксплуатации. Основные требования – экономичность и быстрота модернизации, во избежание долгих простоев оборудования в промышленных масштабах.

Проходная камера тепла и холода (ПК5003) предназначена для замера электрических параметров (статических и динамических) интегральных микросхем (ИМС) при критических температурах (-60/+125 °С) [1]. ИМС, которые проходят через ПК5003, по большей части имеют уровень качества типа военная партия (ВП) и особо стабильная микросхема (ОСМ). Для обеспечения данного уровня качества ИМС в заявленных критических условиях, требуется произвести их проверку именно при таком температурном режиме. А чтобы такие условия воссоздать и поддерживать на протяжении всего времени замера ИМС требуется стабильная работа электрической части камеры. Для модернизации ПК5003 предлагается заменить аналоговую электронику на цифровую, подстроив под реальные условия эксплуатации, значительно уменьшить габариты оборудования и добавить блок диагностики (рисунок 1).

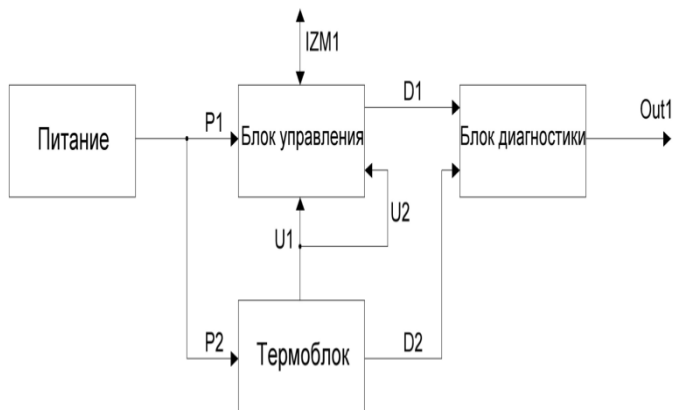


Рисунок 1 – Структурная схема модернизированной ПК5003

ПК5003 состоит из трех основных частей:

- блок питания и пускателей (за время работы ПК5003 наблюдалась их высокая отказоустойчивость, поэтому модернизировать данную часть не требуется);
- термоблок (отвечает за поддержание нужной температуры в камере);
- блок логики (отвечает за работу основных пневмоузлов).

При модернизации блока логики ПК5003 в первую очередь необходимо определить его расположение и расположение его надстраиваемых частей. Рациональный подход к модернизации на данном этапе упростит работу с блоком логики, его настройку и наладку в дальнейшем (рисунок 2).

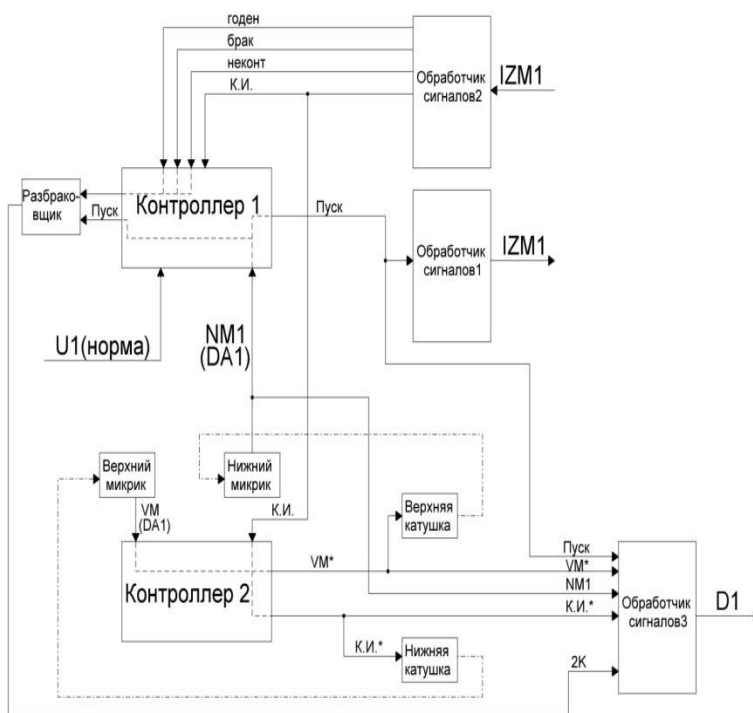


Рисунок 2 – Структурная схема блока логики ПК5003

Далее по значимости подлежит модернизации термоблок (рисунок 3).

Во время работы он сравнивает заданное значение температуры со значением терморпары, которая располагается внутри ПК5003. Если значения различны, то термоблок даёт сигнал на включение того или иного исполнительного элемента до тех пор, пока значения не будут равны. Термоблок можно заменить на измеритель-регулятор технологический ИРТ 5930Н – компактный термоблок (69x87x123) со встроенным цифровым логометром. Модернизация термоблока повысит его отказоустойчивость и упростит его наладку и настройку.

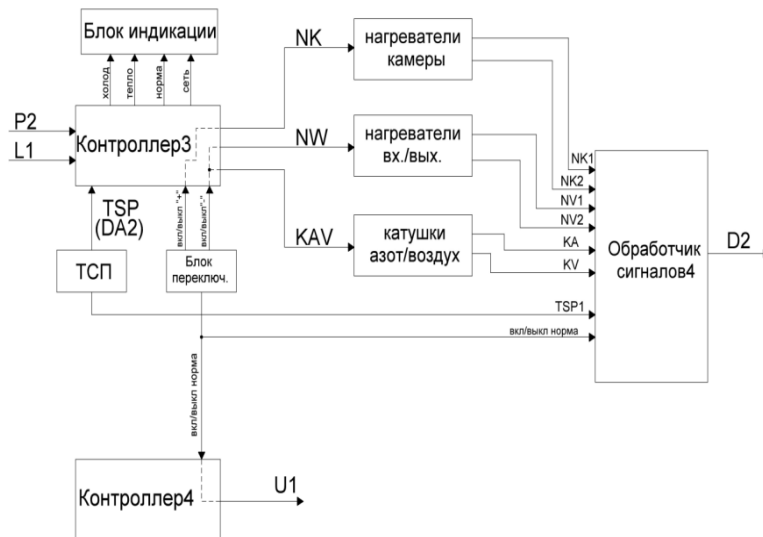


Рисунок 3 – Структурная схема термоблока ПК5003

При проведении модернизации ПК5003 добавляется блок диагностики (рисунок 4). Его суть – упростить наладку и ремонт камеры посредством отображения без дополнительного демонтажа электрических параметров исполнительных элементов ПК5003 и автономного наблюдения за работой камеры, отображения состояния исполнительных элементов, а также автоматического устранения базовых неполадок.

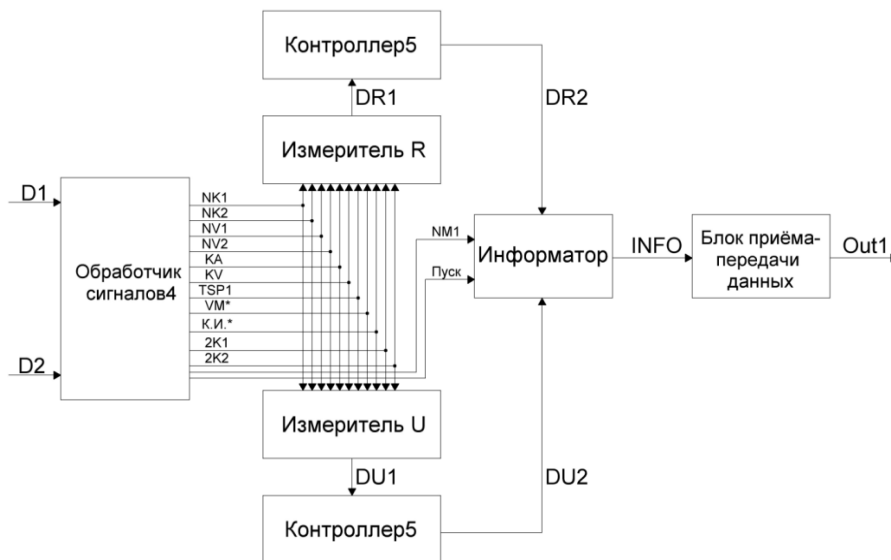


Рисунок 4 – Структурная схема блока диагностики ПК5003.

Для модернизации ПК5003 по соотношению характеристик – функциональные возможности, качество, размер и цена, больше всего подойдет плата-контроллер Arduino Uno R3 [2]. В ее основе лежат легко перепрограммируемые контроллеры ATMeга для простой и удобной настройки под реальные условия эксплуатации ПК5003, а также изменение этих условий.

При модернизации ПК5003 сохранится блочная структура оборудования:

- каждый из блоков будет построен на отдельном контроллере;
- будут разделены на разные платы элементы переключения и разделения сигнала и питания;
- информативно-управляющая панель (с датчиком температур, режимов работы и цифровым логометром) будет демонтируемая.

Модернизация ПК5003 позволит повысить уровень качества тестирования ИМС, уменьшить габариты самого оборудования, а так же сократить время простоя при отказе за счет своевременного обнаружения и замены неисправного блока.

Список использованных источников:

1. Научно-исследовательский институт полупроводникового машиностроения (НИИПМ), Проходные камеры ПК5003; ПК-5003-2; ПК5003-4; ПК-5003-6 ЩЦМ2.708.007.ТО.
2. Омельченко, Е.Я. Краткий обзор и перспективы применения микропроцессорной платформы Arduino/ Е.Я. Омельченко, и др. // Электротехнические системы и комплексы. Магнитогорский государственный технический университет, Россия, г. Магнитогорск, 2013. – С 28 – 34.

СИСТЕМА ИДЕНТИФИКАЦИИ FACE ID

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Гумик В.О.

Образцова О.Н. – доцент каф. ИСиТ, к.т.н., доцент

Доклад посвящен актуальному программному обеспечению способному защитить владельца от кражи его звуковой, текстовой, графической, а также числовой информации. Новым этапом в безопасности современных мобильных устройств встает биометрическая защита, в основе которой лежит уникальность некоторых частей человеческого тела. Например – радужная оболочка и сетчатка глаза, геометрия лица, голос, отпечатки пальцев. Использование процесса биометрической аутентификации является надежной и удобной защитой.

Face ID – сканер 3D формы лица, разработанный корпорацией Apple. Данная система используется в качестве идентификации владельца устройства, для последующей разблокировки или совершения покупок. Технология Face ID является приемником технологии Touch ID, которая, в свою очередь, производила идентификацию по отпечатку пальца.

Принцип работы основан на создании структурной карты лица при помощи камеры TrueDepth. Данная камера проецирует более 30 000 невидимых инфракрасных точек, которые нужны для того, чтобы выделить характерные черты лица (рисунок 1). При первоначальной настройке, необходимо слегка покрутить головой, чтобы система запомнила, как выглядит ваше лицо с разных ракурсов. Считывается, анализируется и сравнивается, на самом устройстве не изображение лица, а его структурная карта.



Рисунок 1. Расположение основных элементов камеры TrueDepth

Face ID встроен в верхнюю часть передней панели iPhone X. Помимо обычного набора датчиков, таких как фронтальная камера, микрофон, динамик, датчик приближения и освещенности, там присутствует инфракрасная камера и два инфракрасных осветителя – проектор точек и заполняющий излучатель (свет от обоих невидим). Заполняющий излучатель необходим для того, чтобы сделать фотографию в условиях слабой освещенности. Проектор точек для проецирования инфракрасных точек на лицо. И в завершение, инфракрасная камера для того, чтобы сделать снимок лица с точками.

Данные с датчиков обрабатываются специально натренированной нейронной сетью, которая способна понять расстояние между спроецированными точками. Сам результат, а именно структурная карта лица, хранится в защищенной области центрального процессора A11 Bionic (там же и хранились данные Touch ID), так называемая Secure Enclave. Доступ к данным происходит при помощи ключа, который доступен только для Secure Enclave. Из этого следует, что данные, которые хранятся в защищенной области – не покидают устройство. Более того, каждый сканер привязан к конкретному процессору и при попытке перестановки датчика из другого устройства, сканер потеряет свою работоспособность. В дополнение к этому, если все же каким-то образом данные были получены, то для их расшифровки необходим специальный ключ, о котором знает только Secure Enclave.

Технология Face ID является самообучаемой, что позволяет ей распознавать лицо владельца при каких-то небольших изменениях (например, естественное старение или очки).

Основным недостатком данной системы является ее энергопотребление. Поэтому, устройство оснащают более ёмким аккумулятором, для поддержания долгой работы.

На сегодняшний день, данная система защиты является наиболее актуальной для безопасного хранения личной информации и совершения платежей. Для сравнения, шанс разблокировать устройство с системой Touch ID, обойдя защиту, является 1 к 50 000, в то время как у Face ID – 1 к 1 000 000.

Список использованных источников:

1.Face ID. [Электронный ресурс].- Режим доступа: https://ru.wikipedia.org/wiki/Face_ID.- Дата доступа: 10.03.2018.

2.Что такое Apple Face ID и насколько это безопасно? [Электронный ресурс]. - Режим доступа: <https://www.kaspersky.ru/blog/apple-face-id-security/18732/>. - Дата доступа: 10.03.2018.

ОБРАЗ СОВРЕМЕННОГО РАЗРАБОТЧИКА ДЛЯ БИЗНЕСА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Гольдюк В. Н.

Бакунова О.М. - ст. преподаватель каф. ИСиТ, м.т.н.

Бакунов А.М. - ст. преподаватель каф. ИСиТ, м.т.н.

Калитеня И.Л. - ассистент каф. ИСиТ, м.т.н.

Хмелевская А.Л. - ассистент каф. ПЭ

В докладе представлен сравнительный анализ требований к специалисту-программисту для различных сфер бизнеса.

В наше время такие люди запросто могут поднять, как и продажи товаров, реализации сервисов в вышеупомянутых разделах бизнеса, так и повышения узнаваемости бренда, и наработка клиентской базы.

Но какие именно нужны разработчики, и какими их видят конечные предприятия?

Чаще всего в крупных, сформировавшихся, компаниях такие специалисты уже сидят для реализации своих бизнес-требований. Есть обязательно уже главный разработчик, который либо спроектировал уже успешную для бизнеса архитектуру ПО, либо участвовал в её создании и становлении лучшим решением для данного предприятия.

Но если затронуть средние и малые сферы бизнеса, то тут вопросы с разработкой ПО уже совсем другие. Чаще всего в таких компаниях либо уже есть специалисты на удаленной поддержке, либо и вовсе обходятся без надлежащего ПО.

Ещё бывает и такой момент в жизненном цикле компании, когда они считают, что им это пока не так необходимо, потому что нету таких крупных оборотов и что не менее важное: нужной для поддержания и инвестирования в ПО выручки.

И чаще всего такие компании разово или, в зависимости от успеха вложения в ПО финансов и ресурсов, больше обращаются либо в студию по разработке ПО, либо к отдельному разработчику. И если первые, всегда знают, что, если они «вложили в разработчика» деньги, то и последний берет на себя обязанность привести в эту компанию прибыль.

Но на практике вырисовывается совсем не такая радостная картина, которую хотели ли бы видеть компании.

Когда обращаются в студии или в компании по разработке ПО, то чаще всего они берут полный комплекс услуг команды по разработке, где они могут, начиная от рассказов дизайнеру о своих предпочтениях по цветовой гамме и палитре и заканчивая согласованию рекламной компании и выделения бюджета на эти цели. Такое решение отлично бы подошло для среднего и малого бизнеса, но часто в большинстве случаев такой спектр услуг предусматривает большое вливание бюджета на начальных этапах и чаще после этого обращаются к самостоятельному разработчику (фрилансер).

Если мы говорим об одном разработчике, то чаще всего, если он достаточно квалифицирован или уже имел опыт в данной предметной области предприятия, сможет объективно четко расписать план действий и сроки реализации программного продукта. Но очень частая проблема таких предприятий в том, что они не хотят рисковать своими ресурсами, выделенными на данную практику и тем самым, отрезают себе доступ к новым возможностям, а значит и автоматически отщепляют некоторую долю клиентов, или просто ЦА (целевую аудиторию). Такие разработчики, помимо технических знаний, должны ещё и обладать следующими навыками, ассоциирующимися или перекрестными с маркетингом, психологией, и умением понимать предпочтения пользователя.

Если говорить про наше время, то мы живем в поколение сервисов, когда пользователю ничего не надо делать, за него уже все придумали и нашли, осталось только взять это. Но не стоит и забывать от том, что сначала этому пользователю надо нарисовать удобный и кратчайший маршрут к его цели. Если говорить о разработчике проще, то он «ведет за руку» нового клиента, и он мотивирует его остаться с ним, и желательно на долго, в его систему.

Но если говорить о разработчике, как об архитекторе нового абстрактного программного продукта, то он должен приложить все возможные силы для реализации этой самой воронки прихода новых клиентов, чтобы она работала стабильно и давала гарантию того, что клиент будет предан именно ей. В наше время разработчик должен быть универсален и быть компетентен в знаниях не только в технических, но и как менеджер по продажам, уметь продавать, или как психолог, который сможет достучаться до самого обиженного клиента. Потому что всегда выгоднее работать и говорить с одним человеком, чем набирать связи с несколькими и

каждому говорить о том, что ему требуется. С одной стороны, от этого может пострадать качество исполнения работ, потому что более узкие специалисты смогут сделать это лучше, но это одновременно и очень затратное дело, не каждый малый или средний бизнес сможет пойти на такие расходы и поэтому готовы работать с одним разработчиком.

Возьмем на примере стандартный необходимыми большинству торговых компаний для удобного распространения услуг или товаров инструмент – интернет-магазин (сайт), он так же может выполнять задачу как корпоративный сайт, который представляет бренд, и чаще всего это лицо компании. В большинстве случаев, как говорится “встречаем по одежке”, но во фразе заменяем вторую часть “проводим по уму” на “и по уму”, вот к такому результату должен прийти разработчик по окончании разработки, чтобы все абстрактное – материализовать, что означает и дизайн хорош и функционал недалеко отстал.

Разработчику необходимо узнать лицо конечного клиента, он должен присутствовать с ним рядом или, ещё лучше, вживиться в его роль, чтобы точно знать, что ему надо и как было бы ему удобнее помочь найти то, что ему необходимо. Если мы говорим о дизайне, то сайт должен иметь приятные на глаз оттенки и сочетания цветов, а также размеры и цвет шрифтов. Потом, после уже дизайна, начинается этап исследования удобного для пользователя взаимодействия интерфейса системы. И если, это все смешать, мы получим очень неплохую “одежку”, но к этому надо ещё и добавить психологию: игра цветов компонентов управления системы. Тут все как в жизни, красный – нельзя, стоять, зеленый – можно ехать. Вот так же с интерфейсом системы, надо помогать клиенту ориентироваться что сейчас он это может сделать, а в другом случае остановить его, что если кнопка зеленая, значит она приведет его в нужное место. Но человеческий фактор никто не отменял, люди ошибаются и надо как можно больше сузить набор возможных ошибок пользователя. Если человек хочет сделать заказ с сайта, то предоставь ему несколько вариантов связаться с тобой, будь это номер телефона чтобы он мог позвонить, или избавьте его от этой процедуры, позвоните ему сами! Но никогда не обещайте ему позвонить за определенное время, лучше скажите правду, что перезвоните в кратчайшие сроки тем самым не обманите его, он будет держать в голове мысль о том, что процесс движется и он нем позаботятся. Все люди работают и все они устают, поэтому чаще всего они что-то ищут уже после заката солнца, когда уже большинство магазинов по закрывались и уже только на следующее утро только откроются, поэтому им необходимо тоже дать возможность, что-то найти и забрать. Такую долю клиентов тоже не стоит забыть, сделайте им удобную форму заказа, которая будет круглосуточно открыта для новых заявок. Упростите её настолько, насколько это возможно, только номер телефона и комментарий, но, когда вы делаете очень подробную форму заполнения информации, что ему надо обязательно вписать свою дату рождения, он устанет и уйдет в другую систему.

Помимо “одежки”, необходим и “ум” – функционал и логичность всей системы в целом. Сюда входят, как и все возможные фильтры при выборе какого-нибудь продукта, так и строка с “умным поиском” сверху, для максимального сокращения пути клиента к цели. Не стоит забывать и про “хлебные крошки” на каждой странице сайта, чтобы он всегда мог без лишних движений вернуться на шаг назад. И, как уже писалось выше про поколение сервисов, что клиенту надо все за него найти, сделайте ему блок, на основе его предыдущего опыта поиска в каталоге, рекомендуемые именно для него товары, но и ни стоит недооценивать и ту возможность, что ему в придачу к этому, можно и ещё что-нибудь порекомендовать, или то, что очень хорошо подходит к данному типу товара, что вместе обычно с ним ещё смотрят и другие товары.

Ну если с дизайном и функционалом уже определились, то необходима и смесь навыков продавца и маркетолога – это продвижение и вручение продукта клиенту. Никогда не стоит забыть про рекламу, любой разработчик, который соглашается конкурировать со студиями или компаниями по разработке ПО, должен знать эти основы рекламы для начального удачного старта проекта. Он должен так иметь навыки в копирайтинге, подготовке рекламных шаблонов, и анализе рынка спроса по всем торговым площадкам и площадка для размещения рекламы. Выявление и анализ для подсчета среднесуточного бюджета и умению разумно оптимизировать рекламу для товаров, потому что никому не хочется платить за воздух, и поэтому отслеживать это, и при отсутствии товара, выключать рекламу.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ «СИСТЕМА ИНФОРМИРОВАНИЯ В ЭКСТРЕННЫХ СИТУАЦИЯХ»

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Гончаров Я.И.

*Пачинин В. И. – зав. кафедрой ИСuТ, к.т.н., доцент
Савенко А.Г. – ассистент каф. ПЭ, м.т.н.*

Основой для реализации задачи по информированию в экстренных ситуациях служит необходимость поддержки людей, попавших в экстренную и опасную для их жизни обстановку, а также обработка их данных о местоположении для помощи им в кратчайшие сроки. Создание сервиса экстренного оповещения на мобильные платформы, который представляет собой разнонаправленное решение, предоставляющее пользователям гибкую систему по организации собственной безопасности и личной неприкосновенности.

Главная задача системы – оповещение пользователей данной системы о находящихся рядом с ними зонах с потенциально опасными для их жизни и безопасности условиями, а также оповещение выбранных пользователем контактов о его местоположении в случае возникновения опасности для жизни.

Для удобства использования мобильного приложения разработан интуитивно понятный пользовательский интерфейс, что непременно улучшит отзывчивость пользователей и уменьшит время на освоение навыков работы с данной системой.

Все требующиеся для работы системы данные хранятся на уровне настроек большого приложения, что позволило отказаться от привычной структуры с использованием СУБД.

Для разработки мобильного приложения был использован фреймворк Xamarin. Xamarin – фреймворк для кроссплатформенной разработки мобильных приложений с использованием языка C#. Он представляет возможность применения любого языка программирования с полным доступом ко всем возможностям SDK платформы и механизму создания UI, получая на выходе приложение, которое имеет высокую производительность.

Разработанное мобильное приложение выполняет следующие функции:

- добавление/изменение/удаление экстренного оповещения;
- создание и редактирование времени отправки экстренного оповещения;
- оповещение о находящимся поблизости зонах с потенциально опасными условиями для жизни и личной безопасности;
- отправка экстренных оповещений выбранным контактам;
- создание и отправка экстренных оповещений на основании маршрута следования.

Список использованных источников:

1. Шарп, Д. Microsoft Visual C#. Подробное руководство / Д. Шарп – СПб.: Питер, 2017. – 848 с.
2. Албахари, Б. C# 7.0. Карманный справочник / Б. Албахари, Дж. Албахари – М.: Вильямс, 2017. – 224 с.
3. Xamarin [Электронный ресурс]. – Режим доступа: <https://developer.xamarin.com/>. – Дата доступа: 18.12.2017.

ПАТТЕРН УПРАВЛЕНИЯ СОСТОЯНИЕМ НА ПРИМЕРЕ VUEX

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Гулис Н.В.

Бакунова О.М., старший преподаватель каф. ИСиТ, м.т.н.

Состояние (от английского state) является поведенческим шаблоном проектирования. Такой шаблон применяется в тех ситуациях, когда для функционирования программы или приложения объекту необходимо изменять свое поведение в зависимости от состояния.

Паттерн (от английского pattern – диаграмма, схема, шаблон) – это схема или принцип решения определенной задачи при проектировании программ. В данной работе будет описан паттерн управления состоянием на примере его реализации на языке программирования JavaScript – Vuex.

Vuex – это паттерн для управления состоянием, а также библиотека для JavaScript-фреймворка «Vue.js». В большинстве случаев он используется при создании одностраничных web-приложений. Он представляет собой централизованное хранилище данных, доступ к которому имеют все компоненты приложения, или так называемый «единственный источник истины». Данные в хранилище изменяются согласно строго определенным правилам. Такой подход позволяет обеспечить предсказуемость изменений.

Хранилище во Vuex включает в себя четыре главных понятия – состояние, мутации, действия и геттеры.

Состояние во Vuex представляет собой JavaScript-объект, содержащий в себе информацию, используемую хранилищем. Изменять состояние напрямую строго запрещается. Все изменения состояния во Vuex производятся через специальные функции – мутации.

Мутации – функции предназначенные для изменения состояния приложения. Каждая мутация включает в себя строковый тип, или имя, и функцию-обработчик, в котором и совершаются изменения состояния, переданного в мутацию аргументом. В мутацию также можно передавать и другой аргумент, так называемую «нагрузку», представляющую собой произвольный объект. Пример объявления мутаций представлен на рисунке 1.

```
mutations: {
  [SET_PROFILE] (state, userInfo) {
    state.user.profile = userInfo;
  },
  [SET_ERROR] (state, error) {
    state.error = error;
    state.pending = false;
  },
  [LOGIN] (state) {
    state.pending = true;
  },
  [LOGIN_SUCCESS] (state) {
    state.user.authenticated = true;
    state.error = null;
    state.pending = false;
  }
}
```

Рисунок 1 – Пример объявления мутаций

Типы, или имена, мутаций можно задать заранее константами либо переменными и вынести в отдельный документ. Такой подход улучшает читаемость кода, облегчает работу других разработчиков и дает представление о всех возможных мутациях для приложения.

Все мутации обязательно должны являться синхронными. Такое требование обусловлено тем, что приложение обязательно должно запоминать свое состояние до и после выполнения мутации. Данное ограничение позволяет отследить и заранее предугадать порядок выполнения мутаций. Это существенно облегчает задачу отладки приложения и позволяет отслеживать все изменения состояния. Для асинхронных операций же предусмотрены действия.

Вызов мутаций в компонентах приложения осуществляется с помощью функции «commit» либо при помощи специальных мапперов, привязывающих мутации к компоненту.

Действия в отличие от мутаций не вносят изменений напрямую в состояние приложения и могут содержать в себе асинхронные функции. Также действия могут использоваться для поочередного вызова мутаций. Асинхронность действий позволяет выполнять в них такие операции как HTTP-запросы или вызов API. Таким образом с помощью действий можно выстраивать цепочки асинхронных запросов, сохраняя полученную информацию в состоянии с помощью промежуточных мутаций. Действия также позволяют вызывать внутри себя другие действия. Вызов действий в компонентах выполняется с помощью функции «dispatch». Пример объявления действия представлен на рисунке 2.

Для получения информации, основанной на состоянии хранилища, во Vuex используются геттеры. Такой функционал необходим, если одна и та же информация из хранилища должна быть отображена в нескольких компонентах. Использование геттеров значительно уменьшает повторение кода и улучшает его читаемость. Примеры геттеров приведены на рисунке 3.

```
actions: {
  login({commit}, credentials) {
    commit(LOGIN);
    return new Promise((resolve) => {
      axios.post('api/login', credentials)
        .then((res) => {
          let token = res.data.token;
          localStorage.setItem('token', token);
          commit(LOGIN_SUCCESS);
          resolve();
        })
        .catch((err) => {
          commit(SET_ERROR, err.response.data.error);
        })
    });
  },
}
```

Рисунок 2 – Пример объявления действия

Хотя хранилище Vuex и представляет собой один большой объект, для удобства разработчика предусмотрена возможность разделять его на модули. Каждый модуль имеет свои собственные мутации, геттеры и состояние. Однако модули могут также получить доступ и к глобальному состоянию хранилища.

Любой модуль можно разделить на подмодули, таким образом структура хранилища являет собой фрактал.

```
getters: {
  authenticated: state => {
    return state.user.authenticated;
  },

  profile: state => {
    return state.user.profile;
  },

  error: state => {
    return state.error;
  }
}
```

Рисунок 3 – Примеры объявления геттеров

Vuex предоставляет собственное расширение для консоли разработчика в браузерах, которое позволяет отслеживать все изменения состояния в режиме реального времени. Помимо отслеживания, оно позволяет откатить любое изменение, а также проверить состояние приложения до и после любого изменения. Данное расширение изображено на рисунке 4.

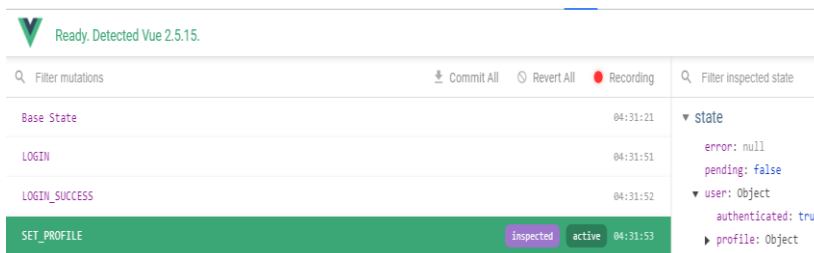


Рисунок 4 – Расширение для браузера «Vue»

Такое расширение значительно ускоряет процесс разработки, облегчает процесс отладки и поиска багов в приложении.

Vueх следует использовать в средних или больших по объему приложениях с большим количеством компонентов, в том числе использующих одинаковую информацию. Использование в небольших приложениях лишь усложнит их логику.

VUE.JS – АНАЛИЗ ФРЕЙМВОРКА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Гулис Н.В.

*Образцова О.Н. – доцент каф. ИСиТ, , к.т.н., доцент
Бакунова О.М. – ст. преподаватель каф. ИСиТ
Бакунов А.М. - ст. преподаватель каф. ИСиТ
Калитеня И.Л., – ассистент каф. ИСиТ*

В настоящее время трудно представить себе разработку актуального web-приложения без использования какого-либо фреймворка. Фреймворк (от английского framework – структура, каркас) – программная платформа, определяющая структуру программной системы, и облегчающая разработку больших программных проектов и их компонентов. В последние годы было написано множество различных фреймворков имеющих разную степень востребованности и качества. В работе будет рассмотрен один из фреймворков для языка JavaScript – Vue.js.

Vue.js был создан бывшим работником Google Эваном Ю и предназначен для разработки пользовательских интерфейсов. Впервые фреймворк был представлен в 2014 году. Он активно набирает популярность, о чем свидетельствует растущее количество его скачиваний, с января по декабрь 2017 года выросшее с 250 тысяч до 1 миллиона. График количества скачиваний Vue.js представлен на рисунке 1. Также о популярности данного фреймворка свидетельствует большое количество крупных компаний, совершивших переход на его использование, таких как Nintendo, GitLab, Baidu.

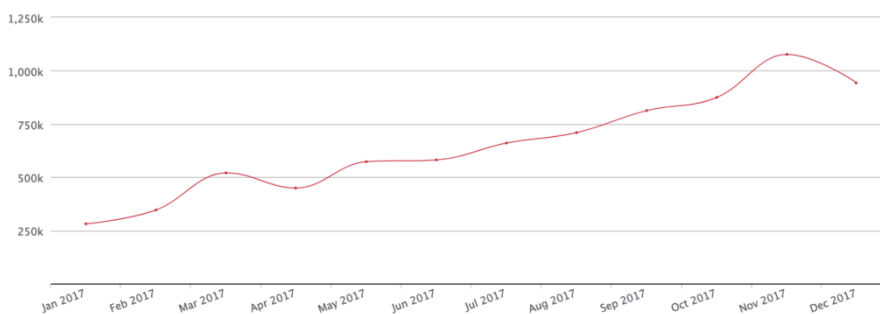


Рисунок 1 – График количества скачиваний Vue.js

Vue.js предоставляется по лицензии MIT для открытого программного обеспечения (другое название «Лицензия X11»). Данная лицензия предоставляет полное право использования, модифицирования, копирования, дистрибуции и публикации программных продуктов, поставленных по этой лицензии[1].

Vue.js обладает высокой производительностью благодаря малому размеру (23кб в архивированном состоянии), а также благодаря использованию виртуального DOM. DOM (от английского Document Object Model – объектная модель документа) – это программный интерфейс, позволяющий скриптам и программам получать доступ к элементам HTML, XML и XHTML-документов. Виртуальный DOM является более легкой копией оригинального. Все изменения первоначально вносятся в копию, затем копия и оригинал сравниваются, и происходит перерисовка только измененных компонентов.

Данный фреймворк зачастую используется для создания одностраничных веб-приложений, однако может быть внедрен и в уже существующий проект на любом этапе разработки или сопровождения. Он

использует принцип реактивности, т.е. приложение загружается в браузере пользователя лишь один раз, а затем переходы между страницами и обновление компонентов происходит без обновления страницы браузера. Также Vue.js предоставляет функционал двунаправленного связывания данных. Таким образом при изменении элемента ввода, изменится также и объект, связанный с ним, и наоборот.

Одной из уникальных особенностей Vue.js является использование так называемых «однокомпонентных файлов»–документов с расширением «.vue». Такой файл включает в себя 3 секции. Первая секция, или шаблон, заключается в специальный тег «template» и содержит в себе HTML-разметку документа, а также выводит данные, хранящиеся в документе. Вторая секция заключается в тег «script» и включает в себя функции, методы, информацию и крюки жизненного цикла компонента. Третья секция заключается в тег «style» и хранит в себе CSS-стили документа, которые могут быть объявлены локально и глобально. Локальные стили будут применены только к документу, в котором они были объявлены, глобальные будут применены ко всему проекту. Пример однокомпонентного файла представлен на рисунке 2.

Однокомпонентные файлы позволяют применять к каждой секции любые пре- и постпроцессоры, такие как Sass, LESS, Pug и другие. Это позволяет уменьшить емкость кода и улучшить его читаемость.

Vue.js можно использовать как в небольших проектах, так и при разработке крупных веб-приложений, так как система компонентов позволяет обеспечить высокую расширяемость проекта. Также его преимуществами являются низкий порог обучения и простой синтаксис. Фреймворк хорошо сочетается со сторонними библиотеками, плагинами, API, серверными приложениями и фреймворками.

```
1 <template lang="pug">
2   div
3     navigation
4     router-view
5     foot
6 </template>
7
8 <script>
9   import Navigation from './layout/Navigation.vue'
10  import Footer from './layout/Footer.vue'
11
12  export default {
13    components: {
14      'navigation': Navigation,
15      'foot': Footer
16    }
17  }
18 </script>
19
20 <style lang="sass">
21   @import '~Global/main.sass'
22 </style>
```

Рисунок 2 – Пример однокомпонентного файла

Список использованных источников

1.Лицензия МТИ/Лицензия X11–Массачусетский технологический институт, 1988.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ДЕЛЕГИРОВАНИЯ ДОСТУПА К ВЕБ-РЕСУРСАМ НА ПЛАТФОРМЕ ANDROID

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Дедович Д.К.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Коренская И. Н. – ст. преподаватель каф. ИСиТ*

Объектом исследования является программное обеспечение для мобильной ОС Android основной функцией которого является ограничение доступа к веб ресурсам. Целью проекта является разработка программного обеспечения на платформе Android для регулирования входящих вызовов. Основной эффект – уменьшение трат на мобильных трафик, для сотрудников предприятия и защита хранимых на смартфоне данных, от различных угроз.

Для решения поставленных целей реализован функционал по ограничению доступа к файлам приложений не входящих в список разрешённых. Запрещён доступ для работы с браузерами по умолчанию. Вместо них можно использовать браузер, который встроен в приложение. Качество приложения оценивается проведением тестирования и корректности работы всех модулей.

К документам входного заполнения относятся параметры, добавляемые администратором в список разрешённых ресурсов для пользователя, так же ограничение работы с приложениями и их доступом к сети интернет.

В базе данных хранятся данные посещённых, пользователем ресурсов с отражением даты и временем их посещения. В это же базе хранятся учётные данные администратора, логин и пароль.

Для разработки программного средства будет использоваться среда программирования Android Studio.

Android Studio – новая и полностью интегрированная среда разработки приложений, для операционной системы Android. Данный продукт призван снабдить разработчиков новыми инструментами для создания приложений, а также предоставить альтернативу Eclipse, являющейся в настоящее время наиболее популярной средой разработки.

Android Studio позволяет увидеть любые визуальные изменения, которые производит разработчик, в реальном времени в приложении.

Для работы приложения требуется устройство с операционной системой Android версии 4.4 и выше. В системе должны быть получены права суперпользователя ROOT, которые предоставляют доступ к необходимым для работы приложения функциям системы.

Для комфортной работы с приложением диагональ экрана должна составлять не менее 3,5 дюймов и разрешение экрана не менее 480x800 пикселей.

Рекомендуемой конфигурацией является размер экрана 5-5,2 дюйма, разрешение экрана 720x1280 пикселей и версия операционной системы Android 5.1.1.

Список использованных источников:

1. Голощапов, А.Л. Google Android: Создание приложений для смартфонов и планшетных ПК / А.Л. Голощапов. - СПб.: БХВ-Петербург, 2013. - 832 с.

2. Коматинени, С. Android 4 для профессионалов: Создание приложений для планшетных компьютеров и смартфонов / С. Коматинени, Д. Маклин, С. Хашими; Пер. с англ. Ю.И. Корниенко [и др.]. - М.: Вильямс, 2012. - 880 с.

ОБРАБОТКА И УЛУЧШЕНИЕ КАЧЕСТВА ИЗОБРАЖЕНИЯ С ПОМОЩЬЮ ПРЕОБРАЗОВАНИЙ ФУРЬЕ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Денис Н.А.

Калитеня И. Л. – ассистент каф. ИСиТ, м.т.н.

В докладе рассмотрен комплекс вопросов, связанных с созданием, тестированием и внедрением программной системы обработки изображений с помощью преобразований Фурье [2].

Тема обработки изображений остается актуальной, к примеру, в современном мире на многих объектах установлены системы видеонаблюдения и используются камеры с низким разрешением. Применяв к видеопотоку таких камер алгоритмы обработки и улучшения качества изображений можно получить улучшенное качество видеопотока [2]. Проанализировав наиболее успешно работающие аналоги, были определены их достоинства и недостатки и учтены в разработке программного средства.

Спектральный анализ является одним из методов обработки сигналов, который позволяет охарактеризовать частотный состав измеряемого сигнала. Преобразование Фурье позволяет представить практически любую функцию или набор данных в виде комбинации таких тригонометрических функций, как синус и косинус, что позволяет выявить периодические компоненты в данных и оценить их вклад в структуру исходных данных или форму функции, а также является математической основой спектрального анализа, которая связывает временной или пространственный сигнал с его представлением в частотной области.

Создаваемое программное средство должно быстро и корректно выполнять обработку изображений, в основе которой лежит метод спектрального анализа, применяя основные алгоритмы фильтрации изображения, которые позволят добиться улучшенного качества изображения, такие как: медианный фильтр (для очистки изображений от лишнего шума), фильтр с заданными параметрами яркости, фильтр с заданными параметрами резкости и др.

На стадии разработки интерфейса пользователя была выделена главная особенность программного средства – обработка и улучшение качества изображения делается в два этапа: загрузка изображения на сервер и нажатие кнопки, которая запустит алгоритм обработки загруженного изображения.

Особенности процесса работы программного средства описываются на рисунке 1.

Исходя из представленной диаграммы формируются основные функции программного средства:

- загрузка исходных изображений;
- обработка и анализ изображений;
- сравнение результатов анализа и обработки изображений;
- визуализация результатов работы программы.

Данное программное средство основано на клиент-серверной архитектуре. Взаимодействие с серверной частью осуществляется по протоколу HTTP. Серверная часть отвечает за бизнес-логику приложения, обеспечивая загрузку и вывод обработанного изображения.

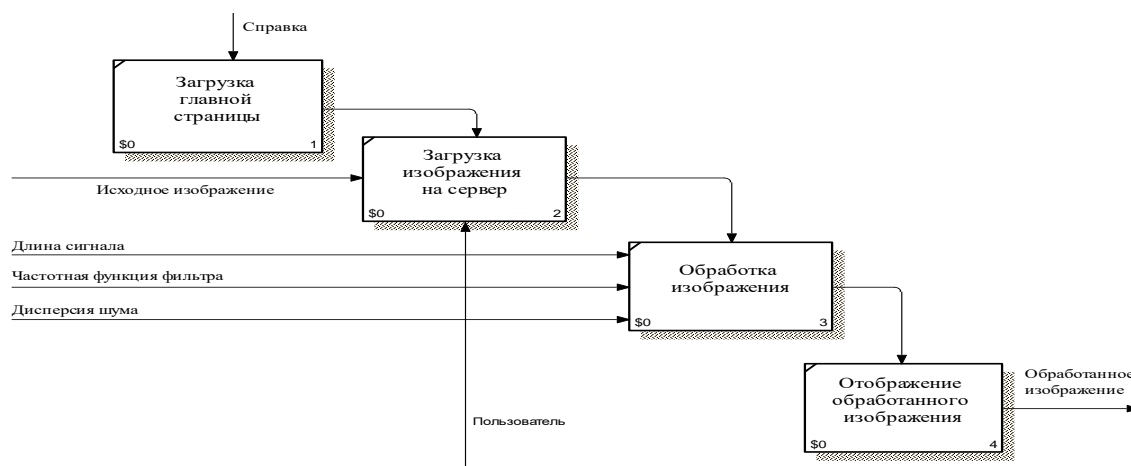


Рисунок 1 - Процесс оформления заявки на доставку груза.

Схема работы программного средства и её структура описана на рисунке 2.

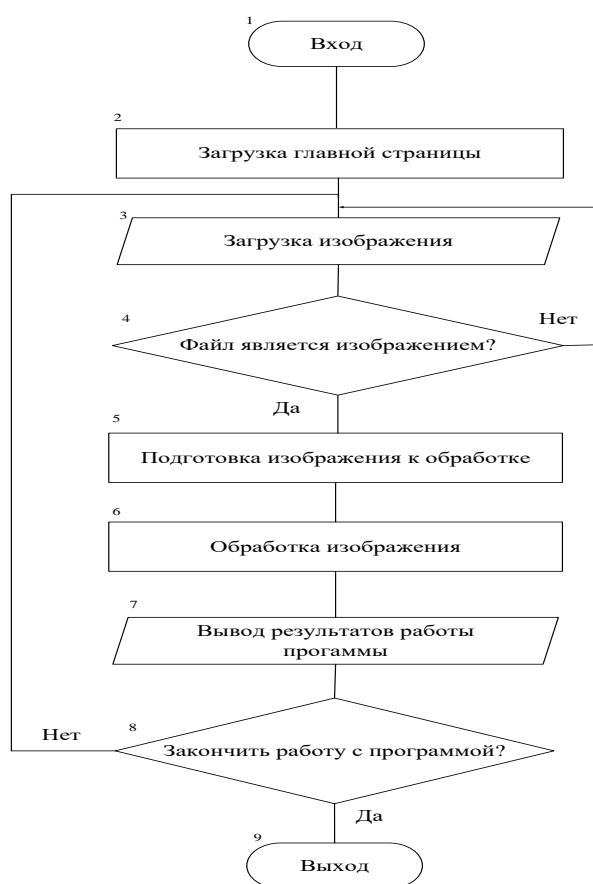


Рисунок 2 - Схема работы программного средства и её пользовательская структура.

Клиентская часть приложения написана на языке JavaScript, использует также фреймворк JQuery. Серверная часть написана на языке C# [1]. Алгоритмы по обработке изображений с помощью преобразований Фурье написаны на языке C++ [3] и представлены в виде динамически подключаемой библиотеки.

Разработанное программное средство положено в основу обработки и улучшения качества изображений с помощью преобразований Фурье.

Список использованных источников:

1. Джеффри, Р. CLR via C# Программирование на платформе Microsoft .NET Framework 4.5 на языке C#/P. Джеффри. - Питер, 2017. – 896 с.
2. Сергиенко, А. Б. Цифровая обработка сигналов/ А. Б. Сергиенко. - Питер, 2002. – 323 с.
3. Страуструп, Б. Программирование: принципы и практика с использованием C++ / Страуструп Б. - Питер 2017. – 543 с.

КРОССБРАУЗЕРНЫЙ ВЕБ-СЕРВИС ДЛЯ РЕАЛИЗАЦИИ АВТОМОБИЛЬНЫХ ЗАПЧАСТЕЙ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Жигуновский А.В.

Шпак И.И. – зав. кафедрой ПЭ, к.т.н., доцент

В докладе рассмотрен комплекс вопросов, связанных с созданием, тестированием и внедрением современных систем реализации товаров через интернет-магазин и их интеграции со сторонними сервисами на основе разработанного веб-сервиса.

Веб-сервисы в современных условиях повсеместно внедряются практически во всех магазинах, с целью сокращения времени поиска товаров и увеличения доходности магазина [1]. Проанализировав наиболее успешно работающие интернет-магазины, определились с их достоинствами и недостатками и учли их при разработке веб-сервиса.

Создаваемый веб-сервис должен обеспечивать потенциальным покупателям возможность доступа к информации о компании, описанию товара и информации о стоимости и наличии этого товара.

Главной особенностью веб-сервисов является набор готовых классов, процедур, функций, структур и констант, предоставляемых веб-приложением для использования во внешних программных продуктах.

Функционал системы поиска товаров на сайте заключается в том, насколько просто покупателю найти на сайте те или иные товары. Это значит, что на сайте магазина обязательно наличие рабочего поиска по сайту, причем такого, который выдает результаты не только при полном совпадении фразы.

Чтобы дать клиенту доступ к магазину не только с компьютера, но и с телефона, необходима мобильная версия сайта. Она должна содержать только самые необходимые функции и не быть перегруженной.

Система обновления товаров предусматривает импорт товаров из файлов поставщиков, которые они отправляют на сервер веб-сервиса.

Важно предусмотреть наличие SSL сертификата — криптографический протокол, который подразумевает более безопасную связь.

В результате улучшаются условия для более полного удовлетворения запросов покупателей с одной стороны и учета этих запросов владельцем интернет-магазина, с другой стороны.

Особенности продажи товаров посредством веб-сервиса [2] с учетом законодательства Республики Беларусь отображает рисунок 1.

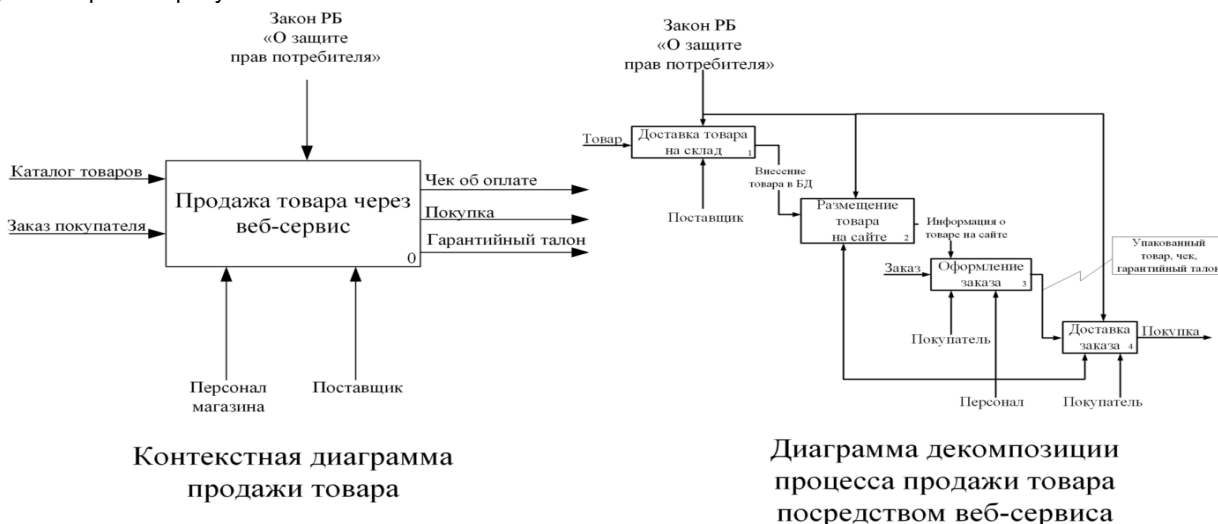


Рисунок 1 - Процесс продажи товаров посредством веб-сервиса в Республике Беларусь

Одним из важнейших моментов разработки стало определение нагрузки на сервер [3]. На рисунке 2 представлена загруженность процессора при выполнении высоконагруженной части операции.

Разработанный веб-сервис положен в основу работы предприятия занимающегося реализацией автомобильных запчастей.

В докладе процессы разработки веб-сервиса его тестирования и особенности внедрения рассмотрены более подробно.

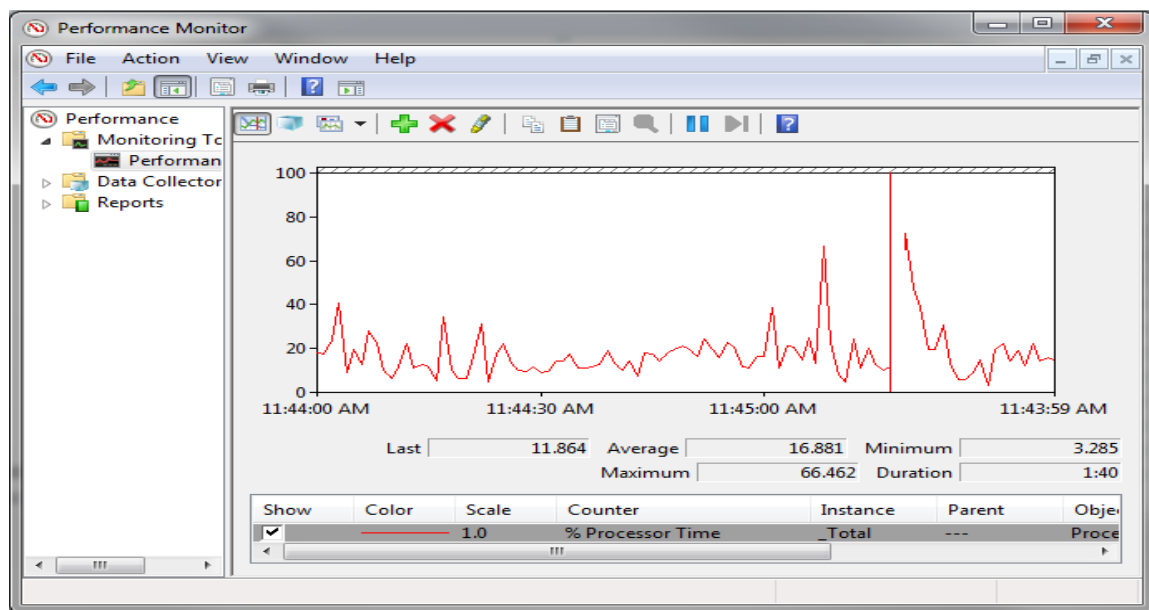


Рисунок 2 - Загруженность процессора при выполнении высоконагруженной части операции

Список использованных источников:

1. Веб-сервисы в теории и на практике [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/46374/>. – Дата доступа: 02.01.2018.
2. Определение архитектуры приложений [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/developerworks/ru/library/r-define-application-architecture-rational-software-architect-2/> - Дата доступа: 02.01.2018.
3. Анализируем скорость загрузки веб-страницы [Электронный ресурс] – Режим доступа: <https://webo.in/articles/habrahabr/16-optimization-page-load-time/> – Дата доступа: 02.01.2018.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ КЛИЕНТОВ КОМПАНИИ «ЕДА ДАРОМ» НА ПЛАТФОРМЕ ios

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Житников С.В.

Калитеня И. Л. – ассистент каф. ИСиТ, м.т.н.

Основная маркетинговая задача любого бизнеса – увеличение продаж, получение прибыли за счет привлечения новых клиентов и взаимодействия с уже имеющимися клиентами. Одним из последних нововведений в области маркетинга и продвижения, является адаптация бонусных программ и программ лояльности заведений для пользователей мобильных устройств.

Следует заметить, что приложение должно стать по-настоящему полезным для клиентов, а не просто быть средством получения рекламных сообщений.

Бонусная программа – один из видов программ поощрения покупателей. При правильном подходе, является механизмом по первичному привлечению клиентов и сбору информации, которая поможет узнать портреты групп покупателей, определить их ценность и уровень возможных маркетинговых вложений в каждую из этих групп.

В приложении «Еда Даром» пользователь получает бонусные баллы за каждый оплаченный заказ в заведении, участвующей в программе начисления бонусных баллов, а также за участие в акциях данного заведения.

В результате анализа отечественных и зарубежных аналогов мобильного приложения была разработана структура приложения и интерфейс пользователя, который бы отвечал функциональным требованиям и бы удобен и понятен для пользователей современных мобильных устройств.

Одной из отличительных особенностей бонусной программы «Еда Даром» является открытость для вступления в неё новых заведений и гибкость в адаптации к требованиям новых клиентов.

Разработанное мобильное приложение выполняет следующие функции:

- создание для пользователя личного кошелька в каждом посещенном заведении и накопление на нём баллов, которыми он мог бы расплатиться при следующем заказе;

- возможность пополнять бонусный счёт при помощи предоставления официанту, использующему вспомогательное приложение на платформе Android, QR- кода, сгенерированного внутри приложения;
- предоставления полной и актуальной информации о заведениях, участвующих в бонусной программе, включая возможность быстрой связи с заведением и прокладывание маршрута от текущего местоположения пользователя до заведения;
- информирование пользователя о новых акциях и скидках;
- простая процедура регистрации и авторизации через существующую учётную запись в социальной сети;
- возможность ознакомиться с приложением без авторизации;
- надёжная защита данных пользователя и системы с помощью шифрования передаваемых и хранимых данных.

Как и большинство современных мобильных приложений, данный проект основан на клиент-серверной архитектуре (рисунок 1). Клиентская часть запускается в среде выполнения операционной системы устройства пользователя, и служит для создания запросов для сервера и обработки ответов на эти запросы. На сегодняшний день существует клиенты приложения для платформ iOS и Android, а также вспомогательное приложение для официантов на платформе Android. Взаимодействие с серверной частью осуществляется с помощью протокола HTTPS с использованием сообщений в формате JSON. Серверная часть отвечает за бизнес-логику и систему управления базой данных.



Рисунок 1 – Диаграмма коммуникации мобильного приложения

Клиентская часть на платформе iOS написана на новом языке программирования, разработанном компанией Apple – Swift. Он представляет собой мультипарадигмальный компилируемый язык общего назначения. Одними из основных концепций, которыми руководствовались создатели языка являются удобство чтения и устойчивость к ошибкам программиста по сравнению с предшествующим ему языку Objective-C. Swift унаследовал множество положительно воспринятых сообществом программистов нововведений из таких языков программирования как C++, Java, Python и Objective-C. Одним из несомненных плюсов использования Swift является более быстрое выполнения части его функций по сравнению с другими современными языками программирования. Например, сортировки комплексных объектов выполняются в 3,9 раз быстрее чем в Python, и почти в 1,5 раза быстрее, чем в Objective-C [1].

Для информирования пользователей о новых акциях, скидках и заведениях используется механизм Push-уведомлений. Они представляют собой краткие всплывающие сообщения на экране устройства, которые доходят до него даже при выключенном приложении.

Рассылка сообщений осуществляется благодаря облачному сервису компании Google – Firebase. Он представляет из себя облачную NoSQL базу данных для приложений как сервис. Firebase предоставляет API для разработчиков, позволяющий осуществлять кроссплатформенную рассылку Push-уведомлений для всех пользователей мобильного приложения, или только для определенной части пользователей. [2].

Список использованных источников.

1. Swift Programming Language [Электронный ресурс]. – 2017. – Режим доступа: https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/. – Дата доступа: 20.12.2017.

2. Firebase [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/docs>. – Дата доступа: 20.12.2017.

РЕКЛАМА, КАК УГРОЗА ИНФОРМАЦИОННО-ПСИХОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ ЛИЧНОСТИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Заяц И.Л., Лазаренко Р.А.

Савенко А.Г. - магистр технических наук, ассистент

Информационно-психологическая безопасность личности в информационном пространстве рассматривается как одна из центральных ценностей современного общества. Одним из каналов воздействия на личность является реклама, влияние которой носит двойственный характер. Рекламная безопасность связана с проблемами свободы выбора и идентичности человека в современном информационном обществе. Общий источник внешних угроз информационно-психологической безопасности личности – та информация, которая не только вводит людей в заблуждение, в мир иллюзий, не позволяет адекватно воспринимать окружающее и самого себя, но и травмирует сознание индивида своим возрастающим количеством [1].

Интернет становится формой организации совместной информационно-познавательной и коммуникативной деятельности людей, выступая носителем нравственных ценностей [2]. Цель рекламы в интернете – умело скрыть недостатки и подчеркнуть достоинства товара или услуги. Поэтому большая часть рекламы – обман. Несмотря на внешнюю простоту, реклама использует достаточно сложные механизмы воздействия на человека. Риски для потребителя могут быть связаны не только с тем, что недобросовестная реклама намеренно вводит его в заблуждение, но и с тем, что она освобождает человека от необходимости мыслить самостоятельно, взвешивать соответствующие обстоятельства.

У большинства населения сознание постепенно начинает исполнять роль хранилища штампов и стереотипов, которые воспроизводятся в том же виде, в котором были получены. Процесс критического восприятия информации часто отсутствует.

Специалисты называют эти психологические изменения «руинизацией психики», при которой человеку становится все труднее использовать свои потенциальные интеллектуальные и волевые возможности. В постсоветских странах феномен руинизации особенно выражен у молодых поколений, у которых формирование сознания пришлось на последнее десятилетие.

Первопроходцами в блокировке рекламы стала компания Apple. В сентябре 2015 года компания представила iOS 9 — новую версию операционной системы для мобильных устройств. Тогда в браузере Safari появилась поддержка расширений, способных «блокировать контент». Adblock Plus и другие поставщики подобных расширений для десктопных браузеров смогли опубликовать в App Store свои продукты для Safari.

По данным счётчика «Рейтинг Mail.ru», 9,66% пользователей рунета используют блокировщики. При этом на пользователей такого ПО приходится 10,44% сессий и 11,92% просмотров, что подтверждает гипотезу: блокировщиками пользуется продвинутая и активная, наиболее интенсивно взаимодействующая с сайтами аудитория. Отсюда вытекает очевидное следствие, что аудитория, которая использует блокировщики, наиболее активна, потому что пользоваться интернетом без рекламы гораздо удобнее и приятнее.

С момента появления первых блокировщиков началась гонка вооружений между рекламщиками и блокировщиками. Рекламщики становились все изощреннее в методах обхода блокировок. Большинство популярных «эдблочеров» сегодня умеют не только блокировать рекламу, но и запрещать рекламным системам собирать информацию о пользователе.

Самые первые блокировщики имели в основе следующий принцип: они скрывали от глаз пользователя рекламные элементы, которые уже были загружены на страницу. Сейчас этот способ используется в некоторых программах как вспомогательный. Также можно вспомнить браузерные расширения, которые скрывают с сайтов и соцсетей тексты на определенные тематики, ориентируясь по ключевому слову.

Современные «эдблочеры» препятствуют коммуникации между отображающей веб-страницу программой, например, браузером, и серверами, с которых загружаются рекламные элементы (баннеры, объявления, видео, попапы и так далее). Или другие элементы, которые он призван блокировать (например, счетчики статистики или кнопки соцсетей).

Основной проблемой на данный момент является то, что в основе блокировки рекламы в интернете лежит не искусственный интеллект на самообучающихся нейросетях, а ручной труд, причем не только разработчиков, но и сообщества.

Продукт этого ручного труда – фильтры, то есть списки правил для определения рекламы и отделения её от полезного контента. Критерии отделения рекламы от всего остального обычно определяются волевым решением основателя того или иного фильтра с учётом мнения сообщества, которое помогает его формировать. Самый популярный набор фильтров называется EasyList. Он не принадлежит какому-то конкретному блокировщику, но используется в большинстве популярных продуктов (в том числе в Adblock Plus, uBlock Origin, AdGuard).

Именно от того, насколько оперативно обновляются фильтры, зависит качество блокировщика. Заинтересованные в показе рекламы компании постоянно работают над обходом блокировки. Они меняют все уже попавшие в фильтры идентификаторы рекламных элементов или шифруют запросы страниц к серверам рекламы, чтобы блокировщик их не остановил. Эта деятельность требует столь же постоянных контрмер.

Однако ещё до того, как началась эта борьба, началась саморегуляция. В 2011 году Adblock Plus объявил о запуске программы Acceptable Ads. Рекламодатели и площадки, согласившиеся адаптировать свою рекламу

под критерии допустимости и качества, помещались в «белый список», а пользователи Adblock Plus видели их рекламу (если не отключали это в настройках).

Учитывая, как быстро происходит смена подходов к отображения рекламных объявления и нахождение способов борьбы с ней, современные способы блокирования рекламы являются неудовлетворительными. Сервисы не могут реагировать достаточно быстро на обратную связь пользователей, в следствие чего новая реклама появляется быстрее, чем блокируют старую.

Для обеспечения безопасности ментального здоровья следует разработать нейронную сеть, с возможностью настройки отображаемой рекламы. Система будет сама подстраиваться под предпочтения пользоваться, и отталкиваясь от них, будет блокировать ту или иную рекламу. Этот подход обеспечит своевременную реакцию на новые угрозы.

Автоматизировать распознавание рекламы сложно, помимо прочего, еще и потому, что даже у людей нет единого мнения насчет того, что является рекламой, а что нет. Поэтому приложение будет отслеживать поведение конкретного пользователя на странице.

После установки приложения пользователю предлагается выбрать определенный паттерн блокировки рекламы. В него будут включаться сайты из «белого списка», а также наиболее приемлемые паттерны рекламы по отзывам других пользователей.

Основываясь на общей статистике, приложение будет скрывать ту рекламу, которая оказалась наиболее неприемлемой, по мнению других пользователей. Рейтинг неодобрения пользователей будет основываться на использовании пользователями одного из способов защиты от негативного информационного влияния – «ухода». Приложение будет отслеживать положение курсора, клики и окна, которые будут активно использоваться. Будет отслеживаться насколько долго реклама находилась на странице, пока пользователь ее не зарыл. Чем это время больше – тем выше лояльность пользователей к данному типу рекламы.

Также активно будет использоваться технология аиртрекинга [3]. Она будет отслеживать на сколько долго пользователь фокусировал внимание на блоках рекламы. Также, как и в случаях с пользовательским вводом, чем выше это время – тем выше лояльность пользователей.

Основываясь на этих данных, приложение будет скрывать «неудобную» для пользователя рекламу. Однако полностью скрыть всю рекламу не представляется возможным. Это происходит потому, что реклама является, в большинстве своем, основным источников доходов для владельцев сайта и крупных поисковых систем.

Постепенно накапливая и анализируя информацию, будут приниматься соответствующие меры по блокированию рекламы.

Список использованных источников:

1. Дроздова, А. В. Воздействие рекламы на безопасность личности в современном информационном обществе: социально-психологический аспект /А. В. Дроздова // Вестник Московского университета. Серия 14. Психология – Москва: МГУ, 2011. – с. 58-65.
2. Филиппова, Т.В. Интернет как инструмент социологического исследования / Т.В. Филиппова, Т.В.Дроздова // Социологические исследования. Выпуск 4 – Красноярск: КГАУ, 2001. – с. 115-122.
3. Окулография. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Окулография>. – Дата доступа: 20.03.2018.

ЗАЩИТА ОТ АВТОМАТИЧЕСКОГО ТРАФИКА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Казанок Д.Ю., Новохрестова А.О.

Савенко А.Г. - ассистент каф ПЭ, м.т.н.

При построении информационных систем различного уровня сложности нередко является актуальным вопрос защиты сервиса от различного рода нежелательных внешних воздействий. Отдельно хотелось бы остановиться на мобильных- и веб-приложениях. В обоих случаях они могут быть тонкими клиентами над серверными приложениями, работающими в облаке. Соответственно, при разработке таких приложений возникает вопрос безопасности серверных приложений против различных форм автоматических запросов, направленных через или в обход разработанного клиентского приложения.

Отсутствие защиты от автоматического трафика при его регулярном поступлении приводит к затратам рабочего времени, оплачиваемого работодателем, на ликвидацию нанесенного ущерба. Так же автоматический трафик значительно увеличивает нагрузку на коммуникации и снижает эффективность работы сервера. Резюмируемый итог: дополнительные финансовые расходы и угроза целостности пользовательских данных (в том числе, при устранении последствий).

Для решения такого рода задач существует понятие CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). Как следует из названия, задача состоит в том, чтобы отличить роботов от человека во время работы приложения [1]. Заметим, что в мобильных приложениях такая задача обретает свою специфику реализации и влияния на пользовательский опыт, что, в свою очередь, оказывает влияние на выбор того или иного способа реализации. Приведем самые актуальные варианты реализации CAPTCHA и особенности их интеграции:

1) Google reCAPTCHA V2

При этом способе анализируются показатели движения мыши, траектория ее движения и отклонения, а также наличие человеческой активности в файлах cookie. Отслеживание данных пользователя покажет, человек

он или нет, и только тогда предложит отметить это галочкой в поле «Я не робот». После этого — все, готово. Однако, если алгоритм Google reCAPTCHA V2 не смог точно определить человека, то следующим этапом будет показано изображение, скажем, кошки. Затем панель с изображениями под картинкой кошки. Нужно будет найти все картинки, связанные с кошкой (котятка, тигры, львы).

Google reCAPTCHA V2 является одним из самых широко используемых вариантов защиты от автоматического трафика в веб-приложениях за счет простоты прохождения теста пользователем и высокого процента определения правдивого результата. Для веб-приложений процесс интеграции состоит в следующих действиях: подключить скрипт reCAPTCHA API к странице, зарегистрировать домен сайта в административной панели при помощи аккаунта Google, вложить в форму div-элемент с классом "g-recaptcha", где в атрибуте "data-sitekey" указать ключ, полученный в административной панели [2].

В мобильных приложениях интеграция сопряжена с трудностями, т.к. потребуются показывать веб-содержимое, а веб-контейнеры в мобильных приложениях не содержат домена, совместимого с reCAPTCHA, что вынуждает хранить данное веб-содержимое на специально приобретаемом для этого веб-хостинге.

2) Invisible reCAPTCHA

Суть новой версии reCAPTCHA от Google в том, что теперь не будет необходимости куда-либо нажимать/вводить/выбирать, чтобы подтвердить, что вы человек. Для пользователей всё будет прозрачно, если у CAPTCHA не будет сомнений в его «человечности».

В новой версии системы сочетаются машинное обучение и продвинутый инструмент для анализа риска, который адаптируется к новым угрозам.

Более подробная информация о системе, вероятно, поможет создателям ботов придумать способ обойти её, поэтому в ближайшее время деталей можно не ждать.

Процесс интеграции в веб-приложениях на данный момент аналогичен процессу интеграции reCAPTCHA V2. Единственное отличие в том, что на новый элемент div нужно добавить атрибут "data-size" со значением "invisible" [2].

Использование Invisible reCAPTCHA в случае мобильных устройств представляется возможным только в гибридных приложениях, где CAPTCHA получит возможность анализировать действия пользователя. В то же время данная технология наследует все трудности реализации в мобильных приложениях от reCAPTCHA V2.

3) Взаимоверяющая CAPTCHA

Еще один вид распознавания человека на основе изображения. Он дает до 96% точности, что очень неплохо, к тому же работает в два раза быстрее, чем классическая CAPTCHA. Взаимоверяющая CAPTCHA действует по такому принципу: показывает несколько пар картинок и затем задает некоторое количество вопросов об этих картинках. Пользователь отвечает, кликая мышкой по нужным изображениям.

Стоит отметить, что данный вид CAPTCHA в совокупности с оригинальным дизайном может стать не просто оригинальным способом проверки, превращенным в мини-игру, но и гармонично вписаться в него, не ухудшая внешний вид приложения.

Готовых решений, применимых в веб- и мобильных приложениях, которые можно было бы широко рекомендовать, как в случае с решением от компании Google, нет. В основном, такой вариант CAPTCHA проектируется разработчиком под конкретное приложение. Общий случай реализации представляет собой следующие действия: создается база данных, содержащая изображения и метки, соответствующие этим изображениям; каждой метке ставится в соответствие вопрос, который будет выводиться пользователю; веб-сервер передает клиенту вопрос, и несколько изображений, имеющих общую метку, и одно или более изображение с иной меткой. В этот момент веб-сервер ассоциирует метку вопроса с пользовательской сессией (хранить в неявном виде) для дальнейшего извлечения. После того, как пользователь выбрал ответы, на сервер отправляются данные о не выбранных вариантах ответа, чтобы обрабатывать меньше данных на сервере. Остается найти соответствующие переданным изображениям метки, привести их в одинаковый вид с хранимым значением сессии пользователя и сравнить их. Если будет найдено хотя бы одно совпадение - тест не пройден [3].

Данный способ одинаково применим и для мобильных, и для веб-приложений.

Резюмируя вышесказанное, компания Google предоставляет наиболее удачное решение для предотвращения автоматического трафика веб-приложений (reCAPTCHA). В мобильных приложениях для устройств с операционной системой iOS интеграция потребует дополнительных ухищрений: наличие хостинга и доменного имени, а также использование инструментов отображения веб-содержимого. В мобильных приложениях для устройств с операционной системой Android, компания Google предоставляет API, что позволяет комфортно интегрировать reCAPTCHA. Однако в случае, когда использование Google reCaptcha не является целесообразным по ряду причин, имеет смысл затратить дополнительные ресурсы на разработку взаимодействующей CAPTCHA (например, в мобильном приложении для устройств с операционной системой iOS).

В приложениях, требующих повышенной безопасности данных пользователя (например, приложения осуществляющие операции с денежными средствами), следует рассмотреть вариант отказа от CAPTCHA в пользу использования привязки к номеру мобильного телефона.

Верификация мобильного номера происходит следующим образом: пользователь вводит свой мобильный номер в специальную форму, система генерирует пароль и отправляет его по SMS на указанный телефонный номер. Пользователь должен получить SMS с кодом и ввести его в предназначенное поле, чтобы подтвердить свою личность. Если введенный код совпадает с тем, что был отправлен системой, то пользователь получает разрешение на совершение дальнейших действий в приложении.

Таким образом, снижается уровень нежелательного автоматического трафика при условии использования дополнительных мер по его выявлению. Этими дополнительными мерами могут являться: введение ограничений на количество запросов с одного номера в определенный период времени; введение системы подачи и

мониторинга пользовательских жалоб. Набор таких мер является более эффективным в ряде случаев, чем CAPTCHA, и применяется компаниями-гигантами в сервисах охватывающих огромную долю рынка.

Список использованных источников:

1. Bursztein, E. How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation / E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, D. Jurafsky. - Stanford, 2010. - 15 с
2. Google Developers [Электронный ресурс] / Support and resources to develop. – Режим доступа: <https://developers.google.com>. – Дата доступа: 11.03.2018.
3. Chew, M. Image Recognition CAPTCHAs / M. Chew, J. D. Tygar. - Berkeley, 2004. - 19 с.

ВИРТУАЛИЗАЦИЯ СЕРВЕРОВ С ПОМОЩЬЮ WEBSHERE

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Кармызов А.С.

Скудняков Ю.А. - доцент каф. ПЭ, к.т.н., доцент

В данной работе рассматриваются возможности виртуализации серверов информационно-вычислительной системы с использованием программно-аппаратного обеспечения WebSphere для достижения оптимальных значений таких показателей качества серверов как: максимальное использование их ресурсов, оперативное предоставление услуг, снижение капитальных затрат и уменьшение административных расходов.

Виртуализация серверов предоставляет множество преимуществ, касающихся многих различных типов сред, включая среды прикладных систем промежуточного программного обеспечения [1]. В первую очередь виртуализация серверов позволяет достигать более высокого коэффициента использования ресурсов, тем самым помогая исключить серьезное недоиспользование серверных ресурсов, характерное для столь многих организаций. Работа нескольких экземпляров программного стека на одной машине часто обеспечивает более полное использование ресурсов сервера по сравнению с ситуацией, когда на сервере работает всего один программный стек. Помимо повышения коэффициента использования ресурсов, виртуализация серверов также может обеспечивать ускоренное предоставление услуг, снижение капитальных затрат и уменьшение административных расходов. Однако, чтобы получить отдачу от этих и любых иных преимуществ виртуализации серверов, необходимо эффективно осуществлять управление ее использованием.

Программно-аппаратное решение IBM WebSphere CloudBurst™ обеспечивает возможность эффективного управления виртуализацией серверов для сред IBM WebSphere Application Server. Как показано на рисунке 1, данная система ориентирована на полный жизненный цикл среды Web-приложений в виртуализированной среде.

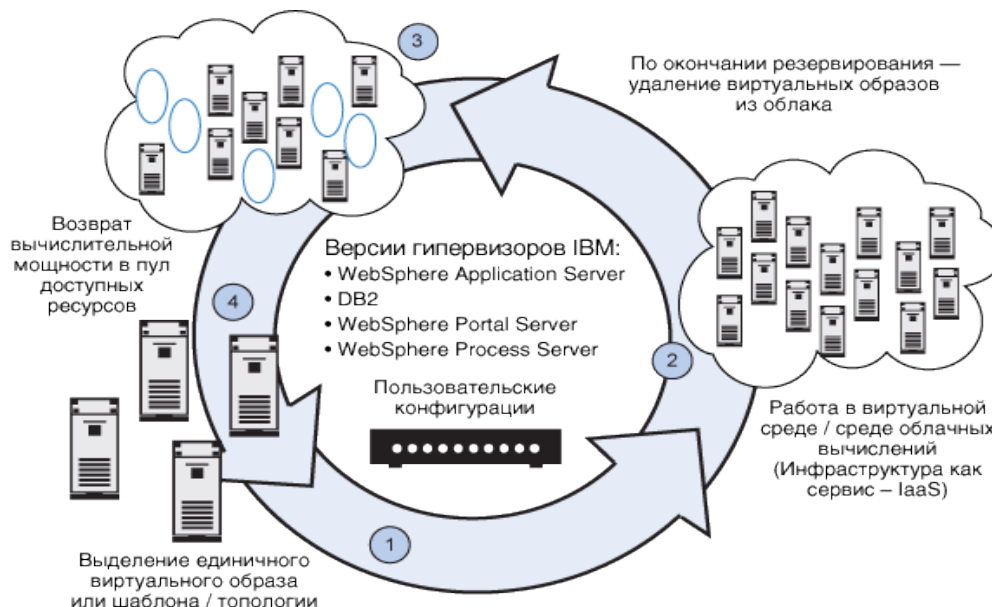


Рисунок 1 – Эффективное управление виртуализацией серверов с помощью WebSphere

CloudBurst начинает работу с того, что дает возможность определить пользовательские среды, охватывающие от одной до нескольких виртуальных машин. Затем решение осуществляет автоматизацию и оркестровку развертывания и конфигурирования виртуальных машин для создания адаптированной среды WebSphere Application Server, работающей на виртуализованных серверах. После развертывания можно использовать решение для централизованного управления и мониторинга виртуализированной среды. Когда

развертывания больше не требуется, возможно инициировать его удаление из WebSphere CloudBurst и возврат использовавшихся ресурсов в более крупный пул ресурсов в пределах конкретной организации.

Использование данного комплексного подхода к виртуализации серверов позволяет пользоваться преимуществами технологии виртуализации, не подвергая себя чрезмерным сложностям.

Список использованных источников:

1. Инновации рядом: Немного о виртуализации [Электронный ресурс]. – Режим доступа: https://www.ibm.com/developerworks/ru/library/wes-1103_inreach/wes-1103_inreach.pdf. – Дата доступа: 20.03.2018.

ПРИМЕНЕНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ В СИСТЕМАХ УМНОГО ДОМА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Кармызов А.С.

Скудняков Ю.А. - доцент каф. ПЭ, к.т.н., доцент

Рассматриваются возможности облачных технологий в системах умного дома. Система умного дома рассматривается как организованная комплексная интеллектуальная система, использование которой позволяет обеспечить высокий уровень безопасности и комфорта обитателям дома. Дано описание аппаратного и программного обеспечения технологии умного дома.

Термин «умный дом» или «интеллектуальное здание» используется для обозначения современных домов и зданий, в которых инженерные, информационные системы и системы безопасности объединены в единую и организованную комплексную интеллектуальную систему. Данная интеллектуальная система призвана обеспечивать большую безопасность, а также наилучший комфорт обитателям дома. Как правило, основная причина установки систем умного дома состоит в повышении домашнего комфорта путем автоматизации решения рутинных задач, таких как управление освещением, климат-контролем, системами мультимедиа и т. п.

Технологии умного дома состоят из двух основных составляющих: аппаратного и программного обеспечения. Типовое аппаратное обеспечение умного дома обычно включает в себя следующие компоненты: 1) контроллер (зачастую в этой роли может выступать обычный персональный компьютер), на который устанавливается программное обеспечение для управления системами умного дома; 2) модули (платы) расширения — специальные устройства, к которым подключаются различные датчики и управляемые устройства; 3) конечное оборудование — датчики для отслеживания различных параметров устройств, управление которыми ложится на плечи умного дома.

На рисунке 1 представлена типовая схема аппаратного обеспечения умного дома.

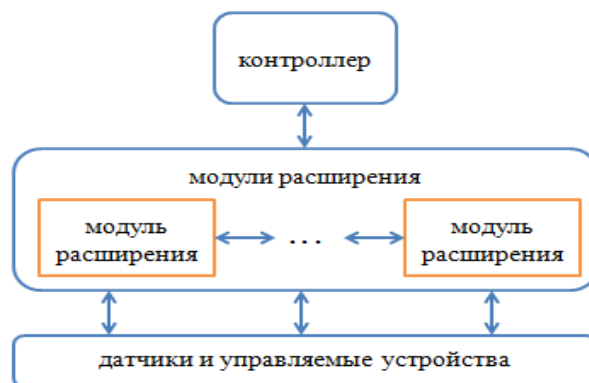


Рисунок 1 – Типовая схема аппаратного обеспечения умного дома

Управление, настройка и диагностика систем умного дома в этом случае осуществляются через стационарный компьютер. В настоящее время большинство устанавливаемых систем умного дома не обладают функцией удаленного управления через Интернет. Между тем мобильные устройства с постоянным доступом к сети стали сегодня обыденным явлением, они есть практически у каждого. В 1999 году основатель исследовательского центра Auto-IDCenter в Массачусетском технологическом институте Кевин Эштон предложил термин InternetofThings (Интернет вещей). Его суть состоит в том, что вещи нового поколения будут не только «умными», но и объединенными в сеть — Интернет вещей [1].

Концепция предполагает, что такие устройства как смартфоны, планшеты, телевизоры, различные датчики и управляемые устройства, имеющие беспроводные модули Wi-Fi и Bluetooth, смогут взаимодействовать между собой и пользователями посредством этих беспроводных модулей.

В связи с массовым распространением мобильных устройств, соответствующих концепции Интернет вещей, стало возможным удаленное управление своим умным домом. Очевидные преимущества при наличии функции удаленного управления системами умного дома: 1) главное преимущество — это конечно же большая

безопасность. При нахождении жильцов за пределами своего дома или квартиры возможно удаленное наблюдение с помощью камер за ситуацией или удаленный мониторинг в доме путем отслеживания состояния различных датчиков, используемых в системах безопасности (пожарные датчики, датчики открытия/закрытия дверей и т. д.). Кроме того, для тех, кто часто забывает выключить свет или какие-нибудь приборы, данная функция будет очень полезной; 2) основное же преимущество — это повышение комфорта пользователей умного дома. Часто в управляющих системах умного дома используют сценарии по управлению светом и теплом, когда вся работа осуществляется в автоматическом режиме. Зачастую некоторые пользователи предпочитают обходиться без таких сценариев. И при наличии функции удаленного управления пользователь, например, может сам при подходе к своему дому или при уходе с работы включить необходимые ему устройства (включить освещение, бытовые приборы, а также заранее включить отопление или кондиционер).

Осуществление функции удаленного доступа возможно с помощью применения облачных вычислений, когда пользователи обеспечиваются повсеместным доступом к сетевым вычислительным ресурсам, сервисам и приложениям. Существует несколько моделей облачных вычислений. Применительно к рассматриваемому в работе варианту удаленного управления системами умного дома больше подходит модель SaaS (программное обеспечение как услуга). Данная модель подразумевает предоставление клиенту доступа к программному обеспечению через Интернет. Основное преимущество модели SaaS для конечного пользователя состоит в отсутствии необходимости установки и обновления программного обеспечения, также ему не нужно заботиться о работоспособности оборудования, на котором функционирует приложение.

При применении облачных вычислений в системах умного дома возможны два варианта. В первом случае контроллер (сервер) для управления устройствами умного дома может быть расположен не в самом доме (эту функцию возьмет на себя облако), благодаря чему управление системами умного дома может осуществляться откуда угодно при наличии доступа к Интернету. При втором варианте (рисунок 2) контроллер может располагаться дома, но при этом через облако будет обеспечиваться только удаленное управление — все программное обеспечение будет установлено на облачном сервере. Кроме того, во втором случае от домашнего контроллера будет требоваться только функция для обеспечения модулям расширения доступа к Интернету, что в свою очередь уменьшает требования к техническим характеристикам контроллера. Также в случае внедрения удаленного управления уже в существующую систему умного дома не потребуется замены никакого оборудования, достаточно будет только обеспечить доступ контроллера к облачному серверу.

Для успешного взаимодействия облачного сервера с устройствами умного дома оба этих компонента должны «разговаривать» друг с другом на одном языке. Самым простым решением в данном случае является обмен данными через XML-сообщения. Одним из протоколов, использующих XML для обмена данными, является SOAP (от англ. Simple Object Access Protocol — простой протокол доступа к объектам). Основным преимуществом использования SOAP является то, что он способен обеспечивать непрерывное взаимодействие веб-сервиса с устройствами, работающими по различным протоколам передачи данных.

Остальные преимущества применения формата SOAP перед другими форматами для передачи данных: 1) кодировать в XML- структуры данных с использованием SOAP так же легко, как и данные простых скалярных типов; 2) при использовании SOAP- сообщений предоставляются дополнительные инструменты, позволяющие легко добавлять, например, функции обеспечения безопасности или трассировки; 3) имеются наборы инструментов SOAP для различных языков программирования [2].



Рисунок 2 – Схема аппаратного обеспечения умного дома с функцией удаленного управления

В данной работе было рассмотрено применение облачных технологий в системах умного дома. Применение подобных технологий сулит обитателям дома значительное повышение безопасности и комфорта, а также упрощается дальнейшее обслуживание и расширение уже установленных систем умного дома. Вместе с тем вопрос взаимодействия между собой облачного сервера с устройствами умного дома требует более детальной проработки. Кроме протокола SOAP следует рассмотреть и другие варианты, что будет являться темой дальнейших исследований.

Список использованных источников:

1. Черняк, Л. Интернет вещей: новые вызовы и новые технологии/ Л.Черняк, // Открытые системы. – 2013. – № 04.
2. Интеграция и взаимодействие в сети Веб [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/485/341/lecture/8211>. – Дата доступа 25.03.2018.

СИСТЕМА УПРАВЛЕНИЯ ЭЛЕКТРОДВИГАТЕЛЕМ С РЕКУПЕРАЦИЕЙ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Карнацкий П.Н.

Стешенко П.П. – доцент кафедры ПЭ, к.т.н., доцент

Рассматривается система управления электродвигателем, используемая в электротранспорте. Предложена структурная электрическая схема управления электродвигателем постоянного тока.

Использование электродвигателей в электротранспорте [1] обусловлено требованиями экологии, возможностью применения новых видов аккумуляторных батарей и упрощенной схемой управления автомобилем [2]. Значительную роль в этом отводится системам управления электродвигателем с рекуперацией (возврата) электроэнергии [3].

Предложенная система управления электродвигателем с рекуперацией состоит из 8 блоков. Электродвигатель постоянного тока с параллельным возбуждением является исполнительным устройством системы управления. Электродвигатель может работать в режиме генерации электроэнергии обратно в батарею. Это возможно при подаче тока на обмотку возбуждения во время торможения, то есть во время отсутствия подачи напряжения на якорь электродвигателя.

Управление режимом работы системы осуществляется педалью, выполненной в виде потенциометрического датчика. Сигнал с датчика поступает в блок управления электродвигателем. Структурная схема системы управления электродвигателем с рекуперацией представлена на рисунке 1.

Устройство защиты предназначено для защиты электродвигателя от внештатных режимов работы: чрезмерной подачи тока на якорь электродвигателя, защита от подмагничивания обмотки возбуждения во время разгона электроавтомобиля.

Задающий генератор формирует частоту пилообразных импульсов, которые управляют ШИМ-регулятором IR2110. Так как микросхемы IR2110 и UC3843 не могут на выходе коммутировать большие токи, применяются мощные силовые ключи на транзисторах IRF640. Режим работы электродвигателя управляется 6-ю полевыми n-канальными транзисторами, работающими в импульсном режиме.

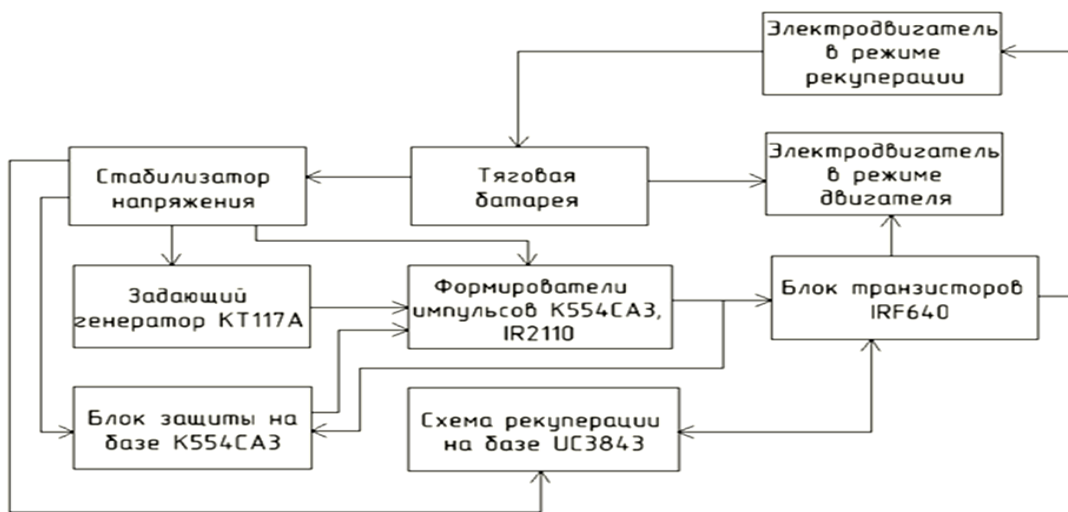


Рисунок 1 - Схема структурная управления электродвигателем

При инициализации цикла управления педалью включается вентилятор охлаждения мощных полевых транзисторов. В этот же момент происходит запуск генератора пилообразных импульсов.

Система управления переходит на следующий уровень: происходит опрос количественного значения сопротивления переменного резистора, расположенного на педали. Если его значение составляет не более 1,5кОм, то идет сигнал на схему включения режима рекуперации. Если же значение сопротивления не удовлетворяет этому требованию, то идет сигнал на микросхему IR2110 для того, чтобы открыть мощные силовые ключи управления электродвигателем (режим движения).

На основе структурной схемы нами разработана принципиальная электрическая схема и проведено моделирование ее работы с помощью программы PROTEUS 6.9 для электродвигателя мощностью 10кВт.

Список использованных источников.

1. Научный журнал «Фундаментальные исследования» [Электронный ресурс]. — Электроавтомобиль и перспективы его развития: [Электронный ресурс]. — Режим доступа: <https://fundamental-research.ru/ru/article/view?id=39606> . - Дата доступа: 04.01.2018.

2. ООО «ОНЛАЙНЕР» [Электронный ресурс]. – В Беларуси подготовили проект программы развития электротранспорта. [Электронный ресурс]. – Режим доступа: <https://auto.onliner.by/2016/06/21/elektro-3>. – Дата доступа: 04.01.2018.

3. Патентный поиск, поиск патентов на изобретения [Электронный ресурс]. – Способ рекуперации электрической энергии на рельсовом транспорте в накопительную установку вагона. [Электронный ресурс]. Режим доступа: <http://www.findpatent.ru/patent/237/2379201.html>. – Дата доступа: 04.01.2018.

СИСТЕМА УПРАВЛЕНИЯ АКТИВНЫМ АНТИКРЫЛОМ СПОРТИВНОГО АВТОМОБИЛЯ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Кожарнович В.В.

Шпак И.И. – зав. кафедрой ПЭ, к.т.н., доцент

Быстрый рост мощности и похожие на крылья аэродинамичные кузова спортивных автомобилей стали причиной появления столь заметной подъемной силы, что ее уже невозможно стало игнорировать. Антикрыло – основной элемент конструкции спортивного автомобиля, на котором в процессе взаимодействия с потоком набегающего воздуха первично возникает сила обратная подъемной, называемая прижимной. Несмотря на явные достоинства, антикрыло имеет и недостаток, заключающийся в том, что его установка повышает сопротивление набегающему воздуху, тем самым ухудшая динамику спортивного автомобиля. Компромиссом здесь является использование активного антикрыла, то есть антикрыла, изменяющего свой угол атаки согласно определенному алгоритму в процессе движения автомобиля. В докладе приводятся результаты разработки системы управления активным антикрылом спортивного автомобиля.

На модуль драйвера антикрыла в составе разработанной системы управления активным антикрылом спортивного автомобиля были возложены задачи по изменению угла атаки антикрыла в ходе движения автомобиля, опираясь на значения текущей скорости движения и состояния выключателя стоп-сигнала. Вышеперечисленные значения модуль драйвера антикрыла получает по бортовой шине CAN силового агрегата автомобиля путем непрерывного чтения шины, фильтрации и получения сообщений с необходимыми идентификаторами. Для демонстрации работы и проверки работоспособности системы был разработан также модуль диагностического симулятора.

Структура разработанной системы управления активным антикрылом спортивного автомобиля представлена на рисунке 1.

Диагностический симулятор – переносимое устройство, предназначенное для симуляции определенных процессов, происходящих в автомобиле и последующих генерации и передачи соответствующих сообщений на шину CAN, а также для визуального отображения состояния симулируемых процессов. Все узлы и блоки диагностического симулятора конструктивно удалены от блока питания, входящего в состав модуля драйвера антикрыла.

Узлы тактирования необходимы для генерации тактовых импульсов для управляющих микроконтроллеров и контроллеров шины CAN.

Узел симуляции скорости необходим для непосредственной установки симулируемого значения скорости посредством механического воздействия на него со стороны пользователя, в результате чего на вывод встроенного АЦП микроконтроллера 1 поступает соответствующий аналоговый сигнал, после преобразованный и сохраненный в ОЗУ микроконтроллера в цифровом двоичном виде.

Узел симуляции состояния выключателя стоп-сигнала необходим для непосредственной установки симулируемого состояния педали тормоза посредством механического воздействия на него со стороны пользователя, в результате чего на вывод порта ввода-вывода (I/O) микроконтроллера 1 поступает соответствующий электрический сигнал, после сохраняющийся в ОЗУ микроконтроллера в цифровом двоичном виде.

Микроконтроллер 1 выполняет обработку симулируемых значений величин и состояний узлов симуляции и сохраняет их в собственном ОЗУ в цифровом двоичном виде. Далее микроконтроллер 1 осуществляет передачу их контроллеру 1 шины CAN по встроенному интерфейсу SPI, а также отправляет соответствующую информацию блоку отображения состояния узлов симуляции, реализованному на модуле LCD, через порт ввода-вывода. Контроллер 1 шины CAN приняв информацию, обрабатывает, преобразовывая ее в CAN-сообщения, которые затем помещаются в буфер отправки для дальнейшей передачи трансиверу 1 шины CAN.

Трансиверы шины CAN необходимы для работы в качестве интерфейсов между контроллерами CAN-протокола и физической шиной, то есть для преобразования CAN-сообщений в сигналы дифференциальной шины CAN и обратно.

Драйвер антикрыла – устройство, стационарно устанавливаемое в автомобиле и предназначенное для приема, анализа, фильтрации и обработки сообщений, поступающих с шины CAN, и на основании определенного алгоритма управляющее углом положения антикрыла.

Блок питания, входящий в состав драйвера антикрыла, необходим для преобразования постоянного напряжения питания бортовой сети автомобиля +12 В в постоянное напряжение +5 В – для питания цифровых микросхем модулей драйвера антикрыла и диагностического симулятора. Блок питания построен по высокоэффективной технологии SEPIC [1] (Single Ended Primary Inductance Converter, преобразователь с несимметричной первичной обмоткой), благодаря чему система может работать в достаточно широком диапазоне входных напряжений. SEPIC-преобразователь включает в себя контроллер ШИМ, реализующий определенные длительности включения и отключения электронного силового ключа, который в свою очередь

задействует соответствующие элементы LC-контура, выполняющие так называемую «прокачку» тока к потребителям.

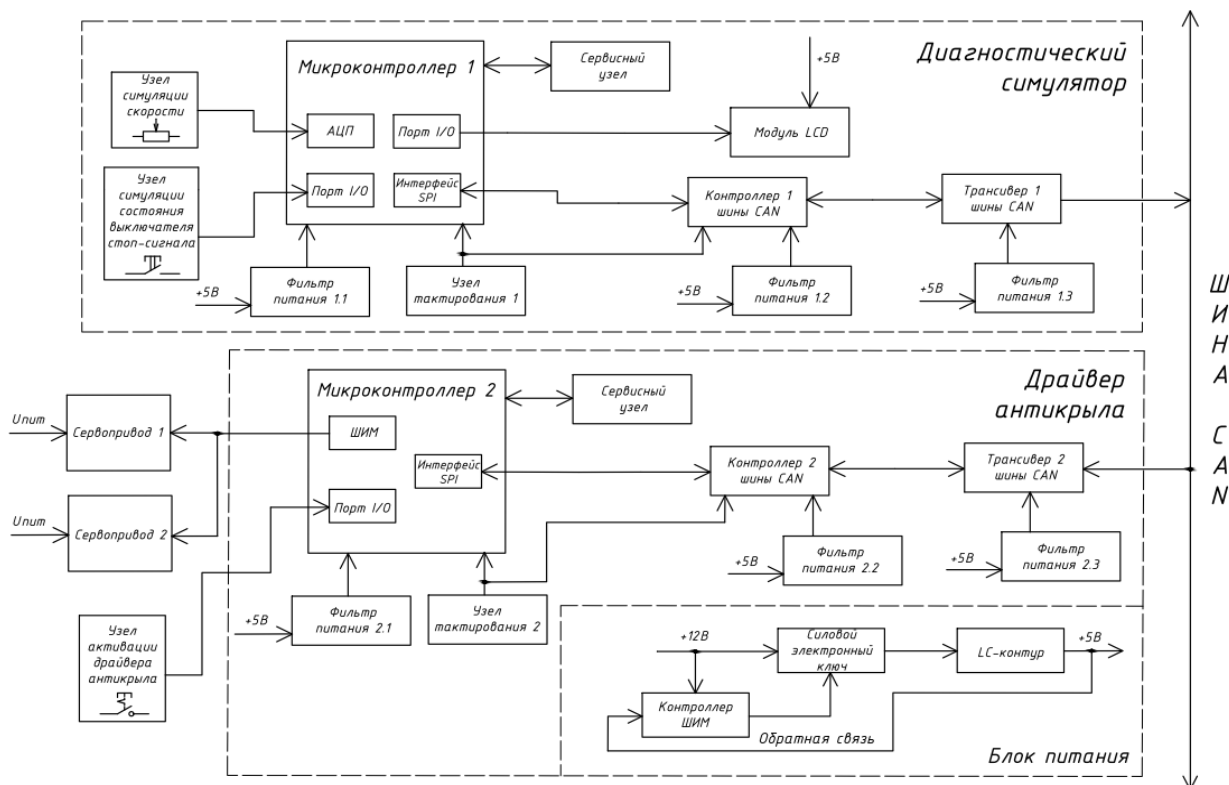


Рисунок 1 - Структура системы управления активным антикрылом спортивного автомобиля

Узел активации драйвера антикрыла необходим для включения и отключения драйвера антикрыла путем механического воздействия на него со стороны водителя, в результате чего на вывод порта ввода-вывода микроконтроллера 2 поступает соответствующий электрический сигнал, включающий либо отключающий алгоритм обработки CAN-сообщений.

Контроллер 2 шины CAN принимает сообщения CAN от трансивера 2 шины CAN в общий буфер сбора сообщений, где далее идентификаторы принятых сообщений, проходя фильтрацию, вместе с полезной информацией сообщения помещаются в один из приемных буферов. Оттуда посредством интерфейса SPI по соответствующему запросу данные передаются управляемому микроконтроллеру 2. Микроконтроллер 2 осуществив прием отфильтрованных CAN сообщений, следуя определенному алгоритму, с помощью таймера-счетчика в режиме PWM генерирует ШИМ-сигнал, управляющий валами сервоприводов. Все возможные варианты работы системы с экспериментальной микропрограммой сведены в таблицу 1.

Таблица 1 – Таблица истинности работы системы

Состояние драйвера антикрыла	Значение текущей скорости	Состояние выключателя стоп-сигнала	Длительность импульсов ШИМ, мс	Угол атаки, град.
Выключен	Любое значение	Любое состояние	1.05	10
Включен	<140	Любое состояние	1.05	10
Включен	≥140	Выключен	1.10	17
Включен	≥200	Выключен	1.17	25
Включен	≥140	Включен	1.25	45

Для обмена служебной информацией между блоками управляющих микроконтроллеров и персональным компьютером либо смартфоном с поддержкой технологии OTG, в частности – для загрузки микропрограммы в память микроконтроллера, предусмотрен Сервисный узел.

В отличие от существующих решений, разработанная система функционирует в среде штатной шины CAN автомобиля, что в свою очередь позволяет избежать установки сторонних датчиков, а соответственно упростить устройство и снизить стоимость. Схемотехнически система управления активным антикрылом спортивного автомобиля подходит большинству современных гражданских автомобилей, а в качестве единственной переменной от одной к другой модели автомобиля используется соответствующая микропрограмма микроконтроллера – прошивка.

Список использованных источников:

1. Гу, В. Разработка преобразователя SEPIC / В. Гу // Компоненты и технологии. – 2008. – №9. – С. 125–128

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ГЕНЕРАЦИИ ТЕСТ-КЕЙСОВ МЕТОДОМ PAIRWISE

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Козырь И.Ю.

Образцова О.Н. – доцент каф. ИСиТ, к.т.н., доцент

Тестирование – процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результаты работ с целью определить, что они соответствуют описанным требованиям, показать, что они подходят для заявленных целей и для определения дефектов.

Существуют различные методики тестирования, одной из которых является исчерпывающее тестирование – методика, в которой набор тестов включает в себя все возможные комбинации входных данных и предусловий. За последние 40 лет были предложены семь принципов тестирования, которые являются общим руководством для тестирования в целом. Один из принципов гласит следующее: исчерпывающее тестирование недостижимо. Это означает, что полное тестирование с использованием всех возможных комбинаций входных данных и предусловий физически невыполнимо (за исключением тривиальных случаев). Вместо исчерпывающего тестирования должны использоваться анализ рисков и расстановка приоритетов, а также различные техники создания тестов: классы эквивалентности, анализ граничных значений, попарное тестирование (pairwise) и др.

Попарное тестирование – методика разработки тестов методом черного ящика, при использовании которой тестовые сценарии разрабатываются таким образом, чтобы выполнить все возможные отдельные комбинации каждой пары входных параметров [1]. Данный метод не обеспечивает исчерпывающее тестирование, однако позволяет покрыть 100% всех пар любых двух параметров системы. По оценкам различных исследований, около 90% дефектов являются результатом обработки пар входных значений параметров [2, 3], значит, грамотно используя технику попарного тестирования, можно обнаружить большой процент дефектов, затратив небольшое количество ресурсов. Для получения набора тестов, отвечающего требованию полного покрытия любых двух пар параметров системы, могут быть использованы различные стратегии/алгоритмы: ортогональные массивы, «латинские квадраты», AETG, TCG, IPO и др.

Целью дипломного проекта является разработка веб-приложения, генерирующего тест-кейсы согласно Pairwise подходу, основываясь на заданных входных параметрах и их значениях. Задачей является непосредственно генерация тест-кейсов.

Для достижения поставленной цели были проанализированы источники литературы, описывающие разные комбинаторные стратегии и алгоритмы, на основе которых может быть реализован Pairwise подход. По результатам анализа была выбрана стратегия IPO.

Для реализации веб-приложения были выбраны языки программирования JavaScript и PHP. JavaScript используется для всех действий на клиентской части веб-приложения, например, импорт параметров и значений из файла, вывод и экспорт полученных тест-кейсов, изменение количества полей для ввода и т. д. Кроме того для ускорения разработки была использована JavaScript-библиотека jQuery, которая является «де-факто» стандартом веб-разработки. На PHP написаны непосредственно функции, используемые для генерации тест-кейсов. Выбор генерации тест-кейсов на стороне сервера обусловлен тем, что сервера имеют куда большую вычислительную мощность, чем клиентские рабочие станции. Кроме перечисленных языков программирования и библиотек был использован фреймворк Bootstrap, который значительно ускоряет разработку пользовательского интерфейса клиентской части веб-приложения.

Результатом является разработанное и готовое к использованию веб-приложение, выполняющее поставленную выше задачу – генерацию тест-кейсов согласно Pairwise подходу. Для работы инженер по тестированию задает вручную (либо импортирует) все параметры системы и все возможные значения всех параметров системы. Выходными данными является набор тест-кейсов, полученный путем применения алгоритма IPO. Для удобства дальнейшей работы тест-кейсы могут быть экспортированы.

Список использованных источников.

1. Erik van Veenendaal, «International Software Testing Qualification Board Glossary» [Электронный ресурс]. – ISTQB®, 2015 –. Режим доступа: <https://www.istqb.org/downloads/glossary.html>. – Дата доступа: 04.01.2018.

2Wallace, D.R. Failure Modes in Medical Device Software: An Analysis of 15 Years of Recall Data/D. R. Wallace, D. R. Kuhn, // Int'l Jour. of Reliability, Quality and Safety Engineering, vol. 8, no. 4, 2001.

3. Smith, B. Challenges and Methods in Testing the Remote Agent Planner/ B. Smith, M. S. Feather, and N. Muscettola, //Proc. 5th Int'l Conf. on Artificial Intelligence Planning and Scheduling (AIPS 2000), 2000.

РИСКИ ИСПОЛЬЗОВАНИЯ ОБЛАЧНЫХ ХРАНИЛИЩ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Крень Н.С., Бакунович Д.С.

Савенко А.Г. - магистр технических наук, ассистент

Облачные хранилища получили большое распространение в последние годы. С развитием технологий стремительно увеличивается объем хранимой и обрабатываемой информации, а также требования к скорости обмена данными. В связи с этим сервисы облачных хранилищ стали популярны и востребованы пользователями.

Существуют различные вариации облачных хранилищ, отличающиеся структурой и функционалом, со своими достоинствами и недостатками. В этом месте встает вопрос о рисках использования такого типа систем.

Данную проблему мы можем разделить на три категории:

1. Сохранность данных;
2. Доступность данных;
3. Конфиденциальность данных.

Сохранность данных напрямую зависит от конфигурации и качества обслуживания сервера.

Применение RAID-технологий, настройка автоматического резервного копирования, своевременное техническое обслуживание сервера и другие подобные меры приводят к обеспечению максимально высокого показателя сохранности, что положительно отражается на безопасности облачных хранилищ.

Однако, в случае несоблюдения данных мер, вероятность потери информации куда выше, чем при хранении данных на локальном диске или переносном носителе.

Рассмотрим категорию доступности данных. Мы не всегда можем получить ту информацию, которая находится на сервере. К примеру, если у нас отсутствует подключение к серверу, либо он неправильно настроен мы не сможем подключиться для получения данных. На доступность данных влияют: настройки сервера, доступ к интернету, аварии на линиях, DDOS-атаки и так далее. Мы не можем повлиять на большую часть этих проблем, если сервер не обслуживается самостоятельно и не находится в локальной сети. Обычно, именно так и происходит: арендуется удаленный облачный сервер, так как настраивать и содержать собственный менее выгодно и безопасно - на крупных сервисах закрыта большая часть уязвимостей.

Минимизировать проблемы доступа в данном случае можно путём использования нескольких различных интернет-подключений. Чтобы иметь возможность получать данные из любой точки мира, рекомендуется использовать спутниковое подключение. Однако, полностью избежать проблем с доступом на данный момент невозможно в связи с возможными авариями и хакерскими атаками.

Последняя и наиболее важная категория - конфиденциальность. При использовании локального сервера, особенно без доступа в интернет, вероятность утечки данных минимальна. В данном случае, она зависит от уязвимостей системы и добросовестности людей, имеющих доступ к серверу.

Первая проблема, с которой можно столкнуться - перехват данных при отправке на сервер, либо загрузке с сервера. Это достижимо при использовании так называемых снифферов - программ для перехвата сетевого трафика. Избежать данной проблемы можно путём использования зашифрованных соединений (например, FTPS, SFTP) либо шифруя данные локально, до начала передачи.

Вторая проблема - утечка данных прямо с сервера. Её возникновение возможно по различным причинам, таким как: уязвимости системы, неправильная конфигурация сервера, не добросовестный администратор и так далее.

Большинство серверов хранят данные в зашифрованном виде, что позволяет обезопасить хранящуюся информацию. В противном случае следует шифровать важные данные на локальном устройстве перед загрузкой на сервер. Шифровать публичные данные просто не имеет смысла.

В качестве третьей проблемы можно выделить авторизацию. Завладев данным для авторизации, злоумышленник может получить доступ ко всей вашей информации, хранящейся на сервере. Чтобы этого избежать, рекомендуется настроить двухфакторную аутентификацию и установить защиту от перебора пароля. Со стороны пользователя требуется хранить в секрете данные аутентификации и сопутствующие личные данные.

Подводя итог, можно сказать, что при корректном использовании и настройке облачного сервера, хранение данных достаточно безопасно. Однако, остаётся открытым вопрос доступа к информации, так как под воздействием различных, часто не зависящих от пользователя факторов, соединение с сервером может быть прервано.

ПРОБЛЕМЫ БЕЗОПАСНОСТИ СОВРЕМЕННЫХ CMS

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Кузнецов В.А., Иваньков А.Ю., Матусевич А.В., Высоких В.А.

Бакунова О.М. - ст. преподаватель каф. ИСиТ, м.т.н.

В современном мире всё больше развиваются технологии, использующие всемирную сеть интернет. Причина такой востребованности данным направлением является развитие малого бизнеса, который не может существовать без web поддержки и рекламы. Для большинства развивающихся компаний нанять программиста для разработки своего интернет ресурса с «нуля» с собственной «админкой» и т.д. очень накладно с финансовой стороны. По этой причине всю большую востребованность стали набирать CMS системы, позволяющие пользователю самому создать сайт на примере существующего шаблона или по крайней мере платить не за всю разработку сайта, а лишь отдельных его частей. Данные системы упростили процесс создания простых сайтов, без каких-либо необычностей. С появлением и развитием CMS систем пропорционально началась востребованность данным направлением и хакеров, взломщиков, недоброжелателей. В данной статье рассмотрим основные проблемы ведущих CMS систем в пространстве интернет.

Joomla

Joomla – одна из самых популярных CMS, и соответственно, к ней приковано внимание большого числа злоумышленников.

С начала 2015 года, в Joomla было найдено 37 уязвимостей. Большинство из них были достаточно серьезными и приводили к утечке данным неавторизованным пользователям, возможности выполнения SQL-инъекций, а также возможности получения комбинаций имени пользователя и пароля, либо же их сброс в начальное состояние, а также способность повышения своих привилегий. Большинство этих уязвимостей были исправлены в последующих версиях Joomla. Однако мы рассмотрим некоторые из них, представляющие интерес:

- Уязвимость в компоненте EQ Event Calendar, позволявшая злоумышленнику удаленно выполнить SQL-инъекцию. Проблема заключается в том, что в обработчике поля ID не была выполнена фильтрация данных. На данным момент не известно, принимались ли какие-то изменения в Joomla для исправления данной проблемы. Общая рекомендация – заменить этот компонент на другой аналогичный.

- Единственная уязвимость за последние 3 года, позволявшая модифицировать данные существующего пользователя. Также использовалась для сброса имени пользователя, пароля и группы. Проблема была исправлена в версии 3.6.4, однако до сих пор активно используется против сайтов, не обновивших версию Joomla.

- Уязвимость, затрагивающая Joomla, начиная с версии 1.7.3, выпущенной еще в 2011 году. Заключалась в некорректной инвалидации кэша, приводившей к утечке содержимого форм. Исправлена в версии 3.7.2.

- Уязвимость в инсталляторе Joomla, которая не проверяла принадлежность webspace пользователю, что позволяло удаленному пользователю получить контроль над приложением оценивая логи, накопленные во время его работы. Исправлено в версии 3.7.4.

- Уязвимость, позволявшая пользователю обойти двухфакторную авторизацию. Проблема, показывающая, что двухфакторная авторизация не является гарантией полной защищенности пользовательских данных.

Большинство уязвимостей исправляются своевременно после их обнаружения. Главная рекомендация пользователю – обновлять свою CMS своевременно.

1С-Битрикс

В CMS Bitrix, используемых на данный момент можно выделить следующие проблемы безопасности: 1) Большое количество XSS

Административный раздел «Дополнительные поля» - стал самым уязвимым местом для XSS атак непостоянного характера. Данный раздел позволяет создавать различные поля для пользователей. При создании указанных полей самыми уязвимыми стали конструкции для создания типов данных «Список» и «Видео».

CSRF атака.

Допустимость принятия CSRF токенов, как при настройке пользователей, так и при настройке аккаунта администратора системы. Например, при смене пароля, или иных учетных данных администратора, данные отправляются на обработчик следующим образом:

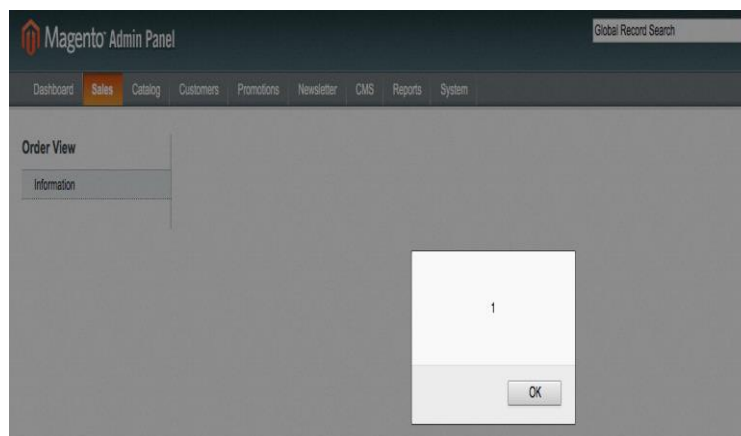


Рисунок 1 – Пример уязвимости Magento

Для устранения подобных уязвимостей разработчики Magento выпускают патчи, чтобы повысить безопасность и устранить всякую уязвимость.

Проанализировав крупные указанные CMS системы можно сделать вывод, что все они имеют достаточное количество недостатков. Из вышеуказанного видно, что в основном все веб системы данного назначения имеют схожие проблемы. Причём разработчики каждой из ведущих CMS систем проводят сбор статистики уязвимых мест и решают данные проблемы с выпуском обновлений. Что в очередной раз доказывает необходимость своевременного обновления.

Что касается выбора той или иной CMS системы для работы, то с уверенностью рекомендовать определённую невозможно. У каждой есть какие-то плюсы и минусы, которые нужно учитывать при выборе.

Список использованных источников:

1. Петренко, С.А. Политики безопасности компании при работе в Интернет./ С.А. Петренко, В.А. Курбатов. – Изд-во ДМК Пресс, 2011, 396 с.
2. Моор, П.К. Информационные системы в экономике: учебные пособие./ П.К. Моор, С.М.Моор, А.П. Моор, – Тюмень: Издательство Тюменского государственного университета, 2011, 192 с.
3. ЗАО «Интеллектуальные системы» [Электронный ресурс]. – 2012. – Режим доступа: <http://www.is.by/>. – Дата доступа: 25.12.2017

ДИАГНОСТИЧЕСКАЯ СИСТЕМА ЭЛЕКТРОНИКИ АВТОМОБИЛЯ НА ПРЕДМЕТ ОШИБОК. ИЗМЕНЕНИЕ ПАРАМЕТРОВ ВПРЫСКА И ЗАЖИГАНИЯ ТОПЛИВА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Кузьмицкий В.И., Биркос В.А.

Бакунова О.М. – ст. преподаватель каф. ИСиТ, м.т.н.

В докладе представлен сравнительный анализ диагностическая система электроники автомобиля.

С развитием технологий, в автомобиле строении начали применяться различного рода электронные системы. Очень значимым шагом стал переход на электронные блоки управления двигателем (ЭБУ). Первые моторы с таким типом управления появились в конце 1970х начале 1980х годов. В самом начале это были примитивные микросхемы, считывающие данные с нескольких значимых датчиков, таких как датчик массового расхода воздуха, датчик положения дроссельной заслонки, датчик температуры поступающей смеси и т.д.. Такие моторы выгодно отличались от старых, производство смеси и впрыск топлива которых осуществлялись механическим карбюратором. Более рациональное распределение топлива, стабильная работа не зависимо от температуры воздуха, существенно увеличили производительность, понизили расход топлива и как следствие снизили загрязнение окружающей среды.

Новый тип управления двигателем позволил добиться от моторов колоссального прироста производительности, при тех же параметрах механических составляющих. Но кроме плюсов появились и минусы. Самый главный, это цена. Новые системы были очень дорогими в производстве и ремонте, и для эффективности определения проблем понадобилось средство диагностики. Информация, поступающая от датчиков, преобразуется специально установленным преобразователем в цифровой сигнал. В состав ЭБУ входят информационный и контрольно-вычислительный модули, с помощью которых и происходит оптимизация и отладка всей системы. Если на ЭБУ приходит сигнал от датчика, который выходит за диапазон допустимого значения, система сигнализирует об ошибке и записывает её код. В этом случае, необходимо лишь считать код ошибки, чтобы найти неисправный датчик.

Для синхронизации в каждом авто есть специальный диагностический разъем, к которому подключается ПК либо другое устройство, способное расшифровать коды ошибок. В современных машинах используются единые протоколы OBD-II с идентичными разъемами подключения. На современном рынке множество устройств, способных подключиться к ЭБУ автомобиля без ПК, кабелей, блютуз-адаптеров. Для современного автолюбителя не составляет труда самостоятельно продиагностировать и скинуть ошибки ЭБУ. Но 10 лет назад, подобные операции проводились только на официальных станциях обслуживания с дорогим оборудованием.

Современные блоки управления так же позволяют изменить параметры работы двигателя, в случае его совершенствования и переделки. Существуют программные средства, созданные специально для перенастройки и адаптации параметров ЭБУ. Но неквалифицированным специалистам не рекомендуется изменять что-либо в заложенной программе, так как в результате может произойти сбой всех систем.

Нужно понимать, что любое изменение параметров от заводских приведет к не рекомендованной производителем эксплуатации. Вырастет отдача мотора, но повысится расход топлива и выбросы вредных веществ. При кардинальном изменении, может снизиться ресурс двигателя. Тем не менее, данный вид тюнинга очень популярен, потому что является самым дешевым и простым способом поднять отдачу мотора на несколько десятков лошадиных сил.

АЛГОРИТМ РАБОТЫ СИСТЕМЫ УЧЕТА ЗАДОЛЖЕННОСТИ ПО БАНКОВСКИМ КРЕДИТНЫМ КАРТАМ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Ламчановский А.Г.

Бахтизин В.В. – профессор каф. ПОИТ, к.т.н., доцент

В докладе рассматривается вопрос автоматизации системы учета задолженностей клиентов банка, построен алгоритм взыскания задолженности по банковским кредитным картам.

Требования банков к функциональности систем учета клиентской задолженности по кредитным картам постоянно повышаются. Банки выводят на рынок кредитные карты с новыми условиями, увеличивается количество клиентов. Для учета и взыскания задолженности банки используют специальные информационные автоматизированные системы.

Просроченная задолженность – это непогашенная в срок задолженность по основному долгу или плановым процентам за пользование кредитом, а также иным платежам по кредитному договору. С момента возникновения просрочки по выплате по кредиту банк начисляет штрафы и пени, размер и принцип расчета которых указываются в кредитном договоре.

До сих пор не существует оптимального алгоритма решения проблемы взимания просроченной задолженности. Существует множество нерешенных задач, связанных с управлением проблемными активами. Каждый банк самостоятельно выбирает пути работы с просроченной задолженностью в соответствии с его структурой, практикой взаимодействия подразделений банка, объемом и спецификой кредитного портфеля.

С клиентами, имеющими просроченную задолженность, работают специальные подразделения банков.

Предлагаемый алгоритм взыскания просроченной задолженности состоит из следующих шагов.

1. Отправить клиенту SMS-сообщение с информацией о задолженности по кредитной карте и просьбой оплатить. Если задолженность будет погашена в течение 3-х дней, перейти к шагу 7. Иначе перейти к шагу 2.
2. Сотрудник колл-центра банка звонит клиенту, продолжается отправка информационных SMS-сообщений. Если задолженность будет погашена в течение 60 дней от образования просроченной задолженности, перейти к шагу 7. Иначе перейти к шагу 3.
3. Сотрудник колл-центра банка связывается с доверенными лицами (как правило, родственниками) клиента. Клиенту банка отправляют SMS-сообщения и сообщения по электронной почте с просьбой погасить долг. Если задолженность будет погашена в течение 120 дней от образования просроченной задолженности, перейти к шагу 7. Иначе перейти к шагу 4.
4. Сотрудники банка выезжают по месту проживания и/или регистрации, а также работы клиента банка, и просят погасить задолженность, общаются с родственниками, коллегами и начальством клиента. Если клиент соглашается погасить просроченную задолженность, перейти к шагу 7. Иначе перейти к шагу 5.
5. Банк обращается в государственные органы с целью защиты своих законных интересов. Если клиент погашает просроченную задолженность, перейти к шагу 7. Иначе перейти к шагу 6.
6. Судебные приставы посещают клиента, и описывают его имущество в пользу банка.
7. Завершить алгоритм.

Автоматизированные информационные системы учета задолженности, прежде всего, предназначены для поддержки стадии переговоров с клиентом, так как эта стадия является самой эффективной на практике. С другой стороны, те же системы могут использоваться для учета работы с клиентами и на более поздних стадиях в целях упрощения формирования общей стратегии взыскания. Методы работы сотрудников банка, связанных с взысканием просроченной задолженности, и внешних подрядчиков разделяются в зависимости от стадии работы.

Данный алгоритм удовлетворяет требования законодательства Республики Беларусь. Его использование позволяет четко разграничить роли подразделений банка, повысить эффективность работы сотрудников,

сократить время выполнения задач за счет автоматизации процессов.

Система учета задолженности по банковским картам должна взаимодействовать с другими системами банка. Для этих целей банки разрабатывают системы самостоятельно или покупают комплектные готовые решения для работы банка.

Примеры внешних систем, представленных на рынке Республики Беларусь:

1. EPAM.Debt Collection – это гибко настраиваемая система, предназначенная для розничных банков;
2. Системные технологии Collection – система для предупреждения возникновения задолженности и управления процессами взыскания.

Список использованных источников:

1. Налогообложение, учет и отчетность в коммерческом банке //Методический журнал «Бухгалтерский учет просроченной задолженности физических лиц». – 2011. - №4 – С. 5-10.

2. Мирошниченко, Ю. В. Работа с просроченной задолженностью // InfoBank.by [Электронный ресурс]. - Режим доступа: <http://infobank.by/rabota-s-prosrochennoj-zadolzhennostju.html>. – Дата доступа: 23.11.2017.

3. Лопатин, Ю. В. Аналитический отчет «ИССЛЕДОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В БАНКОВСКОМ СЕКТОРЕ РЕСПУБЛИКИ БЕЛАРУСЬ» // Научно-технологическая ассоциация "Инфопарк" by [Электронный ресурс]. - Режим доступа: <http://infopark.by/sites/default/files/old/docs/7660.pdf>. – Дата доступа: 24.11.2017.

РОБОТЫ В CRM БИТРИКС24. REST ДЛЯ CRM-ТРИГГЕРОВ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Литовко Д.А.

Бакунова О.М. – ст. преподаватель каф. ИСиТ, м.т.н.

Бакунов А.М. – ст. преподаватель каф. ИСиТ, м.т.н.

Калетня И.Л. – ассистент каф. ИСиТ, м.т.н.

Хмелевская А.Л. – ассистент каф. ПЭ

В современном обществе большинство компаний начинают искать различные способы для автоматизации ведения бизнеса. Для этого были созданы различные CRM, которые помогают вести бизнес и упрощают большую часть бизнес-процессов, экономя этим время и финансы организации. В докладе представлен пример использования в бизнесе CRM Битрикс24 [1].

В CRM Битрикс24 существует обширный REST API, который позволяет добавлять функционал как в облачный, так и в коробочный Битрикс24, а также создавать всевозможные интеграции с внешними системами. Использование методов REST позволяет создавать и модифицировать объекты CRM, задачи, публикации, сделки и т.д.

Работа с клиентами и продажами состоит из множества рутинных операций - планирование встреч, работа с почтой и звонками, постановка задач и контроль их выполнения. CRM Битрикс24 предоставляет огромное количество различных инструментов, которые автоматизируют эти операции до максимально возможного уровня. Одним из таких инструментов являются так называемые роботы.

Пользователи могут создавать и настраивать роботов, которые будут отсылать различные нотификации, ставить задачи при поступлении нового лица, выполнять почтовые рассылки, управлять статусами сделок, планировать встречи и т.д. Роботы позволяют решать базовые задачи автоматизации бизнеса, используя простые линейные алгоритмы. За счет своей линейности являются простым и эффективным инструментом. Для достижения максимальной автоматизации в связке с роботами используются триггеры.

Для использования триггеров необходимо их создать, либо использовать уже существующие (CRM содержит набор базовых триггеров). Создаются триггеры с помощью приложений Битрикс24 [2]. Приложения Битрикс24 делятся на 3 типа:

- Внешние серверные приложения (приложения используют только API, и никак не интегрируются в интерфейс Битрикс24. Доступ для использования API получается посредством протокола авторизации OAuth 2.0);

- Серверные приложения (размещаются на сторонних серверах);

- Статические приложения (размещаются в облаке 1С-Битрикс).

Непосредственно такие приложения и содержат в себе описание событий и линейные алгоритмы триггеров. При добавлении нового триггера, необходимо описать всю цепочку действий, которые данный триггер будет выполнять. После добавления триггера, необходимо описать событие, при наступлении которого будет вызван данный триггер.

Когда все триггеры и события будут описаны необходимо установить приложение в CRM. После установки приложения появится возможность устанавливать триггеры, на различные события (рисунок 1).

Триггеры можно использовать для разных целей и вот одна из возможных:

В CRM, на определенном этапе бизнес-процесса, роботы выполняют почтовую рассылку для определенного списка лиц. В письмо попадает коммерческое предложение для конкретного лица. Делается это для привлечения лица и перевода его в статус клиента. Вместе с предложением отправляется уникальная ссылка на нужную страницу, где можно подробнее ознакомиться с полученным предложением. В ссылку встраиваются данные лица (далее клиент):

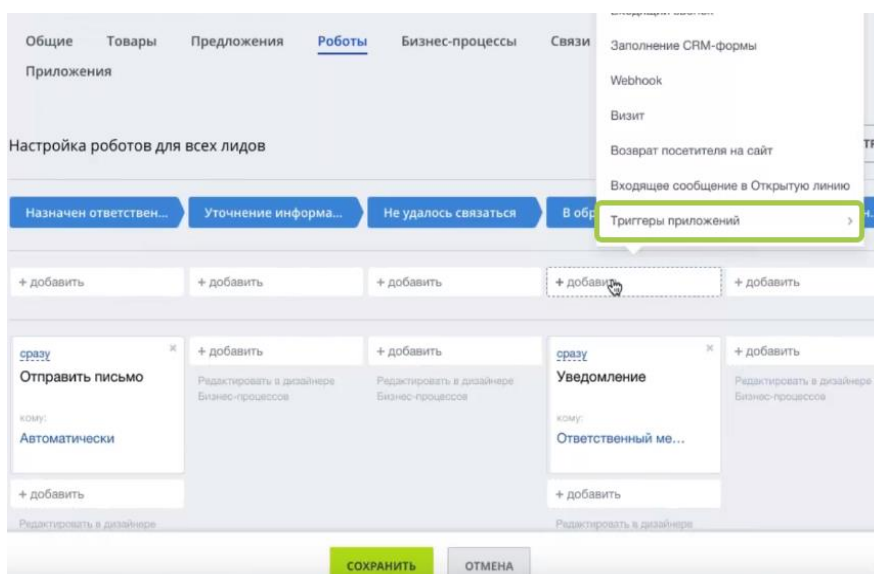


Рисунок 1 – Пример добавления триггера в CRM

- Тип клиента (группа, к которой принадлежит клиента);
- Уникальный идентификатор клиента.

В том случае, если клиент переходит по указанной ссылке, он попадает на страницу организации. В этот момент страница получает данные этого клиента. Эти данные обрабатываются и с помощью ajax-запроса отсылаются во «внешнее приложение». Это приложение в последствии и будет взаимодействовать с данными и использовать различные REST запросы для манипулирования CRM.

Если приложение получило все необходимые данные, тогда через REST API выполняется запрос в CRM. Это и есть событие. Чтобы событие сработало правильно, необходимо передать минимальный набор параметров:

- Идентификатор триггера (символьный код);
- Данные клиента.

Пример события показан на рисунке 2.

CRM обрабатывает событие и вызывает указанный триггер. В свою очередь триггер начинает выполнять свой алгоритм. В данном случае триггер переводит клиента на другой этап бизнес-процесса и создает нотификацию о том, что конкретный клиент, в данный момент времени, просматривает страницу с коммерческим предложением. После этого менеджер, который отвечает за данного клиента получает нотификацию о действиях клиента. Менеджер связывается с этим клиентом и начинает предлагать свои услуги, либо свою помощь.

```
$result = restCommand('crm.automation.trigger.execute', //событие
    array(
        'CODE' => 'символьный_код_триггера',
        'OWNER_TYPE_ID' => intval($_REQUEST['TYPE']), // тип клиента в CRM
        'OWNER_ID' => intval($_REQUEST['ID']) // идентификатор клиента в CRM
    ),
    array('domain' => $_REQUEST['DOMAIN'], 'access_token' => $_REQUEST['AUTH_ID']) //конфигурации приложения
);
```

Рисунок 2 – Пример отправки события с помощью REST запроса

Этот простой алгоритм применения триггеров представлен на рисунке 3.

Это далеко не единственно возможный способ применения роботов, триггеров и REST запросов. Эти инструменты являются огромной средой для творческой деятельности разработчиков.

Для подобных целей так же подходят Вебхуки, принцип работы схож, однако технологии разные, а соответственно и способы достижения поставленной цели будут другими.

Работа с REST не простая задача, которая требует должной квалификации разработчика. Однако при умелом использовании REST, при грамотной настройке бизнес-процессов, роботов, событий, триггеров и сторонних приложений позволяет максимально упростить работу менеджера при ведении клиента по воронке продаж, от статуса лица, до заключения с ним сделки.



Рисунок 3 – Пример применения триггеров

Список используемых источников

1. CRM Битрикс24. [Электронный ресурс]. – Режим доступа: <https://www.bitrix24.by/>. – Дата доступа: 04.03.2018.
2. Создание приложений Битрикс24 [Электронный ресурс]. — Режим доступа: <https://www.bitrix24.ru/apps/dev.php> – Дата доступа: 04.03.2018.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОБУЧЕНИЯ ОСНОВАМ КРИПТОГРАФИИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Лупач В.Д.

Калитеня И.Л. – ассистент каф ИСиТ, м.т.н.

Сегодня едва ли удастся найти компанию, на компьютерах которой не обрабатывалась и не хранилась бы важная коммерческая информация. Полноценная система ее защиты имеет исключительное значение и без криптографии здесь никак не обойтись. Применительно к компаниям шифрование обычно используется в трех случаях, а именно в целях обеспечения безопасности хранения данных, защиты информации при передаче через открытые каналы связи и по локальной сети, наконец, для цифровых подписей. Все представленные задачи решаются разными средствами с применением различных криптографических технологий.

В современных корпоративных сетях вся важная информация сосредоточена на одном или нескольких серверах. Такой подход по сравнению с хранением данных на компьютерах конечных пользователей имеет множество преимуществ. Первостепенной задачей здесь является обеспечение надежного шифрования документов именно на сервере, но при этом система безопасности не должна серьезно осложнять работу сотрудников с информацией, которая необходима для исполнения служебных обязанностей.

Знание криптографических методов важно для развития и профессионального роста программистов в современном мире. Программисты с подобными знаниями смогут профессионально заниматься защитой корпоративной информации.

В дипломном проекте ставилась задача создать программное средство, которое могло бы обучать практическим основам криптографии и проверять знания учащихся. Ставилась задача создать программное средство, которое помогало бы в процессе обучения, а также в котором учащийся смог бы самостоятельно изучить пропущенные темы. Процесс обучения, как и процесс проверки знаний, проходит в несколько этапов. В режиме обучения, при неправильных действиях учащегося, подсказки в каждом этапе помогут найти и исправить свою ошибку. При проверке знаний, учащийся будет иметь возможность проверить свой ответ и продемонстрировать процесс решения.

Программное средство имеет следующие функции:

- авторизация учащегося;
- выбор режима администратора;
- выбор криптографического метода;
- реализация двух методов: обучения и проверки знаний;
- реализация возможности ввода любых текстов для зашифровки и любых возможных ключей;
- помощь и подсказки при обучении;
- проверка знаний по какому-либо криптографическому методу;
- сохранение оценки пользователя в базе данных;
- сохранение в базе данных времени, затраченного на выполнение учащимся задания;
- просмотр оценок учащихся администратором.

Список использованных источников:

1. Криптографические основы безопасности. [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/department/security/networksec/> - Дата доступа: 27.12.2017

ОСОБЕННОСТИ КЭШИРОВАНИЯ КОМПОНЕНТОВ В UNITY

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Мультан Р.И. Мазур А.Д.

*Бакунова О.М., ст. преподаватель каф ИСиТ, м.т.н.
Образцова О.Н., доцент каф. ИСиТ, к.т.н., доцент*

Огромная часть Unity-разработчиков понимают, что не стоит злоупотреблять дорогими для производительности вычислениями, к таким вычислениям относится и получение компонентов. Чтобы избежать дорогих вычислений необходимо использовать кэширование. В докладе рассмотрены различные способы кэширования, их реализация, особенности и производительность.

Данная статья будет рассматривать в основном внутреннее кэширование — это процесс получения различных компонентов на объекте. Объект в Unity, находящийся на сцене, представляет собой GameObject - т.е. вместилище для разнообразных компонентов. Компоненты бывают стандартными (Rigidbody, RectTransform, Animator), так бывают и написанными разработчиками (класс, который наследуется от MonoBehaviour). Для того, чтобы получить компонент на объекте можно использовать метод GetComponent(), но не стоит им сильно злоупотреблять. Если компонент используется в скрипте не единожды, существует подход при котором возможно объявить его переменную и получить его компонент с объекта всего лишь один раз с помощью метода GetComponent() (рисунок 1)

```
public class CachingExample : MonoBehaviour
{
    private Rigidbody rigidbody;

    void Start()
    {
        rigidbody = GetComponent<Rigidbody>();
    }

    void Update()
    {
        rigidbody.AddForce(Vector3.up * Time.deltaTime);
    }
}
```

Рисунок 1 – Объявление компонента, как переменную.

Заметим, что прямое получение компонентов на объекте более производительнее, чем остальные способы кэширования, в большинстве случаев. Другие вариации кэширования опираются на базовое и упрощают нам доступ к компонентам, лишая нас необходимости писать один и тот же код, когда нам необходим какой-либо компонент.

Для следующего метода кэширования используются свойства. Свойство - это разновидность члена класса, предоставляющий гибкий механизм для чтения, записи и вычисления значения частного поля. Явным преимуществом этого метода является то, что кэширование произойдет только тогда, когда мы первый раз обратимся к свойству. Явным недостатком является то, что нам придется писать больше однообразного кода (рисунок 2).


```

public class PropertyCachingExample : MonoBehaviour
{
    private Rigidbody rigidbody;
    private bool isRigidbodyCached = false;

    public Rigidbody CachedRigidbody
    {
        get
        {
            if (!rigidbody)
            {
                isRigidbodyCached = true;
                rigidbody = GetComponent<Rigidbody>();
            }
            return rigidbody;
        }
    }
}

```

Рисунок 2 – Использование для хеширования свойств

Облегчить себе работу можно унаследовав классы от компонента, который кэширует употребляемые свойства. Данный метод не универсален. Для оптимального разрешения проблемы универсальности необходимо использовать шаблоны (рисунок 3).

```

public class InnerCache : MonoBehaviour
{
    Dictionary<Type, Component> cache = new Dictionary<Type, Component>();

    public T Get<T>() where T : Component
    {
        var type = typeof(T);
        Component item;
        if (!cache.TryGetValue(type, out item))
        {
            item = GetComponent<T>();
            cache.Add(type, item);
        }
        return item as T;
    }
}

```

Рисунок 3 – Пример шаблона

В языке программирования С# есть особенность под названием “методы расширения”. Данные методы позволяют нам добавлять свои методы в уже существующие типы без создания нового типа. Данный метод кэширования реализуется следующим способом:

```

public static class ExternalCache
{
    static Dictionary<GameObject, TestComponent> test = new Dictionary<GameObject, TestComponent>();

    public static TestComponent GetCachedTestComponent(this GameObject owner)
    {
        TestComponent item = null;
        if (!test.TryGetValue(owner, out item))
        {
            item = owner.GetComponent<TestComponent>();
            test.Add(owner, item);
        }
        return item;
    }
}

```

Рисунок 4 – Использование особенностей «методы расширения» языка программирования С# .

После реализации метода расширения компонент можно будет получить из любого скрипта таким образом — `gameObject.GetCachedTestComponent()`. Таким образом, написав один раз метод расширения, мы избавляем себя от написания единообразного кода и инкапсулируем внутреннюю реализацию получения компонента у `gameObject`.

Недостатком этого варианта является постоянное слежение за мертвыми ссылками. Если за ними не следить (т.е. не чистить кэш), объем кэша будет расти и загружать память ссылками на уничтоженные объекты. Пока что самым производительным вариантом кэширования остается базовое кэширование — получение компоненты объекта с помощью `GetComponent()` при инициализации объекта. Зачастую окольные пути не совсем оптимальны и требуют от разработчика написания лишних строчек кода.

Теперь рассмотрим другой способ, в котором используются атрибуты. Атрибут представляет собой специальный инструмент, который позволяет вставить в сборку дополнительные метаданные. Метаданные - это данные в двоичном формате с описанием программы, хранящиеся либо в памяти, либо в переносимом хранилище файле. Атрибуты сами по себе не выполняются и их необходимо использовать с помощью рефлексии, хоть рефлексия и достаточно дорогая операция. Рефлексия представляет собой процесс вскрытия типов во время выполнения приложения. Также можно использовать самописный атрибут для кэширования, а именно `[AttributeUsage(AttributeTargets.Field)] private class CacheAttribute:Attribute (){}`. Далее пользоваться им для всех переменных класса `[Cached] private TestComponents Test`. После этого необходимо реализовать класс, который способен принимать переменные классов с их атрибутами, а также подхватывать их при инициализации (рисунок 5).

```
public class AttributeCacheInherit : MonoBehaviour
{
    protected virtual void Awake()
    {
        CacheAll();
    }

    void CacheAll()
    {
        var type = GetType();
        CacheFields(GetFieldsToCache(type));
    }

    List<FieldInfo> GetFieldsToCache(Type type)
    {
        var fields = new List<FieldInfo>();
        foreach (var field in type.GetFields())
        {
            foreach (var a in field.GetCustomAttributes(false))
            {
                if (a is CacheAttribute)
                {
                    fields.Add(field);
                }
            }
        }
        return fields;
    }

    void CacheFields(List<FieldInfo> fields)
    {
        var iter = fields.GetEnumerator();
        while (iter.MoveNext())
        {
            var type = iter.Current.FieldType;
            iter.Current.SetValue(this, GetComponent(type));
        }
    }
}
```

Рисунок 5 – Пример вставки в сборку дополнительных метаданных

При создании наследуемого класса можно получать его переменные помощью атрибута `[Cache]`, это позволит не думать о кэшировании. При этом можно использовать статический класс, с помощью которого при выполнении программы мы можем достать компоненту только единожды, и в дальнейшем ее использовать. Тогда и использование довольно сильно влияющей на производительность рефлексии нам не понадобится. Чтобы закэшировать типы нам нужно добавить строчку `CacheHelper.CacheAll(this)` в один из методов инициализации (методы `Awake` или `Start`). Теперь все, что было помечено в этом классе атрибутом `[Cache]`, будет получено при вызове метода `GetComponent()`.

Компоненту можно и не кэшировать в уже запущенном приложении, для этого можно воспользоваться редактором, перед запуском проекта. По функциональности такой метод кардинально не выделяется среди остальных, за исключением того, что все это происходит непосредственно перед запуском редактора. Эта функция кэширует компоненты для их дальнейшего использования. Хотя в этом варианте есть свои особенности. В конце статьи хотелось бы пройтись по тому, что было затронуто. Были рассмотрены разнообразные методы кэширования компонент, применения атрибутов. Функции, в которых основой является рефлексия. Такие методы могут быть использованы при создании Unity-приложений, если знать и учитывать его особенности. Один из методов позволяет писать меньше единообразного кода, но его производительность немного проигрывает “прямолинейному” решению. Другой на сегодняшний день требует немного больше внимания, но при этом не затрагивает конечную производительность.

Например, он не нуждается в ресурсах на явную манипуляцию, объекты нужно подготавливать явно, при приготовлении объекты обязаны находится на сцене, в шаблоны и объекты не должны вноситься никаких изменений на других сценах.

ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ ТЕКСТУР НА ОСНОВЕ АЛГОРИТМА DIAMOND-SQUARE

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Мультан Р.И. Мазур А.Д.

Бакунова О.М. - ст. преподаватель каф. ИСиТ, м.т.н.
Бакунов А.М. - ст. преподаватель каф. ИСиТ, м.т.н.
Образцова О.Н. - доцент каф. ИСиТ, к.т.н., доцент
Калитеня И.Л. - ассистент каф. ИСиТ, м.т.н..

В данной статье рассмотрен алгоритм diamond-square(далее DS), его практические реализации. Также будет рассмотрено использование этого алгоритма для того, чтобы процедурно сгенерировать текстуру.

Алгоритм DS представляет собой расширенную версию алгоритма midpoint displacement. Расширенная реализация заключается в использовании двухмерной плоскости и наличием 2 этапов (рисунок 1). Первый этап— “square” — на данном этапе каждый элемент массива — ромб, для него определяется центральная точка, для которой считается среднее значение из крайних точек и добавляется случайное смещение. Вторым этапом — “diamond” — каждому квадрату в массиве определяется срединная точка, которой устанавливается среднее значение угловых точек и добавляется случайное смещение. На каждом этапе и при каждой итерации случайное отклонение, которое прибавляется к срединным точкам, уменьшается. На выходе у него получается карта высот, точки в ней расположены по сетке. Таким образом, выходит, что вся плоскость покрыта квадратами.

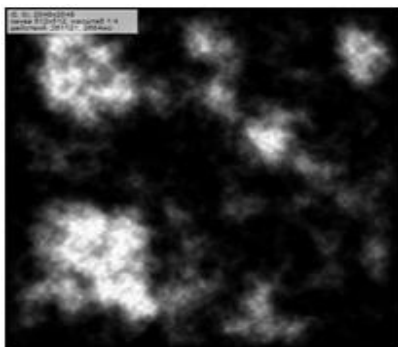


Рисунок 1 – Реализация в двухмерной плоскости алгоритма DS

С помощью алгоритма DS генерируют весьма реалистичные ландшафты, которые называют фрактальными. Но при максимальном увеличении можно заметить небольшое количество мелких островков вместо равнин и морей. Разрешить такого рода проблему можно комбинируя фрактальные ландшафты различной размерности. Значения таких ландшафтов можно складывать, перемножать, используя различные коэффициенты. Чтобы сделать равнины более пологими, а склоны гор более крутыми, можно нормализованные значения возводить в квадрат. Можно экспериментировать не стесняясь.

Для того, чтобы создать процедурно сгенерированную текстуру существует несколько шагов. В основе генерации лежит вышеописанный алгоритм DS, который поможет нам получить карту высот. Сначала необходим двумерный массив размерностью $2^n + 1$. Далее необходимо реализовать два основных шага DS – square и diamond (рисунок 2).

```
public static int ysize = 1025, xsize = ysize * 2 - 1;
public static float[,] heighmap = new float[xsize, ysize];
public static float roughness = 2f;

public static void Square(int lx, int ly, int rx, int ry)
{
    int l = (rx - lx) / 2;

    float a = heighmap[lx, ly]; // B-----C
    float b = heighmap[lx, ry]; // |         |
    float c = heighmap[rx, ry]; // |   ce   |
    float d = heighmap[rx, ly]; // |         |
    int cex = lx + l; // A-----D
    int ceY = ly + l;

    heighmap[cex, ceY] = (a + b + c + d) / 4 + Random.Range(-1 * 2 * roughness / ysize, 1 * 2 * roughness / ysize);
}
```

Рисунок 2 - Создание процедурно сгенерированной текстуры

Square будет принимать две противоположные точки квадрата — левую нижнюю и правую верхнюю, и потом сохранять значение в центр квадрата. Следующий метод — diamond — будет принимать значение точки, у которой нужно вычислять центральную точку сторон квадрата. Далее вычисляет точку на базе значений перекрестных углов квадрата, центра и центральной точки соседнего квадрата. Ниже показана реализация одного из шагов DS:

Далее необходимо проверить координаты всех точек, не выходят ли они за пределы массива. Когда это все же случается, то точке необходимо присвоить обратное ей значение (рисунок 3). После этого если попробовать нанести текстуру на шар, то получается бесшовная текстура. Для следующего шага нам необходимо написать небольшой метод объединяющий их в квадрат.

```
public static void DiamondSquare(int lx, int ly, int rx, int ry)
{
    int l = (rx - lx) / 2;

    Square(lx, ly, rx, ry);

    Diamond(lx, ly + l, l);
    Diamond(rx, ry - l, l);
    Diamond(rx - l, ry, l);
    Diamond(lx + l, ly, l);
}
```

Рисунок 3 – Проверка координаты всех точек и объединение в квадрат

При использовании этого метода не используется рекурсия. Потому что при использовании метода DiamondSquare(){} только первый квадрат будет рассчитываться правильно, так как точки находятся за размерностью массива, поэтому иногда вместо этого используется центр квадрата (рисунок 4).

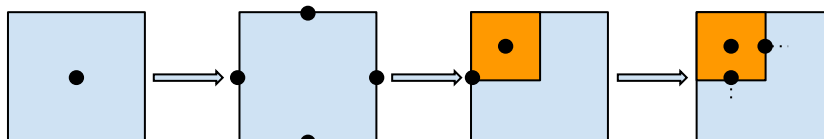


Рисунок 4 – Варианты объединения точек в квадраты

Последующие квадраты, созданные внутри, верно считаться уже не будут, потому что центры окружающих квадратов еще не подсчитаны. И получается, что использовать рекурсию не имеет смысла. Использовать надо обсчет по слоям. Он заключается в том, что надо просмотреть все длины сторон квадратов, и в каждом необходимо рассмотреть левые нижние углы в текущем слое. Выходит, что мы всегда будем знать центральные точки окружающих квадратов.

В следующем шаге необходимо перейти к цвету. Генерировать цвет мы будем при помощи midpoint displacement. В текущем этапе он будет использоваться напрямую. Т.е. он будет содержать список координат по оси y, которые сопоставлены координате по оси x. Ось x является границей пояса. Для придания поясу замкнутости необходимо делать края равными. Для пуцего реализма необходимо делать различные значения roughness, таким образом, чтобы пояса могли двигаться по неожиданным траекториям. Но тут важно не зайти слишком далеко и не увлечься. Ведь выходная сгенерированная текстура должна быть приближена к реальной.

Необходимо будет инициализировать массив цветов (для этого в Unity существует метод, позволяющий читать массив, а также записывать значения элементов в текстуру) (рисунок 5). Для того чтобы сделать текстуру более реалистичной, можно возвести ее значения в степень, что придаст ей более гладкий и пологий вид. Ну и после всех этапов остается только чуть сгладить картинку и использовать нашу текстуру.

```
private Texture2D tex;
public static Color[] colors = new Color[Heighmap_class.xsize * Heighmap_class.ysize];

float waterLvl = 0.2f;
```

Рисунок 5 – Использование текстуры

Подводя итог этой статьи, необходимо понимать, что без алгоритма DS получить процедурно сгенерированную реалистичную текстуру никак не получится. Этот алгоритм может быть использован для генерации реалистичных ландшафтов, генерации миров, а также и для генерации бесконечных уровней.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ВЕДЕНИЯ УЧЕТА ГРУЗОПЕРЕВОЗОК

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Малашенко А.Ю.

Калитеня И. Л. – ассистент каф. ИСиТ, м.т.н.

В докладе рассмотрен комплекс вопросов, связанных с созданием, тестированием и внедрением современной системы ведения учета и управления грузоперевозками [1].

Автоматизированные системы в настоящее время все чаще внедряются в современные логистические процессы, с целью ускорения времени доставки груза до потребителя и улучшения качества обслуживания клиентов [2]. Проанализировав наиболее успешно работающие аналоги на рынке, были определены их достоинства и недостатки и учтены в разработке программного средства.

Создаваемое программное средство должно быстро и корректно выполнять допустимые действия пользователей системы, рассчитывать оптимальные пути доставки груза, демонстрировать графически местонахождение груза на карте, рассчитывать загрузку транспортного средства товарами, регистрировать в системе новые транспортные средства и т.д.

Главной особенностью программного средства является разделение пользователей системы по ролям, т.е. каждый пользователь имеет свой интерфейс для работы с программным средством.

Разработанное программное средство имеет следующие функции:

- оформление заявки на транспортировку груза;
- создание заказа на доставку груза;
- возможность регистрирования в системе транспортных средств;
- возможность учета клиентской базы;
- создание матрицы биллинга;
- планирование маршрута грузоперевозки;
- отслеживание транспортного средства;
- возможность поиска и фильтрации данных;
- возможность назначения транспортных средств на водителей;
- расчет загруженности транспортного средства;

Особенности процесса оформления заявки на поставку груза описывается на рисунке 1.

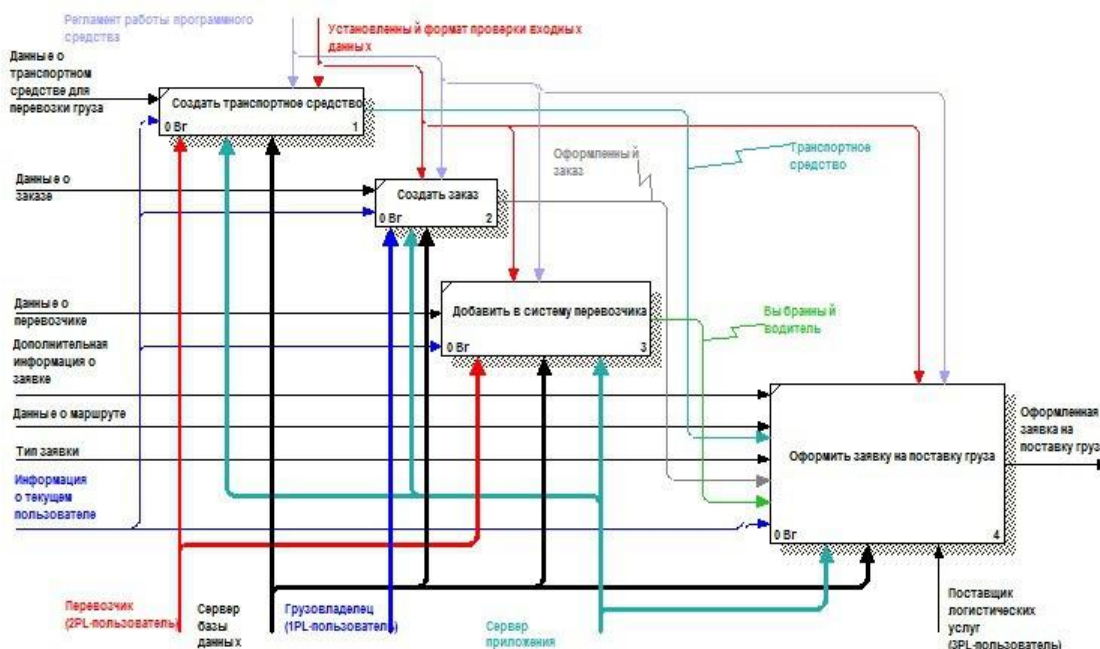


Рисунок 1 - Процесс оформления заявки на доставку груза.

Данное программное средство основано на клиент-серверной архитектуре [1]. Взаимодействие с серверной частью осуществляется по протоколу HTTP. Серверная часть отвечает за бизнес-логику и систему управления базой данных. Схема работы программного средства и её структура описана на рисунке 2.

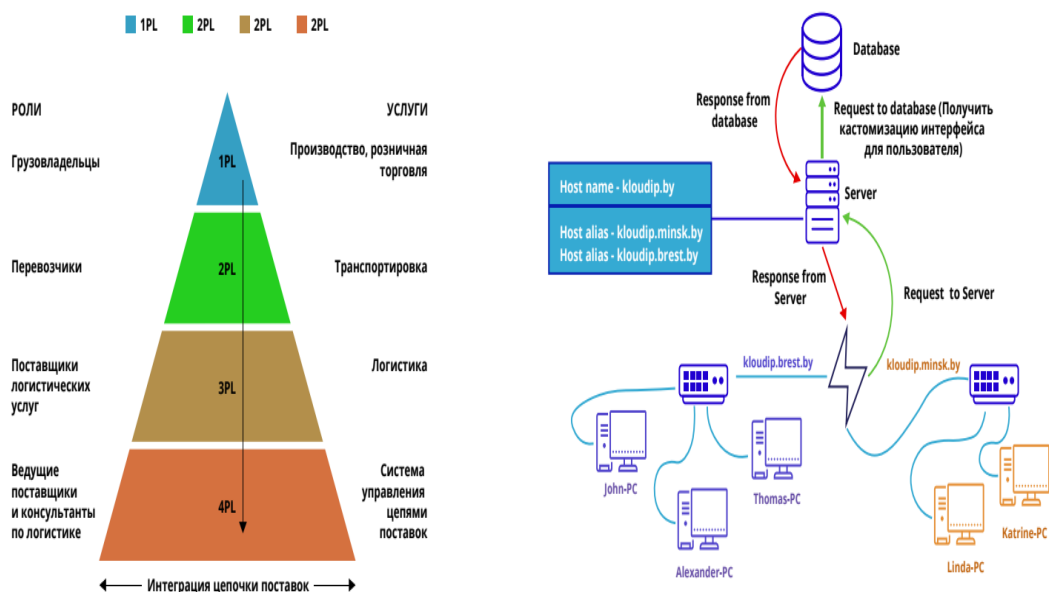


Рисунок 2 - Схема работы программного средства и её пользовательская структура.

Клиентская часть приложения написана на языке JavaScript, использует также фреймворк JQuery и библиотеку Leaflet.js. Серверная часть написана на языке PHP [1]. Используемая система управления базой данных приложения PostgreSQL.

Разработанное программное средство положено в основу работы логистической компании.

Список использованных источников:

1. PHP 7 - Руководство по использованию [Электронный ресурс]. – Режим доступа к ресурсу: <https://www.tutorialspoint.com/php7/>. – Дата доступа: 10.01.2018.
2. Логистические процессы и значимость [Электронный ресурс]. – Режим доступа к ресурсу: <https://cyberleninka.ru/article/v/logisticheskie-prots-essy-i-ih-znachimost-na-predpriyatii/>. – Дата доступа: 10.01.2018.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ УПРАВЛЕНИЯ ЗАДАЧАМИ И ОТСЛЕЖИВАНИЯ ОШИБОК ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Мисюра А.Ф.

Маковский М.Л. – и.о. декана ФКТ

Промышленное применение компьютеров и мобильных устройств и растущий спрос на программное обеспечение поставили актуальные задачи существенного повышения производительности при разработке программного обеспечения, разработки промышленных методов планирования и проектирования программ, переноса организационно-технических, технико-экономических и социально-психологических приемов, закономерностей и методов из сферы материального производства в сферу применения информационных технологий [1].

Системы управления задачами и отслеживания ошибок помогают оптимизировать работу команды, повышают дисциплину в команде (так как всегда можно отследить, сколько времени было потрачено на выполнение задачи), увидеть проблемные места в процессах разработки. Они включают в себя логику распределения приоритетов, позволяя разработчикам самостоятельно выбирать себе наиболее важные задачи. Показатели эффективности позволяют менеджерам проектов лучше понять, сможет ли команда соответствовать установленным срокам и анализировать возможные причины отклонений от графиков реализации проекта, а также учитывать и контролировать пожелания пользователей, и следить за процессом выполнения или невыполнения пожеланий. [2]

Использование такого программного средства является необходимым условием для эффективного распределения рабочего времени сотрудников. В современном мире разработка серьезного и крупного проекта невозможна без использования систем, осуществляющих управление задачами и отслеживание ошибок, так как это добавляет определенные риски по планированию рабочей нагрузки, распределению ресурсов, контролю процесса разработки, а также подтверждению выполненного объема задач.

Целью дипломного проекта является разработка клиент-серверного программного средства, предназначенного для обеспечения контроля качества разрабатываемого программного обеспечения. Задачей является непосредственно обеспечение и повышение качества разрабатываемого программного обеспечения.

Для достижения поставленной цели были проанализированы источники литературы, были рассмотрены существующие аналоги на рынке систем управления задачами, учтены их достоинства и недостатки; изучены теоретические основы управления разработкой программного обеспечения и тестирования программного обеспечения.

Для реализации проекта были выбраны языки Java и Javascript. В качестве СУБД среди аналогов была выбрана PostgreSQL. Также в разработке были применены java-Фреймворки Hibernate и Spring. На клиентской части приложения использованы библиотеки Bootstrap и jQuery.

Результатом является разработанное и готовое к использованию веб-приложение, выполняющее поставленную выше задачу – обеспечение и повышение качества разрабатываемого программного обеспечения. Программное средство состоит из клиентской и серверной части. Использование сервера обусловлено задачей хранения и обмена данными между клиентами. Обмен происходит посредством HTTP-запросов. База данных на сервере отвечает за хранение информации о задачах и дефектах, о пользователях, о прикрепленных файлах и др. Предусмотрены две категории пользователей системы: авторизованные пользователи (разработчики, тестировщики, бизнес-аналитики и др.) и администраторы системы. Программное средство может работать на любых устройствах (мобильных телефонах, планшетах и т.д.) благодаря использованию адаптивной верстки на клиентской части.

Список использованных источников.

1. Фримен, Э. Паттерны проектирования / Э. Фримен, К. Сьерра – СПб.: Питер, 2015. – 656 с.
2. Канер, С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложения / С. Канер, Дж. Фолк, Е. К. Нгуен. – К: ДиаСофт, 2001. – 544 с.

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ФОРМИРОВАНИЯ ЗАКАЗОВ И ПРОДАЖИ СТРОИТЕЛЬНЫХ МАТЕРИАЛОВ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Наливко В.Н.

Калитеня И. Л. – ассистент каф. ИСиТ, м.т.н.

В данной работе рассматривается разработка веб-приложения для формирования заказов и продажи строительных материалов. Тема проекта актуальна в связи с необходимостью внедрения процессов электронной коммерции при формировании заказов и продажи стройматериалов.

Электронная коммерция - это сфера экономики, которая включает в себя все финансовые и торговые транзакции, осуществляемые при помощи компьютерных сетей, и бизнес-процессы, связанные с проведением таких транзакций [1].

Электронная коммерция имеет ряд преимуществ:

- отсутствие географических, временных и языковых барьеров, что позволяет продвигать товары и услуги на новые рынки сбыта;
- низкий уровень издержек производства и обращения, что достигается путем внедрения новых технологий во все сферы деятельности организации;
- более высокий уровень конкуренции перед традиционной торговлей;
- большой ассортимент различных товаров и предоставляемых услуг;
- возможность получения дополнительной прибыли за счет предоставления рекламных мест.

Веб-приложение для формирования заказов и продажи строительных материалов представляет собой клиент-серверное приложение, позволяющее пользователям реализовать доступ к бизнес-логике с помощью браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется на сервере, обмен информацией осуществляется по сети. Преимуществом такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами. На рисунке 1 показана общая схема веб-приложения.

Проанализировав существующие аналоги, веб-приложение должно иметь следующие критерии:

- пользовательский интерфейс, который обеспечивает корректную передачу информации между пользователем и программно-аппаратными компонентами компьютерной системы;
- адаптивный веб-дизайн, обеспечивающий правильное отображение сайта на различных устройствах, подключенных к интернету и динамически подстраивающийся под заданные размеры окна браузера;
- личный кабинет пользователя;
- панель администратора, предоставляющая полный доступ управления веб-приложением;
- наличие актуальной информации и услуг;
- возможность динамического добавления контента;
- возможность формирования и оформления заказа;

- удобство навигации должно быть организовано таким образом, чтобы у пользователя не было затруднений перехода из одной части страницы в другую;
- каталог товаров должен обеспечивать понятность отображения реализуемой продукции;
- корзина покупок даст посетителю возможность отложить товар в корзину, просмотреть корзину перед оформлением заказа, а также удалить ненужные товары.

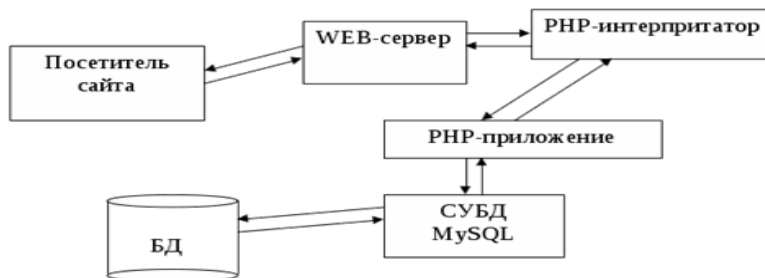


Рисунок 1 - Общая схема веб-приложения

Основным назначением данного программного средства является создание механизма, обеспечивающего привлечение новых клиентов и расширение рынков сбыта продукции, а также предоставляющего необходимую информацию о товарах и услугах предприятия и способах ее реализации, посредством использования современных технологий. Под структурой веб-приложения понимается систематизация информации таким образом, чтобы пользователь, придя впервые на сайт, мог за короткий период времени ознакомиться с существующей информацией и приобрести интересующий его товар. Разработка структуры сайта велась с учётом наиболее важной информации. В процессе создания структуры сайта необходимо акцентировать внимание пользователей именно на той информации и разделах сайта, которые наиболее важны в соответствии с позиционированием продвигаемых товаров и услуг. Разработка структуры веб-приложения состоит из программной части, клиентской части и администрирования, как показано на рисунке 2.



Рисунок 2 – Структура веб-приложения

Программная часть структуры веб-приложения рассматривается как взаимосвязь серверной и операционной части. Панель администратора содержит инструменты управления веб-приложением. Для создания гибкого пользовательского интерфейса клиентской части использовался модуль Flexbox Layout (Flexible Box). Flexbox Layout – это модуль CSS, который направлен на предоставление более эффективного способа расположения, выравнивания и распределения свободного пространства между элементами в контейнере [2]. Использование Ajax позволило перезагружать не весь сайт, а только лишь необходимую часть страницы. Использование формы обратной связи повысило качество предоставляемых услуг и производства продукции.

Главной при разработке больших систем считается задача снижения сложности. Сложная система строится из небольшого количества более простых подсистем, каждая из которых, в свою очередь, строится из частей меньшего размера. Архитектура веб-приложения идентифицирует главные компоненты системы и способы их взаимодействия. Веб-приложение для формирования заказов и продажи строительных материалов разработано с помощью шаблона MVC (Model View Controller) [3]. Шаблон разделяет работу веб-приложения на три отдельные функциональные роли:

- модель данных (model) - содержат бизнес-логику приложения и включают методы выборки, обработки и предоставления конкретных данных;
- пользовательский интерфейс (view) – представления, реализованные в проекте, используются для задания внешнего отображения данных, полученных из контроллера и модели. Представления содержат HTML-разметку и вставки РНР-кода, использующегося для получения готовых к выводу данных за счет прямого обращения к свойствам и методам контроллера или модели. Представления разделены на общий шаблон, содержащий разметку, общую для всех страниц (верхнюю и нижнюю часть), и части шаблона, которые используют для отображения данных выводимых из модели и отображения форм ввода данных;

– управляющую логику (controller) - связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя.

Таким образом, изменения, вносимые в один из компонентов, оказывают минимально возможное воздействие на другие компоненты. В данном паттерне модель не зависит от представления или управляющей логики, что делает возможным проектирование модели как независимого компонента и, например, создавать несколько представлений для одной модели.

Одной из главных составляющих проекта является база данных. Целью базы данных является представление в объективной форме совокупности самостоятельных материалов, систематизированных таким образом, чтобы эти материалы были найдены и обработаны с помощью электронной вычислительной машины. Для построения БД использовался phpMyAdmin. Для получения данных из БД использовался специальный язык запросов к базам данных (SQL). Через SQL можно запросить данные в различном виде, в нужной сортировке и с нужным фильтром, изменять данные в БД, удалять и создавать новые записи. Реализация веб-приложения включает в себя HTML, CSS, jQuery, JavaScript, PHP решения, а также использование фреймворка Bootstrap. Каждый модуль сайта отвечает за решение определенной задачи.

В результате выполнения данного проекта было разработано веб-приложение для формирования заказов и продажи строительных материалов. Разработанное программное средство имеет комплекс гибких настроек, что позволяет легко масштабировать данную систему и расширить область ее применения.

Список использованных источников.

1. Электронная коммерция – Lpgenerator [Электронный ресурс]. – Режим доступа: <https://lpgenerator.ru/blog/2015/07/23/cto-takoe-elektronnaya-kommerciya-v-internete/>. – Дата доступа: 29.01.2018.
2. Описание всех CSS свойств – Flexbox [Электронный ресурс]. – Режим доступа: <http://html5.by/blog/flexbox/>. - Дата доступа: 29.01.2018.
3. MVC в web-программировании: проще некуда [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/181772/>. – Дата доступа: 29.01.2018.

КОРПОРАТИВНЫЕ ПОРТАЛЫ НА ПРЕДПРИЯТИИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Нестерович Р.В.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Куликовский Д. В. – ассистент каф. ПЭ*

В современном мире всё больше руководителей осознают важность внедрения в работу предприятия корпоративных порталов как необходимого инструментария для успешного управления бизнесом. В докладе представлен анализ проблем возникающих при реализации корпоративных порталов.

Понятие корпоративного портала многогранно: с одной стороны, это веб-интерфейс к корпоративной информационной системе (КИС) компании, с другой стороны, это система распределения корпоративной информации, с третьей, автоматизированная система управления отдельными бизнес-процессами, а при увеличении числа управляемых бизнес-процессов корпоративный портал может приближаться к КИС, которая в данном случае строится с применением интернет-технологий. Сейчас корпоративный портал — это неотъемлемая часть новейших КИС, идеальная среда для приложений SCM (управление отношениями с поставщиками) и CRM (управление отношениями с клиентами).

Корпоративный портал — это программное обеспечение, предоставляющее сотрудникам компании клиентам и простым пользователям доступ к различной информации из внутренних и внешних сетей с целью организации производственной деятельности в соответствии с имеющимся уровнем прав. Корпоративный портал, как правило, предоставляет возможности внутренних и внешних коммуникаций, интеграции сторонних приложений.

Реализация основных функций корпоративных порталов позволяет предприятиям эффективно решать целый ряд задач:

1. Организация единой среды для осуществления всех типов коммуникаций и совместной работы сотрудников.
2. Автоматизация управления потоками работ и процессов документооборота.
3. Эффективный поиск информации.
4. Разграничение прав доступа к информационным узлам портала.
5. Возможность интеграции портала с корпоративными бизнес-приложениями и внешними системами.

Корпоративные порталы как инструменты управления бизнесом обеспечивают эффективное управление бизнес-процессами и коммуникациями в компании, что создает конкурентные преимущества – снижение издержек, повышение управляемости и гибкости.

Как уже отмечалось ранее корпоративные порталы идеальная среда для приложений SCM (управление отношениями с поставщиками) и CRM (управление отношениями с клиентами).

SCM(Supply Chain Management) – управленческая концепция и организационная стратегия, заключающаяся в интегрированном подходе к планированию и управлению всем потоком информации о сырье, материалах, продуктах, услугах, возникающих и преобразующихся в логистических и производственных

процессах предприятия, нацеленном на измеримый совокупный экономический эффект (снижение издержек, удовлетворение спроса на конечную продукцию).

Система управления цепями поставок (SCM-система) — прикладное программное обеспечение, предназначенное для автоматизации и управления всеми этапами снабжения предприятия и для контроля всего товародвижения: закупку сырья и материалов, производство, распространение продукции. Существуют самостоятельные тиражируемые SCM-системы, решения, реализуемые как составная часть ERP-систем, а также уникальные системы, создаваемые для конкретного предприятия.

В составе SCM-систем обычно выделяется два крупных блока:

Планирование цепей поставок (Supply Chain Planning, SCP) – планирование и формирование календарных графиков, решения для совместной разработки прогнозов, проектирование сетей поставок, моделирование различных ситуаций, анализ установления и выполнения операций;

Исполнение цепей поставок (англ. Supply Chain Execution, SCE) – отслеживание и контроль выполнения логистических операций.

CRM-система (Customer Relationship Management или Управление отношениями с клиентами) – это прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путем сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

CRM-системы необходимы любому бизнесу, который работает напрямую с клиентами и стремится расширять число покупателей.

Наиболее распространённым поставщиком корпоративных решений на территории Беларуси является Битрикс24. Полный комплект инструментов для организации включает в себя: социальную сеть, решение задач и проектов, просмотр и редактирование документов онлайн, Битрикс24.Диск, планирование встреч с помощью календаря, CRM, HR: Управление персоналом.

Корпоративные порталы способны снабжать любую организацию разнообразным специфическим функционалом под потребности бизнеса. Существует возможность самостоятельной разработки индивидуальных и уникальных корпоративных порталов для работы, но более выгодным и простым методом станет готовое решение, настроенное под свою специфику.

Список использованных источников.

1. Михаил Елашкин, SAP Business One. Строим эффективный бизнес./Михаил Елашкин – КУДИЦ-Пресс, 2007.– 238 с.

2. Наэм Ариф. Интеграция SAP ERP "Учет и отчетность". Конфигурация и проектирование / Наэм Ариф, Шейх Мухаммед Таусеф – Эксперт РП, 2015. – 440 с.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ УПРАВЛЕНИЯ ЭЛЕКТРОННЫМИ АКТИВАМИ

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Пагин С.В.

Леванцевич В.А. – старший преподаватель каф. ПОИТ

В докладе представлены результаты разработки программного средства для управления электронными активами, решающее проблему хранения и доступа к электронным активам компании. ПС предоставляет веб-интерфейс, который позволяет обеспечить параллельный и независимый доступ пользователей системы к хранимому контенту.

В процессе проектирования был проведен анализ существующих аналогов и выявлены их преимущества и недостатки. [1]. На основе этого анализа было определено, какие функции необходимо реализовать в данном программном средстве. В экономическом разделе был проведен расчет экономической целесообразности разработки данного программного средства, из которого следует, что рентабельность инвестиций проекта составит 139%, а срок окупаемости – 4 года.

В дипломном проекте были реализованы следующие функции:

- импорт файлов в систему;
- добавление описания загруженного актива и назначение классификаций;
- поиск по метаданным и дереву классификаций;
- разграничение прав доступа;
- загрузка актива из системы на компьютер пользователя.

При проектировании баз данных была использована СУБД Microsoft SQL Server 2016. Данная СУБД представляет собой многофункциональный инструмент для создания, разработки и редактирования файлов баз данных, а также управления сервером базы данных. [2]. СУБД использует привычный табличный интерфейс и имеет функционал выполнения SQL-команды для проверки запросов.

Для разработки самого приложения использовалась среда разработки Microsoft Visual Studio 2017. Это интегрированная среда разработки программного обеспечения с поддержкой технологии IntelliSense и предоставляет редактор форм графического интерфейса, веб-редактор, дизайнер классов и дизайнер схемы базы данных. [3]. Данная среда имеет ряд преимуществ над конкурентами:

- высокая интеграция с продуктами Microsoft, в частности SQL Server и сервером веб-приложений IIS;

- встроенный SDK (software development kit);
- полнофункциональный конструктор интерфейсов.

Таким образом, в ходе дипломного проекта я изучил проблему управления электронными активами, проанализировал существующие аналоги и разработал собственное ПС, которое успешно выполняет поставленные задачи.

Список использованных источников:

1. Forrester Research - Digital Asset Management, 2017 [Электронный ресурс]. – 2017 – Режим доступа: <http://forrester.com/report/Digital+Asset+Management+2017/> – Дата доступа: 20.11.2017.

2. SQL Server 2016 Developer Edition [Электронный ресурс]. – 2016 – Режим доступа: <https://habrahabr.ru/post/280656/> – Дата доступа: 28.12.2017.

3. Visual Studio 2017 [Электронный ресурс]. – 2017 – Режим доступа: <https://docs.microsoft.com/en-us/visual-studio/> – Дата доступа: 25.12.2017.

ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ И ОЦЕНКИ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Пашковский Н.О.

Скудняков Ю.А. - доцент каф. ПЭ, к.т.н., доцент

В работе рассматриваются возможности разработанного программного средства (ПС) для осуществления оптимальной организации и оценки учебной деятельности студентов.

Основное назначение разработанного ПС – оптимизация организации тестирования знаний студентов и сбора необходимой статистики для подсчёта результатов по всем студенческим группам. Данное ПС используется преподавателями по дисциплине «Государственное управление и право» с целью ускорить процесс тестирования, дать детальный отчёт об уровне подготовки студентов и сохранить информацию для возможности подсчёта статистики по нескольким студенческим группам.

Аналогичные ПС такого рода зачастую имеют некоторые однотипные недостатки. К примеру, отсутствие возможности сохранения информации для дальнейшего использования, для организации чего может быть использована база данных (БД), а также невозможность вести организованный сбор данных сразу с нескольких машин в одну БД. На основе результатов анализа недостатков некоторых аналогичных ПС, в работе сделан вывод о необходимости их совершенствования в дальнейшем.

Для повышения эффективности организации тестирования знаний и сбора общей статистики по студенческим группам использование разработанного ПС позволяет проводить тестирование групп студентов по 12-24 человека, давать подробный отчёт об уровне подготовки как каждого студента в отдельности, так и группы в целом. Также имеется возможность сравнения результатов по группам студентов.

Разработанное ПС состоит из нескольких компонентов: клиент администратора/преподавателя, клиент для пользователей/студентов, БД для собранной статистики и тестовых заданий (которую можно дополнять по мере необходимости), веб-сервер для хранения данных (рисунок 1).

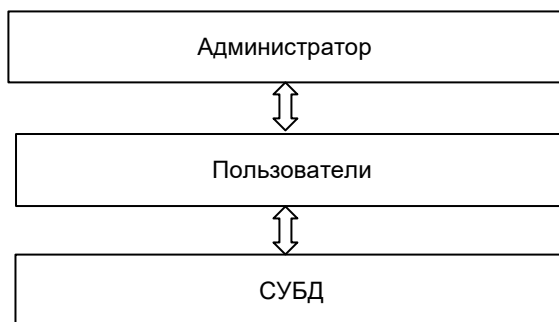


Рисунок 1 – Архитектура ПС

Связь между компонентами осуществляется посредством локальной или корпоративной сети, в зависимости от учреждения образования, в котором ПС будет использоваться.

В качестве веб-сервера ПС был использован MySQLserver, подключенный к компьютерной сети учреждения образования. Сервер хранит всю информацию в БД.

В качестве СУБД для данного ПС была выбрана MySQL. Использование данной СУБД обусловлено её свободной доступностью и наличием возможности по хранению данных в объектно-реляционном представлении [1].

Клиентская часть ПС была реализована посредством APIWindowsForms. Выбор обоснован тем, что данный интерфейс является частью Microsoft .NETFramework и его использование упрощает доступ к элементам MicrosoftWindows. Также, данный API достаточно просто связать с СУБД на базе MySQLserver [1].

Клиент состоит из логики отправки запросов к серверному приложению и предоставляет следующие возможности: 1) регистрация и аутентификация пользователя в системе; 2) возможность просмотра пройденных тестовых заданий; 3) возможность сравнения результатов по группе пользователей/студентов; 4) возможность сравнения результатов по нескольким группам пользователей/студентов.

Список использованных источников:

1.Wikipedia [Электронный ресурс]. – Режим доступа: <http://wikipedia.org> – Дата доступа: 15.03.2018.

РАЗВЕРТЫВАНИЕ МИКРОСЕРВИСОВ

*Институт Информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Петрович А.С.

*Образцова О.Н. – доцент каф. ИСиТ, к.т.н., доцент
Бакунова О.М. – ст. преподаватель каф. ИСиТ, м.т.н.
Бакунов А.М. - ст. преподаватель каф. ИСиТ, м.т.н.
Калитеня И.Л. - ассистент каф. ИСиТ, м.т.н.*

В работе проведен анализ использования микросервисов в программировании и особенности использования микросервисных архитектур.

Микросервисы – это маленькие независимые сервисы, которые совместно работают для достижения общих целей. Этот концепт не нов – он был придуман более десяти лет назад, но популярность и широкое распространение приобрёл в недалёком прошлом. Это произошло ввиду быстрого роста IT сферы, которое создало таких гигантов индустрии как Гугл, Макрософт, Нетфликс, Амазон и прочих. И эти гиганты осознали, что монолитные системы крайне невыгодны ввиду проблем с их масштабированием. Чтобы разрешить эту проблему была создана сервис-ориентированная архитектура. Но сегодня и её уже недостаточно – на смену SOA приходит микросервисная архитектура.

Микросервисы привносят не так много улучшений по сравнению с сервисами, потому что MSA не является новшеством само по себе. Тем не менее, каноническое их представление имеет множество преимуществ даже в сравнении с родительской архитектурой. Но достичь этого можно только чётко следуя принципам построения MSA (рисунок 1).

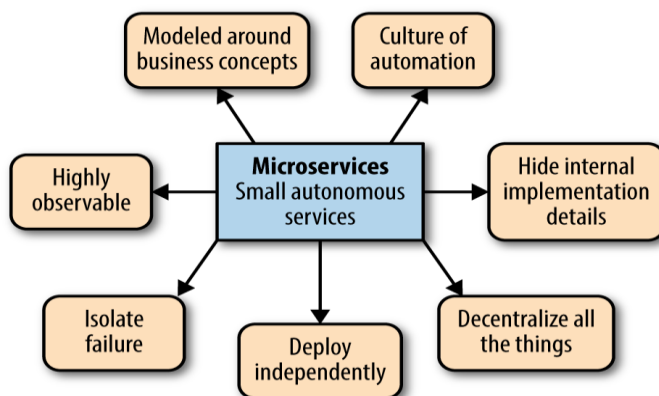


Рисунок 1 - Принципы MSA

Отказ от ответственности: микросервисы – это, сравнительно, очень молодая технология, так что некоторые не указанные на схеме принципы могут быть лучше в тех или иных случаях, также как представленные на рисунке ниже могут представлять собой проблему при развёртывании. Принципы построения должны определяться только архитектором системы.

Моделирование вокруг бизнес-процессов. Отличным способом разбить монолит на микросервисы – это смоделировать их вокруг бизнес-процессов (рисунок 2). В настоящих компаниях разные департаменты имеют слабые связи между собой, так что создание микросервисов, повторяющих их структуру может быть крайне полезным.

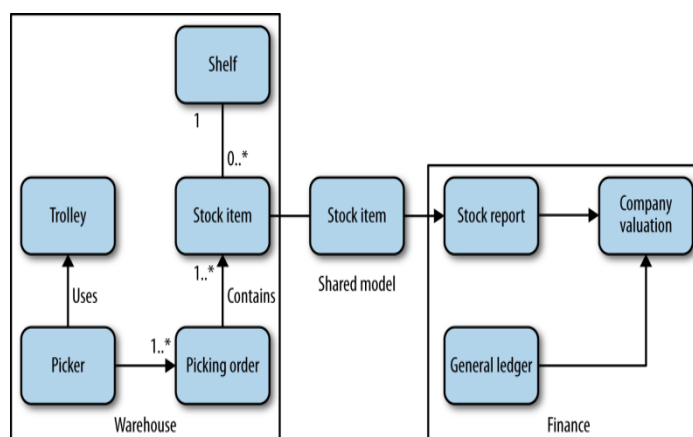


Рисунок 2 - Пример моделирования системы вокруг бизнес-процессов

Тем не менее, в случае создания микросервиса с нуля (вместо разбиения монолита) куда чаще можно столкнуться с проблемой неверного определения границ сердца из-за недостаточного понимания процессов на предприятии. На самом деле это – не большая проблема, потому что размер микросервисов подразумевает, что каждая ошибка может и должна быть исправлена быстро.

Соккрытие деталей внутренней реализации. Как показано на втором рисунке микросервисы, которые представляют собой складской и финансовый отделы “находятся в коробке”. Это означает, что детали реализации каждого сервиса должны быть сокрыты от прочих сервисов. В идеале микросервисы должны представляться другим сервисами только посредством API.

Децентрализация. Тенденции децентрализации наблюдаются во множестве современных программ (например, в Blockchain). Так что микросервисы тоже стремятся уйти от единой точки отказа. Это означает, что какой-либо сбой одного компонента не должен привести к краху целой системы. Это также означает, что при построении микросервисов мы стремимся избежать расположения их на единой платформе (золотое правило: один хост – один сервис) или даже в едином дата-центре в целях предотвращения потери работоспособности системы целиком ввиду отказа каких-либо физических серверов.

Изоляция сбоев. Потеря дата-центра не единственная проблема, с которой можно столкнуться при разработке систем, основанных на микросервисной архитектуре. Цель хорошего архитектора построить стабильную систему микросервисов. Когда одна из частей системы работает плохо это не должно повлиять на другие её части. Это – одно из ключевых преимуществ микросервисной архитектуры, потому что изоляция сбоев практически невозможна в монолитных системах.

Независимое развёртывание. Когда микросервисы разделены, децентрализованы и изолированы, приятным дополнением может стать возможность независимого их развёртывания. Это приносит множество преимуществ, но также налагает обязанности по предоставлению устаревшего API совместно с его новой версией потому что клиенты разрабатываемого той или иной командой микросервиса могут быть не готовы к переходу.

Микросервисы подразумевают, что разработчик может столкнётся с множеством ошибок. Но в отличие от монолитных систем, к ошибкам в MSA система готова. Как описано выше, все сбои должны быть изолированы, но для команд, участвующих в разработке и поддержке MSA важно быстро распознавать и реагировать на их появление. Для достижения данной цели важно иметь правильно сконфигурированные системы мониторинга.

Следует отметить, что хорошие микросервисные системы должны быть максимально автоматизированы. Чем больше сервисов реализуется, тем сложнее отследить их работу. Представленный программный комплекс способен выполнять базовую работу по проверке системы автономно. В идеале разработчик должен создать полностью автоматического наблюдателя, который контролирует выпуск новой версии сервиса в стадию производства и контролирует его в целях уменьшения затрат на его разработку.

Список использованных источников:

1. Микросервисы. [Электронный ресурс]. – Режим доступа: http://ra07.twirpx.net/1645/1645210_78383899/newman_sam_building_microservices.pdf. – Дата доступа: 04.03.2018.
2. Микросервисы (microservices). [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/249183/>. – Дата доступа: 04.03.2018.
3. Микросервисы. [Электронный ресурс].- Код доступа: <https://www.youtube.com/watch?v=hqnylrf81a>. – Дата доступа: 15.03.2018.

ПОСТРОЕНИЕ ЗАКРЫТОГО УЧАСТКА ИНФРАСТРУКТУРЫ С ЦЕЛЮ ОБЕСПЕЧЕНИЯ ПОЛЬЗОВАТЕЛЕЙ БЕЗОПАСНОЙ ТЕСТОВОЙ ВИРТУАЛЬНОЙ СРЕДОЙ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Петрович А.С., Думиков А.А.

*Бакунова О.М. – ст. преподаватель каф. ИСиТ, м.т.н.
Бакунов А.М., ст. преподаватель каф. ИСиТ, м.т.н.
Калитеня И.Л. – ассистент каф. ИСиТ, м.т.н.*

Одной из самых сложных задач инженерии является создание такой системы, которая одновременно надежна и эффективна. В работе рассмотрены особенности решения инженерных задач, реализуемых в закрытых зонах, в так называемых песочницах, на примере средства автоматизации развертывания виртуальных машин.

В древности словом «инженер» называли создателя военных машин; в средние века — создателя мостов, иных сооружений; современное понимание термина «инженер» возникло после Первой мировой войны, когда словом «инженер» стали называть создателей комплексов и систем различного назначения.

Примерно в то же время возникло понятие «инфраструктура». Оно обозначало комплекс сооружений, созданный для поддержки военных сил, однако в скором времени понятие «инфраструктура» получило более широкое значение и стало обозначать комплекс взаимосвязанных структур и объектов обслуживающего назначения, обеспечивающих функционирование системы.

В области информационных технологий инфраструктурой называют комплекс вычислительных, телекоммуникационных и программных средств, которые предоставлены сотрудникам для выполнения различных рабочих задач.

Для администраторов компьютерных систем и сетей одной из самых сложных задач является создание надежной инфраструктуры.

Одним из способов решения данной задачи является разделение заданной сети на несколько подсетей различного функционального назначения, сообщение между которыми осуществляется по заданным правилам. Простейший пример — разделение сети предприятия на пользовательскую и серверную подсети.

К правилам сообщения между подсетями относят правила маршрутизации и фильтрации сетевого трафика. Разумный выбор этих правил может значительно повысить безопасность сети. Например, можно допустить к работе с серверной сетью лишь отдельные сетевые адреса, уменьшая таким образом риск несанкционированного доступа.

Однако данный способ не подходит в случаях, когда произвольному числу пользователей необходим доступ к произвольному числу серверов. Тогда наиболее эффективным способом обезопасить сеть является выделение в серверной подсети так называемой «демилитаризованной зоны».

Демилитаризованная зона — буфер между наиболее важными серверами и пользовательской подсетью. Получение злоумышленником доступа к серверам демилитаризованной зоны не может привести к серьезным последствиям для инфраструктуры в целом, т.к. не дает злоумышленнику никаких прав в серверной подсети. Согласно примерным подсчетам, выделение демилитаризованной зоны в сети позволяет уменьшить стоимость инфраструктуры в 4 раза за счет сокращения расходов на отдельные отказоустойчивые серверы для наиболее важных систем, расходов на проведение балансировки нагрузки, регулярного резервного копирования и т.п.

Однако существует и другой способ. Он предлагает выделение закрытой подсети, которая функционирует независимо от демилитаризованной зоны. Такую подсеть называют «песочницей».

Песочница не проверяется и не контролируется системным администратором. Размещение в песочнице важных данных и систем запрещается административно. Работа с песочницей организуется таким образом, что любые данные и системы, которые в ней размещены, могут быть уничтожены в любой момент без последствий для инфраструктуры в целом и демилитаризованной зоны в частности.

Поэтому тестируемые в песочнице программные средства должны разрабатываться с использованием методов непрерывной поставки и интеграции, которые позволяют хранить код на одном носителе, а выполнять на другом. В этом отношении эталоном является такое программное средство, тестирование которого в песочнице требует меньше времени, чем получение несанкционированного доступа к ней, или такое, что несанкционированный доступ не может оказать на него негативных последствий.

Дополнительной линией защиты может служить физическое выделение песочницы в отдельную сеть. Для этого необходимы отдельный сервер и интернет-канал, посредством которого будет осуществляться доступ к песочнице из основной сети.

Принцип работы песочницы на примере средства автоматизации развертывания виртуальных машин «SelfPortal» изображен на рисунке 1.

Получение доступа к ресурсам песочницы сопряжено с рядом затруднений. Чтобы разъяснить это, условно разделим виртуальные машины по признаку используемой операционной системы:

1. Виртуальные машины с ОС семейства Windows.
2. Виртуальные машины с ОС семейства Unix.
3. Виртуальные машины с другими операционными системами.

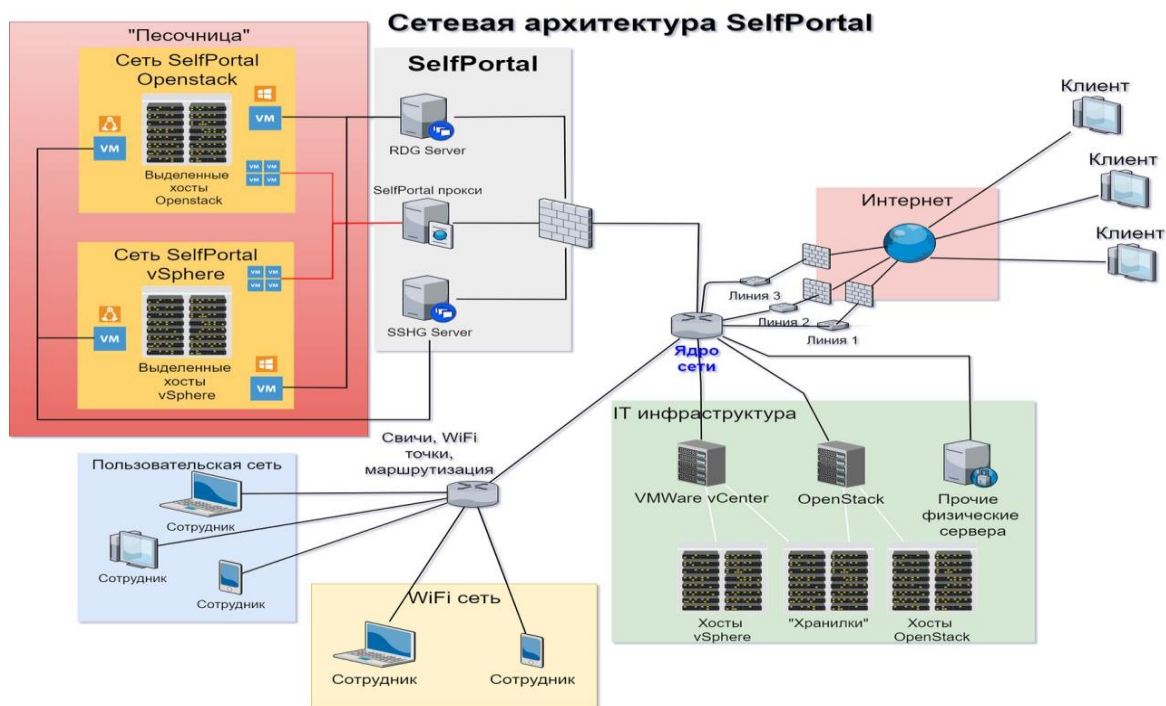


Рисунок 1 – Принцип работы песочницы на примере ПК «SelfPortal»

Это деление удобно в приложении к методам, которые позволяют получить доступ к системе. Например, основным способом доступа к виртуальным машинам категории 1 является Remote Desktop Protocol, тогда как для машин категории 2 — Secure Shell или Virtual Network Computing.

NoVNC Viewer является частью программного пакета, предоставляемого непосредственно системой виртуализации как инструмент контроля над виртуальной машиной. Его невозможно централизовать, так что общее число «входных точек» в песочницу будет равно числу используемых провайдеров виртуализации плюс два.

Поэтому необходимо строго настроить брандмауэр, чтобы запретить любой трафик, не прошедший через заданные системы. Кроме того, следует блокировать любой трафик, созданный данными системами, чтобы предотвратить доступ к основной сети предприятия злоумышленником, который получил возможность пользоваться ими. Такие меры обеспечивают сотрудникам предприятия качественно новый уровень сервиса.

Отметим в заключение, что описанная методика не принесет экономической выгоды предприятиям, которые не предоставляют сотрудникам непосредственный доступ к инфраструктуре, поскольку тогда любой виртуальный сервер будет проверяться системным администратором, так что сложные защитные меры по организации песочницы становятся лишними. Вместе с тем для предприятий, которые непосредственно выделяют сотрудникам вычислительные ресурсы, данная методика может стать инновационным решением.

МОДЕЛИ И АЛГОРИТМЫ ОБЕСПЕЧЕНИЯ НЕОБХОДИМЫМИ РЕСУРСАМИ ДЛЯ ЖИЗНЕДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА НА ОСНОВЕ ТЕОРИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Поплёвка В.И.

Скудняков Ю.А. - доцент каф. ПЭ, к.т.н., доцент

В работе рассматриваются возможности использования некоторых подходов и предложенных в ней моделей и алгоритмов обеспечения необходимыми ресурсами для жизнедеятельности человека на основе теории искусственного интеллекта [1,2].

Для формирования поведения человека наиболее подходящим является игровой искусственный интеллект, так как он позволяет задавать алгоритмы объекта в зависимости от текущего состояния системы. При формировании игрового искусственного интеллекта используют следующие подходы [3]:

1. Машины конечных состояний

Машины конечных состояний являются наиболее распространенным алгоритмом поведенческого моделирования. Они концептуально просты и быстро кодируются, что приводит к созданию мощной и гибкой структуры искусственного интеллекта с небольшими расходами. Машины разбивают искусственный интеллект на

дискретные части, именуемые состояниями. Состояния связаны между собой переходами, ответственными за переход в новое активное состояние всякий раз, когда выполняются определенные условия.

2. Иерархические машины конечных состояний

Машины конечных состояний – полезный инструмент, но у него есть недостатки. Работа с парой состояний не представляет никаких проблем. Однако, когда происходит работа с объемным искусственным интеллектом, содержащем в себе десятки состояний, то добавление новых состояний может быть чрезвычайно сложной. Иерархические машины конечных состояний решают эту проблему, позволяя объединить наборы состояний в отдельные машины и осуществлять переходы между ними, а не состояниями.

3. Дерево поведения

Дерево поведения описывает структуру данных, начинающуюся с некоторого корневого узла и состоящую из поведений, являющихся отдельными действиями, которые объект может выполнять. Каждое поведение определяется предварительным условием, определяющим возможность совершения данного действия. Также поведения, в свою очередь, могут иметь поведения-наследники, что и дает алгоритму свои древовидные качества.

4. Целеориентированный планировщик действий

При формировании искусственного интеллекта, для которого приоритетом является достижение конкретной цели (например, предотвращение голода или жажды), используют целеориентированный планировщик действий. Данный алгоритм позволяет системе искусственного интеллекта создавать свои собственные подходы к решению проблем, предоставляя описание того, как работает мир, то есть формировать список возможных действий объекта, требования перед каждым возможным действием и последствия совершения конкретного действия. Из возможных действий алгоритм находит то, которое приведет к поставленной цели. Далее, анализируя текущее состояние системы и условия, необходимые для выполнения конкретного действия, алгоритм находит действия, необходимые для достижения условий выполнения следующего действия. И так до тех пор, пока алгоритм не достигнет действия, выполнимого с начального состояния системы. Данный алгоритм можно доработать, заменив действия на состояния, что позволит расширить логику поведения, заменив метод выполнения действия методами входа, выхода и обновления состояния.

Разработанная система использует игровой движок Unity в качестве среды демонстрации результатов своего функционирования и целеориентированный планировщик действий в качестве основной части искусственного интеллекта объекта, симулирующего поведение человека или бота. В системе используется компонентно-ориентированный подход, благодаря чему система легко встраивается в приложения, разработанные с помощью игрового движка Unity.

Моделирование искусственного интеллекта выглядит следующим образом: объект инициализирует компонент AIController значениями из компонента AISetup. AIController, передает значения условий из класса AISense и цель объекта в класс IAgent. IAgent передает полученную информацию в класс AIPlanner, который формирует план AIPlan и выбирает следующее действие для объекта. Далее IAgent устанавливает объекту новое действие. Это вызывает смену состояния объекта AIState. Для хранения значений условий используется класс AICondition. Для хранения значений графа поведения объекта используется класс AINode. Для разработки действий, возможных для объекта, используется класс AIScenario. Графическое представление структуры компонентов системы искусственного интеллекта показано на рисунке 1.

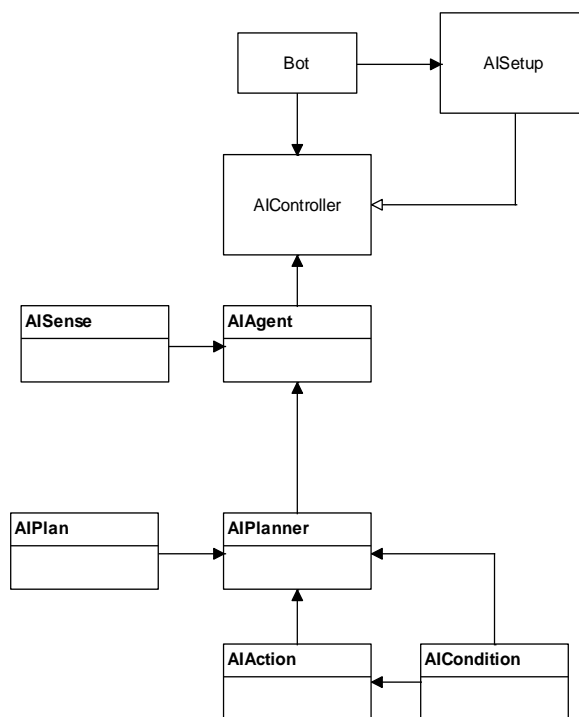


Рисунок 1 – Графическое представление структуры системы искусственного интеллекта

Для работы системы необходимо добавить к объекту компоненты AISetup и AIController. С целью формирования состояний объекта реализуются классы-наследники от класса AIState, а в классе AISense осуществляется реализация логики определения состояний системы. Описание сценария поведения объекта с помощью AIScenario приведено на рисунке 2, которое указывается в компоненте AISetup.

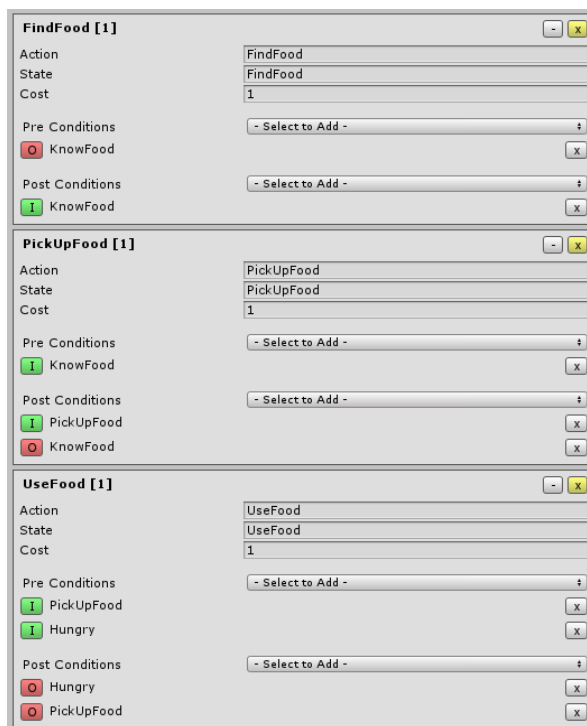


Рисунок 2 – Пример сценария поведения объекта

Список использованных источников:

1. Wikipedia [Электронный ресурс]. – Режим доступа: <http://wikipedia.org>. – Дата доступа: 15.03.2018.
2. Остроух, А.В. Интеллектуальные системы / А.В. Остроух. – Красноярск: Научно инновационный центр, 2015. – 110 с.
3. Rabin, S. Game AI Pro: Collected Wisdom of Game AI Professionals / S. Rabin - A K Peters/CRC Press, 2013. - 626 с.

МИКРОПРОЦЕССОРНОЕ УСТРОЙСТВО МОНИТОРИНГА ПЕРЕМЕЩЕНИЯ АВТОМОБИЛЯ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Прибыткин М.А.

Журавлёв В.И. – доцент каф. ПЭ, к.т.н., доцент

В работе рассмотрены особенности проектирования устройств для мониторинга движения автотранспорта.

На сегодняшний день одной из главных областей применения GPS-технологий является мониторинг автотранспортных средств. Современные спутниковые системы контроля позволяют получать оперативную информацию о точном местоположении объекта, его остановках, скорости, с которой он передвигается, уровне топлива в баке и т.д. [1,2].

Разрабатываемое устройство должно осуществлять следующие функции:

- слежение за транспортным средством;
- индикация местоположения;
- приём передача и данных;
- возможность аварийного вызова вручную;
- резервное питание в случае аварийной ситуации.

Для этого в структуре устройства должны быть предусмотрены: модуль GPS/ГЛОНАСС, модуль GSM, устройство громкой связи, устройство резервного питания, устройство вызова вручную (рисунок 1).

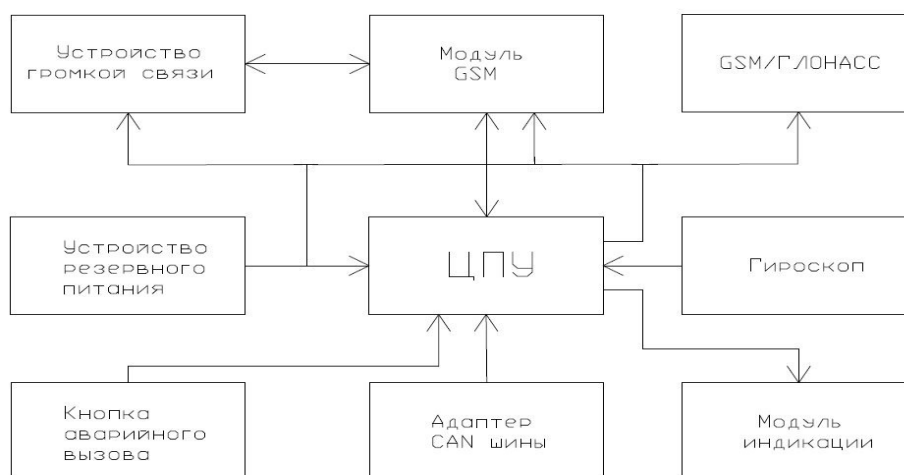


Рисунок 1 – Схема электрическая структурная

Основой в разрабатываемом устройстве является ЦПУ. Микропроцессор отслеживает текущее состояние отдельных модулей системы, их настройки и передачу данных между ними. При дорожно-транспортном происшествии ЦПУ должно автоматически отправить SMS сообщение в службу спасения или звонок может быть инициирован нажатием кнопки SOS. При этом SMS сообщение должно содержать координаты местоположения. При неудачной отправке SMS сообщения, как правило, при неисправной или разряженной аккумуляторной батарее, система должна формировать сообщение с указанием ошибки на дисплей. Устройство должно учитывать малую скорость и небольшое изменение координат. При получении SMS сообщения по каналу GSM на модем или GSM модуль, ЦПУ должно иметь возможность дешифровать текст сообщения.

В задачу устройства громкой связи входит обеспечение возможности общения водителя или пассажиров транспортного средства в салоне автомобиля через встроенные микрофон и динамик. Если в машине уже предусмотрена громкая связь, она будет подключаться к устройству.

Инновационной составляющей разрабатываемого устройства является применение повышающего-понижающего преобразователя, что позволяет увеличить срок службы аккумулятора.

Список использованных источников:

1. ГЛОНАСС мониторинг транспорта [Электронный ресурс]. – Режим доступа: <http://rnsinfo.ru/materials/articles/monitoring/1417/>. – Дата доступа 15.03.2018.
2. Слепушкин, Ю. GPS технологии на транспорте/ Ю.Слепушкин // Беспроводные технологии – №4. – 2007. – С.50-52.

О КОМПЛЕКСНОСТИ НАУЧНЫХ ИССЛЕДОВАНИЙ В ОБЛАСТИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Прузан А.Н.

Таболитч Т.Г. – к.т.н., доцент

Обсуждается проблема комплексности научных исследований в области безопасности облаков учёных СНГ с точки зрения числа рассматриваемых направлений информационной безопасности по сравнению с практическими направлениями аналогичных исследований учёных дальнего зарубежья. Делается вывод.

Информационная безопасность облачных вычислений является, несомненно, одним из приоритетных направлений в современных ИТ-технологиях. Подтверждением этому служат составляемые раз в полугодие компанией Cisco Systems, Inc. отчеты по информационной безопасности, где обязательно присутствует раздел «Интеллектуальные системы кибербезопасности компании Cisco для облачных вычислений» [1]. Этот раздел может служить примером комплексности практических исследований в области информационной безопасности в облаках. Например, в [1, с. 62] одновременно (в комплексе) исследуются:

- 1) предотвращение потери данных;
- 2) защита от DDoS-атак;
- 3) защита электронной почты;
- 4) шифрование/конфиденциальность/защита данных;
- 5) защита оконечных устройств/антивирус/защита от вредоносных программ;
- 6) веб-безопасность;
- 7) защищённая беспроводная связь и ряд других направлений защиты информации в облаке.

Возникает закономерный вопрос: настолько ли комплексными с точки зрения числа рассматриваемых

направлений информационной безопасности являются научные исследования учёных СНГ как практические исследования в той же области компании Cisco Systems, Inc.?

Для ответа на вопрос проанализируем результаты работ по защите информации в облаке известного на Украине коллектива исследователей под руководством академика Украины, д.т.н., профессора В. С. Харченко [3] (Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт»). Среди работ коллектива ввиду ограниченного объёма этой публикации рассмотрим только три статьи – работы [4–7]. В [4–6] рассматриваются только DDoS-атаки на облако. В [7] анализируются только стандарты информационной безопасности для облачных технологий и тенденции их развития.

Коллектив из БГУИР (д.т.н., профессор В. А. Вишняков и его аспирант М. М. Гондаг Саз) в своих работах [8–11], опубликованных в реферируемом журнале, рассматривает только проблемы аутентификации пользователей в облаках. Здесь, правда, неясно, используют ли модели аутентификации в облачных вычислениях для мобильных приложений из статьи [10] алгоритмы аутентификации пользователей в мобильной среде из тезисов [11] (в [11] нет упоминаний об облаках, поэтому может показаться, что данные алгоритмы пригодны везде, а не только в облачных вычислениях).

ВЫВОД: по состоянию на сегодня ни о какой комплексности научных исследований в области безопасности облаков учёных СНГ с точки зрения числа рассматриваемых направлений информационной безопасности по сравнению с практическими направлениями аналогичных исследований учёных дальнего зарубежья не может быть и речи.

Список использованных источников:

1. Отчет. Cisco по информационной безопасности за первое полугодие 2017 г. – Сан-Хосе (Калифорния): Cisco Systems, Inc., июль 2017. – 90 с.
2. Журавлёв, М. С. Краткий обзор украинских работ по защите данных в облаках / М. С. Журавлёв // В наст. сборнике.
3. Профессор Вячеслав Сергеевич Харченко: библиографический указатель к 60-летию со дня рождения / И. В. Олейник, В. С. Гресь, К. М. Нестеренко, В. М. Новичкова. – Харьков, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт». 2012. – 258 с.
4. Меленец, А. В. Защита Cloud-архитектур от DDoS-атак / А. В. Меленец // Радіоелектронні і комп'ютерні системи. – 2013. – № 5 (64). – С. 64–69.
5. Меленец, А. В. Многоверсионная модель защиты облака от DDOS-атаки / А. В. Меленец // Радіоелектронні і комп'ютерні системи. – 2014. – № 6 (70). – С. 140–144.
6. Melenets, A. The State Corporate Cloud Computing- Based Network for Registration of Potentially Dangerous Objects / A. Melenets // Information & Security: An International Journal. – Sofia, Bulgaria, 2012, – Vol. 28, – N 1 & 2. – P. 52–62.6.
7. Поночовный, Ю. Л. Стандарты информационной безопасности для облачных технологий и тенденции их развития / Ю. Л. Поночовный, А. А. Фурманов, В. С. Харченко // Радіоелектронні і комп'ютерні системи. – 2015. – № 4 (74). – С. 25–33.
8. Вишняков, В. А. Модели и средства аутентификации пользователей в корпоративных системах управления и облачных вычислениях / В. А. Вишняков, М. М. Гондаг Саз // Доклады БГУИР. – 2016. – № 3 (97). – С. 111–114.
9. Вишняков, В. А. Концепция и обеспечение безопасности корпоративных информационных систем с использованием облачных вычислений / В. А. Вишняков, М. М. Гондаг Саз, М. Г. Моздураны Шираз // Доклады БГУИР. – 2016. – № 8 (102). – С. 101–102.
10. Вишняков, В. А. Модели аутентификации в облачных вычислениях для мобильных приложений с интеллектуальной поддержкой выбора / В. А. Вишняков, М. М. Гондаг Саз // Доклады БГУИР. – 2017. – № 1 (103). – С. 82–86.
11. Гондаг, С. М. Алгоритмы аутентификации санкционированных пользователей в мобильной среде / С. М. Гондаг, В. А. Вишняков // Технические средства защиты информации: Тезисы докладов XIV Белорусско-российской научно-технической конференции, 25-26 мая 2016 г., Минск. – Минск: БГУИР, 2016. – С. 28–29.

КОМПЛЕКСНАЯ ЗАЩИТА ПЕРСОНАЛЬНЫХ ДАННЫХ НА БАЗЕ ПО NET STUDIO

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Пуйдак В.А.

Пачинин В. И. – зав. кафедрой ИСиТ, канд. техн. наук, доцент

В работе проведен анализ основные методы комплексной защиты персональных данных на основе современных технологий.

Значительную часть работ по защите информации составляют задачи обеспечения безопасности рабочих станций и серверов. Для их решения применяются продукты класса Endpoint Security, которые компенсируют внутренние и внешние угрозы с помощью различных подсистем безопасности (антивирус, СЗИ от НСД, персональный межсетевой экран и др.) [1].

Модель угроз информационной безопасности традиционно включает целый перечень актуальных для рабочих станций и серверов угроз. Угрозы не получалось нейтрализовать одним-двумя средствами защиты информации (далее — СЗИ), поэтому администраторы устанавливали 3-5 различных продуктов, каждый из которых выполнял определенный набор задач: защиту от несанкционированного доступа, вирусов, фильтрацию сетевого трафика, криптографическую защиту информации и т. д.

Такой подход сводит работу администраторов к непрерывной поддержке СЗИ из различных консолей управления и мониторинга. Кроме того, продукты разных вендоров плохо совместимы, что приводит к нарушению функционирования и замедлению защищаемой системы, а в некоторых случаях — и вовсе ко сбою в работе.

Сегодня на рынке появляются комплексные решения, которые объединяют несколько защитных механизмов. Такие СЗИ упрощают администрирование и выбор мер по обеспечению безопасности: у них единая консоль управления, отсутствуют конфликты в работе подсистем безопасности, продукты легко масштабируются и могут применяться в распределенных инфраструктурах. Одним из комплексных средств защиты является продукт Secret Net Studio 8.1, разработанный компанией «Код Безопасности» [2].

СЗИ Secret Net Studio — это комплексное решение для защиты рабочих станций и серверов на уровне данных, приложений, сети, операционной системы и периферийного оборудования. Продукт объединяет в себе функционал нескольких средств защиты «Кода Безопасности» (СЗИ от НСД Secret Net, межсетевой экран TrustAccess, СЗИ Trusted Boot Loader, СКЗИ «Континент-АП»), а также включает ряд новых защитных механизмов.

В рамках данного продукта решаются следующие задачи:

Защита от внешних угроз:

- защита рабочих станций и серверов от вирусов и вредоносных программ;
- защита от сетевых атак;
- от подделки и перехвата сетевого трафика внутри локальной сети;

- защищенный обмен данными с удаленными рабочими станциями.

Защита от внутренних угроз:

- защита информации от несанкционированного доступа;
- контроль утечек и каналов распространения защищаемой информации;
- защита от действий инсайдеров;
- защита от кражи информации при утере носителей.

Защита входа в систему.

Защита входа в систему обеспечивает предотвращение доступа посторонних лиц к компьютеру. К механизму защиты входа относятся следующие средства:

- средства для идентификации и аутентификации пользователей;
- средства блокировки компьютера;
- аппаратные средства защиты от загрузки ОС со съемных носителей (интеграция с ПАК «Соболь»).

Идентификация и аутентификация пользователей.

Идентификация и аутентификация выполняются при каждом входе в систему. В системе Secret Net Studio идентификация пользователей осуществляется по одному из 3-х вариантов: по имени (логин и пароль), по имени или токenu, только по токenu.

В Secret Net Studio 8.1 поддерживается работа со следующими аппаратными средствами:

– средства идентификации и аутентификации на базе идентификаторов eToken, iKey, Rutoken, JaCarta и ESMART;

- устройство Secret Net Card;
- программно-аппаратный комплекс (ПАК) «Соболь».

Блокировка компьютера.

Средства блокировки компьютера предназначены для предотвращения его несанкционированного использования. В этом режиме блокируются устройства ввода (клавиатура и мышь) и экран монитора. Предусмотрены следующие варианты:

- блокировка при неудачных попытках входа в систему;
- временная блокировка компьютера;
- блокировка компьютера при срабатывании защитных подсистем (например, при нарушении функциональной целостности системы Secret Net Studio);
- блокировка компьютера администратором оперативного управления.

Функциональный контроль подсистем.

Функциональный контроль предназначен для обеспечения гарантии того, что к моменту входа пользователя в ОС все ключевые защитные подсистемы загружены и функционируют. В случае успешного завершения функционального контроля этот факт регистрируется в журнале Secret Net Studio. При неуспешном завершении регистрируется событие с указанием причин, вход в систему разрешается только пользователям из локальной группы администраторов компьютера.

Контроль целостности.

Механизм контроля целостности следит за неизменностью контролируемых объектов. Контроль проводится в автоматическом режиме в соответствии с заданным расписанием. Объектами контроля могут быть файлы, каталоги, элементы системного реестра и секторы дисков (последние только при использовании ПАК «Соболь»).

При обнаружении несоответствия возможны различные варианты реакции на ситуации нарушения целостности. Например, регистрация события в журнале Secret Net Studio и блокировка компьютера.

Вся информация об объектах, методах, расписаниях контроля сосредоточена в модели данных. Она хранится в локальной базе данных системы Secret Net Studio и представляет собой иерархический список объектов с описанием связей между ними.

Дискреционное управление доступом к ресурсам файловой системы.

В состав системы Secret Net Studio 8.1 входит механизм дискреционного управления доступом к ресурсам файловой системы. Этот механизм обеспечивает:

- разграничение доступа пользователей к каталогам и файлам на локальных дисках на основе матрицы доступа субъектов (пользователей, групп) к объектам доступа;

- контроль доступа к объектам при локальных или сетевых обращениях, включая обращения от имени системной учетной записи;
 - запрет доступа к объектам в обход установленных прав доступа;
 - независимость действия от встроенного механизма избирательного разграничения доступа ОС Windows.
- То есть установленные права доступа к файловым объектам в системе Secret Net Studio не влияют на аналогичные права доступа в ОС Windows и наоборот [2].

Персональный межсетевой.

Система Secret Net Studio 8.1 обеспечивает контроль сетевого трафика на сетевом, транспортном и прикладном уровнях на основе формируемых правил фильтрации. Подсистема межсетевого экранирования Secret Net Studio реализует следующие основные функции:

- фильтрация на сетевом уровне с независимым принятием решений по каждому пакету;
- фильтрация пакетов служебных протоколов (ICMP, IGMP и т. д.), необходимых для диагностики и управления работой сетевых устройств;
- фильтрация с учетом входного и выходного сетевого интерфейса для проверки подлинности сетевых адресов;
- фильтрация на транспортном уровне запросов на установление виртуальных соединений (TCP-сессий);
- фильтрация на прикладном уровне запросов к прикладным сервисам (фильтрация по символической последовательности в пакетах);
- фильтрация с учетом полей сетевых пакетов;
- фильтрация по дате / времени суток.

Фильтрация сетевого трафика осуществляется на интерфейсах Ethernet (IEEE 802.3) и Wi-Fi (IEEE 802.11b/g/n). Авторизация сетевых соединений в Secret Net Studio осуществляется с помощью механизма, основанного на протоколе Kerberos [3].

С его помощью удостоверяются не только субъекты доступа, но и защищаемые объекты, что препятствует реализации угроз несанкционированной подмены (имитации) защищаемой информационной системы с целью осуществления некоторых видов атак. Механизмы аутентификации защищены от прослушивания, попыток подбора и перехвата паролей. События, связанные с работой межсетевого экрана, регистрируются в журнале Secret Net Studio.

Антивирус.

Защитный модуль «Антивирус» в Secret Net Studio 8.1 осуществляет обнаружение и блокировку вредоносного кода по технологии ESET NOD32 [4]. Таким образом, Secret Net Studio позволяет осуществлять эвристический анализ данных и автоматическую проверку на наличие вредоносных программ, зарегистрированных в базе сигнатур. При проверке компьютера осуществляется сканирование жестких дисков, сетевых папок, внешних запоминающих устройств и др. Это позволяет обнаружить и заблокировать внешние и внутренние сетевые атаки, направленные на защищаемый компьютер. Благодаря использованию в рамках одного продукта СЗИ от НСД и антивируса время на проверку и открытие файлов составляет на 30% меньше, чем при независимой реализации защитных механизмов.

Шифрование сетевого трафика.

В состав клиентского ПО системы Secret Net Studio 8.1 включен VPN-клиент, предназначенный для организации доступа удаленных пользователей к ресурсам, защищаемым средствами АПКШ «Континент». VPN-клиент «Континент-АП» обеспечивает криптографическую защиту трафика, циркулирующего по каналу связи, по алгоритму ГОСТ 28147-89 [4]. При подключении абонентского пункта к серверу доступа выполняется процедура установки соединения, в ходе которой осуществляется взаимная аутентификация абонентского пункта и сервера доступа. Процедура установки соединения завершается генерацией сеансового ключа, который используется для шифрования трафика между удаленным компьютером и сетью предприятия.

Таким образом, проведенный анализ показывает, что ПО системы Secret Net Studio 8.1 обеспечивает комплексное решение для защиты рабочих станций и серверов, которое обеспечивает защиту информации одновременно на уровне данных, приложений, сети, операционной системы и периферийного оборудования и позволяет осуществлять централизованное управление и мониторинг всех защитных механизмов с возможностью сбора, корреляции, фильтрации и приоритизации событий информационной безопасности. комплексное решение для защиты рабочих станций и серверов, которое обеспечивает защиту информации одновременно на уровне данных, приложений, сети, операционной системы и периферийного оборудования и позволяет осуществлять централизованное управление и мониторинг всех защитных механизмов с возможностью сбора, корреляции, фильтрации и приоритизации событий информационной безопасности.

Список использованных источников.

1. IBA Endpoint Security. [Электронный ресурс]. - Режим доступа: <http://ibasecurity.com>. - Дата доступа: 23.02.2018.
2. Методы и средства защиты компьютерной информации. [Электронный ресурс]. – Режим доступа: <http://www.volpi.ru/umkd/zki/index.php?man=1&page=37/>. - Дата доступа: 23.02.2018.
3. Протокол сетевой аутентификации Kerberos 5. [Электронный ресурс]. – Режим доступа: <https://www.ixbt.com/comm/kerberos5.shtml/>. Дата доступа: 23.02.2018.
4. Secret Net Studio 8.1 [Электронный ресурс]. – Режим доступа: <https://www.securitycode.ru/products/secret-net-studio/>. - Дата доступа: 23.02.2018.
5. ESET Technology – The multi-layered approach and its effectiveness. [Электронный ресурс]. – Режим доступа: <https://mirror2.esetnod32.ru/technology/ESET-Technology.pdf>. - Дата доступа: 23.02.2018.
6. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/gost-28147-89>. - Дата доступа: 23.02.2018.

ИГРОВОЕ ANDROID-ПРИЛОЖЕНИЕ «KHASAGI: THE LONE WANDERER»

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Путицкий А.И.

Маковский М.Л. – и.о. декана ФКТ

Рассмотрены особенности разработки игрового Android-приложения, с целью интеграции в интернет-магазин Google Play Market для получения финансовой выгоды.

С приходом информационных технологий, сфера развлечений, связанных с компьютерными программами и мобильными приложениями крепко вошла в жизнь современного человека.

Компьютерные игры в современном обществе являются не только способом отвлечься от повседневной рутины, но и видом искусства, способом самовыражения, а для многих и способом заработка. Киберспортивные дисциплины, видеоблогинг и геймдевелопмент являются в современном мире официальными профессиональными областями, приносящими конкурентоспособные прибыли профессионалам в данных областях [1].

Одним из жанров компьютерных игр является жанр «платформер».

Платформер — жанр компьютерных игр, в которых основной чертой игрового процесса является прыгание по платформам, лазанье по лестницам, собирание предметов, обычно необходимых для завершения уровня. Некоторые предметы, называемые усилителями, наделяют управляемого игроком персонажа особой силой, которая обычно иссякает со временем (к примеру: силовое поле, ускорение, увеличение высоты прыжков). Коллекционные предметы, оружие и усилители собираются обычно простым прикосновением персонажа и для применения не требуют специальных действий со стороны игрока. Реже предметы собираются в «инвентарь» героя и применяются специальной командой (такое поведение более характерно для аркадных головоломок).

Яркие и красочные миры компьютерных игр выступают не только как способ уйти от повседневных проблем. ученые все чаще отмечают положительное влияние умеренного увлечения видеоиграми на психическое здоровье, когнитивные и социальные навыки.

Сюжетные игры, где внимание ребенка сосредоточено на повествовании, повышают способность к концентрации и вырабатывают полезную привычку к чтению. Психологи из Университета Фрайбурга утверждают, что система виртуальных поощрений также усиливает мотивацию, которую игрок сможет использовать и в реальной жизни.

Целью проекта явилась разработка игрового-Android приложения, с целью интеграции в интернет-магазин Google Play Market для получения финансовой выгоды.

Для достижения поставленной цели были проанализированы источники литературы, были рассмотрены существующие аналоги на рынке мобильных игровых приложений, учтены их достоинства и недостатки; изучены теоретические основы разработки мобильных приложений и игр, а также тестирования программного обеспечения.

Для реализации проекта были выбраны языки Javascript и UnityScript. Также в разработке был применен Unity3d Game Engine. Для реализации графических и звуковых материалов были использованы Adobe Photoshop и Adobe Audition.

Результатом является разработанное и готовое к использованию игровое Android-приложение, готовое к последующей реализации на рынке игровых приложений. Приложение имеет иерархическую архитектуру, классы приложения созданы по методу множественного наследования «один родитель – несколько детей». Все графические элементы приложения были созданы специально для приложения. В приложении преобладают сложные универсальные алгоритмы, приспособленные к использованию различными игровыми объектами.

По итогам экономических расчетов, прогнозируемая рентабельность приложения составляет более 600%, что определяет поставленную задачу как выполненную.

Список использованных источников.

1. Новак, Д. Game Development Essentials: An Introduction / Новак, Д – USA: CENGAGE, 2012. – 510 с.
2. Lengyel, E. Mathematics for 3D Game Programming and Computer Graphics / Lengyel E.USA: Course Technology PTR, 2011. – 576 с.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПРОЕКТИРОВАНИЯ САДОВЫХ ДОМОВ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Радионон М.Г.

Сечко Г. В. – доцент каф. ИСиТ, к.т.н., доцент

В работе представлена реализация программного средства для проектирования некоторых видов садовых домиков.

Компании-производители и организации, занимающиеся строительством недвижимости и обустройством интерьеров, в последнее время остро нуждаются в инструментах визуализации своей продукции для конечного пользователя. Стремительное развитие производительных мощностей персональных компьютеров позволяет задействовать трёхмерную графику для решения данной задачи, предоставляя пользователю интерфейс для демонстрации и интерактивного редактирования трёхмерной модели.

В данной работе сделан упор на проектирование и конфигурирование садовых домов. Перечень моделей и возможных модификаций заранее утвержден с производителем, однако разработанная система позволяет дополнять и изменять первоначально установленную информацию.

Основываясь на анализе отечественных и зарубежных аналогов программного средства, в рамках которого были выявлены преимущества и недостатки существующих программных продуктов, был разработан интерфейс пользователя, содержащий в себе все минимально необходимые элементы для взаимодействия с трёхмерной моделью садового дома, при этом не ограничивающий всевозможные модификации, запланированные производителем. В программном средстве предусмотрен расчет стоимости и заказ выбранной конфигурации садового дома, система авторизации и хранения информации о пользователе и его реквизитах. Для обработки оставленных заказов, а также редактирования исходных данных о конфигурациях, разработан модуль администрирования, доступ к которому предоставляется ограниченному числу лиц посредством безопасной авторизации.

В основе работы программного средства положена модель взаимодействия клиент-сервер, которая позволяет разделять функционал и вычислительную нагрузку между клиентскими и серверными частями приложениями. Клиентская часть запускается в веб-браузере конечного пользователя, которая взаимодействует с серверной частью по протоколу HTTP. На серверной части располагается основная бизнес-логика и программный интерфейс приложения. Конфигурация и связь узлов программной системы представлены на рисунке 1.

Модуль конфигурирования, расположенный на клиентской части программного средства, разработан на скриптовом языке JavaScript, с использованием библиотеки Three.js, предоставляющей высокоуровневый интерфейс над технологией WebGL для работы с трёхмерной графикой в веб-браузере. WebGL предназначен для создания приложений для работы с 3D в веб-браузерах, которые будут успешно выполняться на самых разных устройствах: персональных компьютерах, смартфонах, планшетных компьютерах и игровых консолях [1].

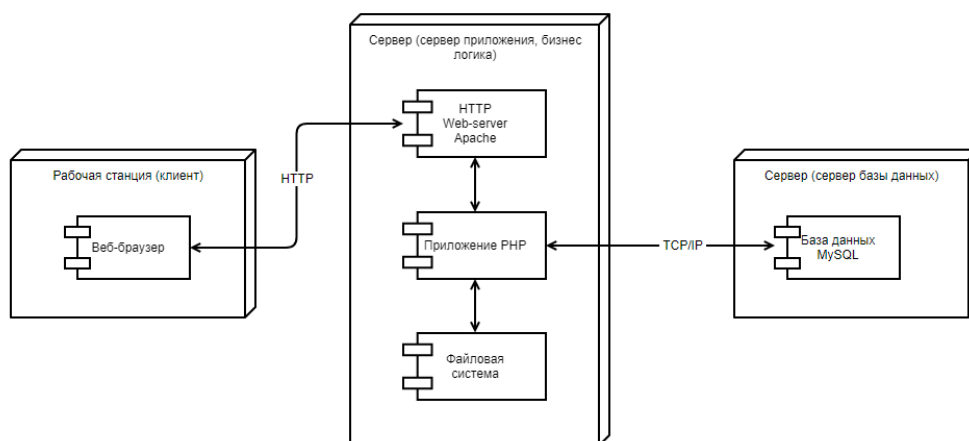


Рисунок 1 - Сетевая архитектура программного средства

Серверная часть приложения использует язык программирования PHP, на котором построена вся бизнес-логика и программный интерфейс программного средства. Функционал оформления и обработке пользовательских заказов реализован с помощью популярной системы управления контентом Magento, преимущество которой заключается в возможности редактирования данных без необходимости правки исходного кода программного средства [2].

Разработанное программное средство выполняет следующие функции:

- конфигурация модели происходит исключительно в трёхмерном режиме;

- режим 3D-демонстрации максимально оптимизирован и не влияет на производительность пользовательского персонального компьютера;
- поддержка большого набора моделей садовых домов различных размеров и форм;
- конфигурация следующих элементов модели: стены, дождевые стоки, крыша, веранда;
- установка окон и дверей;
- поддержка текстур и материалов;
- удобные элементы управления демонстрацией: поворот, приближение, смещение;
- процесс проектирования должен проходить на одной веб-странице для удобства пользования;
- функционал публикации спроектированной модели в социальных сетях;
- расчет предварительной стоимости и оформление заказа.

Список использованных источников:

- 1.Трехмерная графика в вебе [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/325646>. – Дата доступа: 05.12.2017.
- 2.Иствуд, Дж. Magento 1.8 Development Cookbook./Дж, Иствуд, Л. Вьескас.– СПб.: Книга по Требованию, 2004 – 24 с.

СПРАВОЧНО-ИНФОРМАЦИОННАЯ СИСТЕМА ФИНАНСОВОГО ПОРТАЛА MYFIN.BY

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Русак Д.С.

Калитеня И.Л. – ассистент каф ИСиТ, м.т.н.

В настоящее время одним из главных источников получения информации является глобальная сеть Интернет. И эта роль Интернет с каждым годом только увеличивается, заменяя классические телевидение, радио и печатные издания. В работе рассмотрен процесс создания справочно-информационных систем.

Информационные технологии заняли значимое место в промышленности, сельском хозяйстве, управлении персоналом. Информатизация охватила и социальную сферу: образование, науку, культуру, здравоохранение.

Наибольшее распространение информационные технологии получили в сфере финансов, что неудивительно, потому как эта сфера подразумевает много расчетов, которые должны выполняться быстро и требует хранения большого количества информации.

Поскольку в мире финансов все меняется очень быстро, многим людям необходимо держать под рукой актуальные данные о курсах валют, банковских продуктах, торгах на биржах, котировкам и прочей финансовой информации. Именно поэтому много финансовых проектов являются веб-сервисами, доступ к которым можно получить практически в любом месте и с любого устройства (умные часы, смартфон, планшет, персональный компьютер).

Анализ существующих программных средств финансовых справочно-информационных систем показал, что большинство программ имеют серьезные недостатки и функционально ограничены, что не удовлетворяет всем потребностям пользователей, но при этом так же были выявлены и преимущества, которые успешно реализованы в программе. Разработаны алгоритмы загрузки данных в виде XML файлов и вручную, выбора наилучшего продукта подходящего под требования пользователя, а также расчета платежей по каждому банковскому продукту. Быстрый и удобный подбор данных, четкая и конкретная информация, адаптированный дизайн под мобильную версию, все это было реализовано в справочно-информационной системе финансового портала MYFIN.BY.

Так как задача стояла в том, чтобы создать веб-приложение, преимущества, которых в доступности с любого устройства любой операционной системы, то язык программирования также должен быть веб-ориентированным. В качестве быстрого, функционального и одновременно простого серверного языка был выбран PHP. В качестве СУБД был выбран MySQL за его тесную интеграцию с PHP [1].

Программное средство обладает следующими достоинствами:

- возможность расчета ежемесячных выплат по продукту;
- поиск по банковским продуктам;
- предоставление информации о банках и времени работы филиалов;
- загрузка банковских продуктов вручную и автоматически;
- мобильная версия;
- актуальная информация о курсах валют для отделений банков в разных городах;
- предоставление информации о торгах на Белорусской валютно-фондовой бирже.

Список использованных источников:

- 1.PHP: Руководство по PHP - Manual [Электронный ресурс]. – Режим доступа: <http://php.net/manual/ru/>. – Дата доступа: 05.12.2017.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ONLINE БРОНИРОВАНИЯ И ТРАНСФЕРА АВТОМОБИЛЕЙ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Рыбаков Е.А.

Кузнецов В.П. – проф. каф. ИСиТ, д.т.н., профессор
Образцова О. Н. – доцент каф. ИСиТ, к.т.н., доцент

Основной задачей, поставленной перед разработчиком, являлась необходимость реализации программного средства, для которое позволит формировать заявки на бронирование и трансфер автомобиля через online интерфейс, после чего отправлять их менеджеру для последующей обработки. Разработка данного программного средства была вызвана необходимостью автоматизации процесса формирования заявки на бронирование и трансфер, что влечет за собой сокращение времени на её обработку.

В основе работы программного средства будет положена двухзвенная клиент-серверная архитектура. Двухзвенная архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные ресурсы. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части запроса.

На рисунке 1 представлена диаграмма развертывания программного средства, демонстрирующая конфигурацию и связь основных узлов программной системы.

Для обмена данными между клиентом и сервером осуществляется по протоколу HTTP используя как синхронные, так и асинхронные запросы.

В качестве СУБД для данного программного средства была выбрана MySQL. MySQL – свободная реляционная система управления базами данных, которая является решением для малых и средних приложений. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

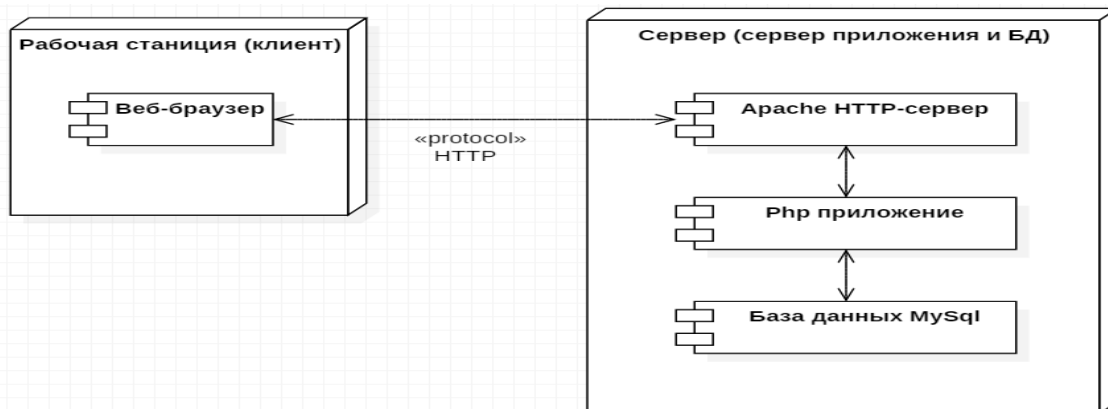


Рисунок 1 - Диаграмма развертывания

Разработка серверной части велась на языке PHP с использованием системы управления содержимым WordPress. WordPress является одной из самых популярных в мире систем управления содержимым, и известен благодаря своей простоте и гибкости. Для данной платформы было разработано огромное количество модулей, решающие большое количество задач, что позволяет значительно сэкономить время на разработку программного средства. В разработанном программном средстве был использован готовый модуль для реализации функции мультиязычности сайта.

Основной особенностью программного средства является подключение интерактивных карт для отображения точек бронирования и трансфера автомобиля в удобной для восприятия форме при формировании заявки пользователем. Для реализации данной функции была произведена интеграция с сервисом «Яндекс.Карты». Общение с сервисом производится через его открытое API используя javascript-библиотеку предоставляемую самим сервисом.

Разработанное программное средство выполняет следующие функции:

- управление пользователями: добавление, редактирование, удаление, блокирование, смена пароля;
- выбор точки взятия автомобиля и точки сдачи автомобиля с использованием интерактивной карты;
- возможность выбора точной даты начала и конца аренды автомобиля при аренде автомобиля;
- добавление, редактирование, удаление категорий автомобилей;

- добавление, редактирование, удаление дополнительных услуг при бронировании и трансфере автомобиля;
 - добавление, редактирование, удаление точек для аренды автомобилями, которые в последствие будут отображаться на интерактивной карте;
 - добавление, редактирование, удаление точек для трансфера автомобилями, которые в последствие будут отображаться на интерактивной карте;
 - возможность перевода на разные языки контента программного средства;
 - возможность смены языка интерфейса ПС пользователем;
 - добавление, редактирование, удаление заявок на бронирование и трансфер автомобиля;
 - создание статичных страниц на сайте и управление их содержимым;
 - фильтр при бронировании и трансфере по различным параметрам групп автомобилей.
- Данное программное средство успешно внедрено в производство (данные отсутствуют ввиду подписанного соглашения о неразглашении конфиденциальной информации).

Список использованных источников:

[1] Дюбуа. MySQL, 3-е издание / Поль Дюбуа. — М.: «Вильямс», 2006. — 1168 с.

[2] Scott. WordPress for Education / Adam D. Scott — Birmingham: Packt Publishing Ltd, 2012. — 144 с.

АНАЛИТИЧЕСКАЯ СИСТЕМА ПРОГНОЗИРОВАНИЯ ОШИБОК

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Сеглюк И.А., Камоцкий Р.Г.

Савенко А.Г. - ассистент каф ПЭ, м.т.н.

В современном мире многие web-приложения очень быстро развиваются: у них появляются новые возможности, увеличивается количество обрабатываемой информации. В результате расширения приложения, также расширяется его инфраструктура. Расширение инфраструктуры приводит к ее усложнению, и, как следствие, возникают сложности в отслеживании работоспособности системы.

Существуют системы мониторинга, позволяющие отслеживать возникающие ошибки/неисправности, в результате работы всей системы. Но, если ошибка все же возникла (особенно на production-сервере), это влечет за собой финансовые потери, которые, в свою очередь, могут быть явными (в случае прямой потери прибыли за время простоя) или косвенными (репутационные потери, которые, как правило, более опасны для бизнеса).

Для того, чтобы максимально обезопасить систему от сбоев, не достаточно иметь квалифицированный персонал, ввиду того, что специалист попросту может не обратить внимания на факторы, указывающие на скорое возникновение ошибки.

Поэтому мы предлагаем вариант системы, осуществляющей сбор и анализ всех необходимых данных, для последующего указания мест, где может возникнуть ошибка. В таком случае будет возможность заранее устранить неисправность, или максимально подготовиться к ней, дабы минимизировать, а то и вовсе избежать, потери.

Работу системы можно разбить на три этапа:

1. Сбор информации
2. Анализ данных и обучение системы анализа
3. Мониторинг и оповещение

Рассмотрим каждый шаг по отдельности:

Сбор информации. Сбор статистической информации происходит посредством установки на каждую виртуальную/физическую машину, программ-агентов, анализирующих систему.

В нашем случае, интересны два типа информации: информация о сервере и информация о приложении.

Так как в компаниях чаще всего сервера базируются на основе ОС одного семейства (чаще всего это операционные системы семейства Linux), информация о системе будет почти равноправна по отношению ко всем серверам системы, в отличие от систем, где могут быть сервера различных семейств (например семейства Linux и семейства Windows).

При сборе информации о системе в основном нас интересуют следующие характеристики:

- загруженность ЦП;
- состояние ОЗУ;
- состояние сети;
- загруженность HDD и количество bad block;
- Количество циклов перезаписи (для SSD);

Более сложной задачей является сбор информации о работе приложений. Точкой вывода данной информации, как правило, являются логи. Чаще всего, формат логирования как правило у каждого приложения свой, в следствии чего их нужно привести к единому виду, для последующего хранения и обработки. Для этого необходимо указать системе, на файлы или каталоги, в которые пишутся логи, а также обозначить данные,

которые являются важными для нас. Ключевым значением в данных о работе приложения является время, поэтому при его сохранении необходимо учитывать и часовые пояса.

Для передачи данных могут использоваться различные форматы представления данных, например, JSON или XML. Данных очень много, поэтому хранить их в ОЗУ не целесообразно. Для таких целей необходимо использовать энергозависимую память с максимально возможной скоростью чтения данных. Скорость записи не так важна, так как предусмотрена возможность скапливания очереди на запись данных. Однако возникновение очень большой очереди, влечет за собой несвоевременную обработку данных, что противоречит первоначальным целям данной системы.

Анализ данных. После сбора данных агентами, информация поступает в единый в конкретной системе центр обработки.

Для обучения системы можно использовать, как ручной, как и автоматический подход.

Для ручного подхода оператору по графикам активности достаточно указать время возникновения сбоев с разделением на точки возникновения, последствия, которые произошли из-за этого сбоя и методы решения данной проблемы.

При использовании автоматического подхода система сама анализирует предыдущие инциденты. Анализируются скачки изменения полученных данных о системе и работе приложения(ий), устанавливается взаимосвязь. На основании полученных в анализе данных, составляется перечень зависимостей (сценарий) приводящих к критическим ситуациям. Таким образом система будет находить чаще всего повторяющиеся проблемы, тем самым обучая себя обнаруживать их в дальнейшем. Так как качество и точность прогноза ошибки, пропорционально зависит от количества и точности внесенных данных, база данных о состоянии системы должна постоянно обновляться и при этом не задерживать у себя большие объемы информации о нормальной работе системы.

Мониторинг и оповещение. Мониторинг системы, как и предыдущие два этапа, осуществляются параллельно. В случае, когда система зафиксирует сценарий возникновения ошибки (полученный на предыдущем этапе), произойдет оповещение отвечающих за это сотрудников, в результате чего, они смогут предотвратить возникновение сбоя.

К достоинствам следует отнести возможность следующее:

- самостоятельность обучение системы;
- использование единой базы, для схожих по архитектуре приложений;
- независимость от человека;
- возможность использования системы не только для веб-приложений, но и для серверных приложений.

К недостаткам данной системы можно отнести, то, что она не способна выявлять уникальные ошибки, которые раньше не происходили. Однако если использовать ее для большего количества схожих проектов, есть вероятность предугадывать неисправности, которые еще не возникали в конкретном проекте, но возникали в других.

В перспективе возможно рассмотрение таких возможностей, как ведение общей базы данных инцидентов и сценариев, которую могут использовать различные организации для предотвращения ошибок в своей системе, а так же оповещение разработчиков программных средств посредством автоматического создания запроса в баг-трекинг-системах (таких как, JIRA, TRACK).

BLOCKCHAIN

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Сергиеня П.В.

Бакунова О.М. – ст. преподаватель каф. ИСиТ, м.т.н.

Бакунов А.М. – ст. преподаватель каф. ИСиТ, м.т.н.

Калетя И.Л. – ассистент каф. ИСиТ, м.т.н.

В докладе рассмотрена технология, лежащая в основе криптовалют.

Технология, которая по-видимому, окажет огромное влияние в скором времени уже существует. И это не социальные сети, не робототехника и не искусственный интеллект. Вы будете удивлены, когда узнаете, что это технология, которая лежит в основе цифровых валют. Это цепочка из блоков, содержащих в себе транзакции – blockchain. Многие считают, что эта технология на уровень выше, чем интернет, в ней огромный потенциал для общества в целом. Последние несколько десятков лет мы знакомы с интернетом. Информация (например, документ), которую вам отправляют по электронной почте, не является оригиналом, вам отправляют копию. Этот метод замечателен, это демократизация информации. Если дело дойдет до ценностей, таких как деньги, например, тогда копия будет плохой идеей. Если вам посылают 200 долларов, то для этого человека очень важно, чтобы вся сумма оказалась у вас, а у него ее не было. Такая проблема называется удвоение расходов у кодировщиков. На сегодняшний день мы не можем представить себе перевод денег без банка, государства и других посредников. Эти люди занимаются построением и обслуживанием всей финансовой деятельности, начиная от удостоверения личности и заканчивая созданием документов. В целом они справляются, но системы посредников не редко взламываются. Электронная почта в мгновение отправляет данные на другой конец

планеты, что касается банковского перевода, то в одном городе он может обрабатываться днями. Посредники берут большие проценты за перевод денег в другую страну. А что если вам приходится переводить деньги в другую страну не раз в месяц, а то и не раз в неделю? Несправедливость в том, что посредники используют возможности информационного века в своих интересах. Богатство растет, а вместе с ним и неравенство.

Что если бы интернет хранил в себе не только информацию, но и ценности? Вроде огромной продуманной базы данных, разбитой на категории и доступная для всех ее пользователей. Любые ценности от денег, до произведений искусства, могли бы управляться, храниться, перемещаться и обмениваться без могущественных посредников.

В начале 2009 года случилось ключевое событие для blockchain'a. Безызвестный Satoshi Nakamoto разработал протокол криптовалюты bitcoin и создал первую версию программного обеспечения, в котором он реализовал его. Эта криптовалюта обеспечила надежность и позволила совершать денежные переводы без посредников. Это стало событием, которое разбудило интерес к технологии, которая вызвала у всех интерес, а у некоторых и ужас. Но bitcoin это актив, он непостоянен, в более широком понятии это криптовалюта, это не деньги, находящиеся под контролем государства. Однако главная ценность здесь в базовой технологии, названная blockchain.

Теперь люди смогут доверять друг другу, так как их доверие будет основано на продуманности системы, на том, что контроль ведется не из одной точки, а множеством компьютеров по всему миру. Теперь доверие выражается в умном коде. Цифровые активы, будь это деньги, музыка или фильмы, не будут находиться в одном месте, а будут тщательно распределены по реестру blockchain'a, со своими критериями. При появлении какой-либо операции, множество компьютеров начинают следить за этой операцией и составлять очередной блок операций. Обладателями этих компьютеров являются «майнеры», добытчики bitcoin'ов. У них много работы: каждые 10 минут компьютеры создают новый блок, куда собирают данные обо всех последних транзакциях. Майнеры работают, решая сложные задачи, у них проводится своего рода соревнование: кто первый обнаружит истину и утвердит блок, тот получает криптовалюту, а потом происходит очень важный момент для безопасности данных, блок соединяется с предыдущим блоком, хранящим в себе операции, и с блоком до него, таким образом создается цепочка из блоков. Блоки связаны со временем, если бы кто-то захотел взломать блок, чтобы заплатить кому-то деньгами из прошлой транзакции, ему надо было бы взломать блок и все блоки, идущие за ним, и не на одном компьютере, а на миллионах ЭВМ в один момент времени. Сложная задача. Эта система в тысячи раз безопаснее, чем системы на наших компьютерах сегодня. Это blockchain.

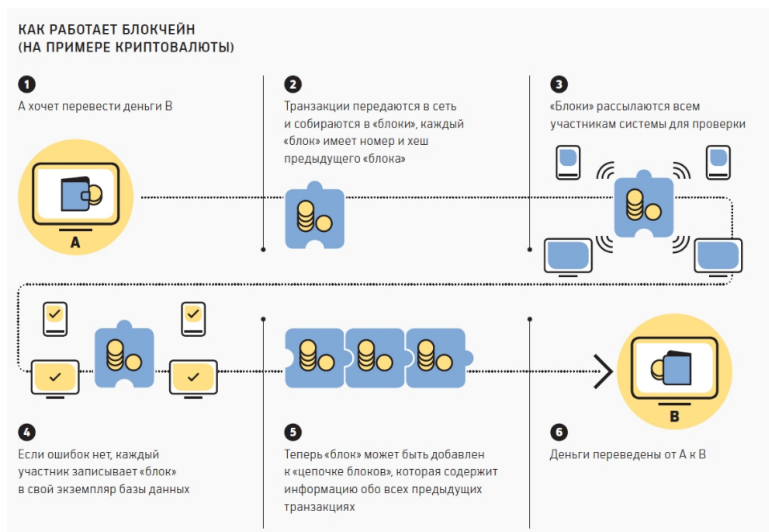


Рисунок 1 – Blockchain на примере криптовалюты

Рассмотрим финансовые услуги. Дедушка решил передать внуку денег. Он пошел в единственный банк в своем поселке, чтобы осуществить перевод. Он отдал деньги и только спустя 3 дня, а то и неделю внук сможет получить эти деньги, так как проводятся различные сверки. После появления blockchain'a больше не будет затрачено времени на проверку платежа, потому что платеж и проверка будет одним и тем же, просто сменой в реестре.

В наше время большинство землевладельцев имеют довольно смутные права на свою землю. Допустим, что вы владелец небольшой фермы, в вашей стране меняется правительство. К вам приходит новоизбранный диктатор и говорит о том, что в базе данных правительства на имя владельца этой земли записан другой человек. Такая проблема действительно была в Гондурасе и подобное может случиться в любой стране, никто от этого не защищен. Hernando de Soto (великий латиноамериканский экономист) считает эту проблему первостепенной, в плане того, что мобильность экономики важнее, чем счет в банке. Государство с помощью заинтересованных компаний работает над тем, чтобы право собственности было связано с blockchain. Тогда никто не сможет изменить владельца в базе данных, а уж тем более взломать эту систему. Это создает условия благосостояния для миллиардов людей.

Самый ценный актив информационного века – это данные. Данные – это актив, являющийся более обширным, чем все вышеперечисленные, такие как земля или промышленное предприятие, или даже деньги. Эти данные появляются благодаря нам, мы создаем данные, мы создаем активы. И за всю нашу жизнь мы воссоздаем свой след из операций. Этот след – виртуальное отражение нас самих. Это виртуальное досье не

находится у нас в распоряжении, оно не принадлежит нам, в этом и заключается проблема. Это досье скажет о нас больше, чем мы сами, так как вы не помните и половины всех событий, что происходили с вами до этого момента. Есть компании, которые занимаются созданием персональных черных ящиков, подобных вашему виртуальному досье. Ваш персональный цифровой профиль хранит и дополняет всю эту информацию и ее можно монетизировать. Суть этого в том, что это также помогает защищать личные данные. А если мы стремимся к свободному обществу, то стоит начать с защиты доступа к личным данным.

Большинство контентмейкеров получают несправедливо мало, потому что система интеллектуальной собственности несовершенна. На примере музыкантов видно, что их записи слушают миллионами, но при этом исполнитель по большей части получает деньги не с продажи своей музыки, а с концертов. Певица Imogen Heap сегодня выкладывает песни в среду blockchain. Музыка находится под защитой, интеллектуальная собственность находится под защитой. Прослушать песню вы сможете либо прослушать бесплатно, либо необходимо заплатить некоторую криптовалюту, которая переведется на цифровой счет. Хотите, чтобы песня была в вашем фильме – придется следовать условиям, прописанным в коде. Поставить песню на рингтон – другие условия. Все эти права специально оговорены. Песня будто становится самостоятельной, все деньги, которые переводятся на счет песни, идут непосредственно к музыканту, таким образом музыканту остается только отправить песню в реестр, а в дальнейшем получать с нее прибыль.

Если вы еще не поняли, что же такое blockchain, то есть один пример. Представьте, что есть какой-то остров, где живет небольшое племя. Их валюта - это большие камни, которые они не могут передвинуть. Назовем такой камень стоун. Они лежат в разных точках острова, но все жители об этом знают. Если кто-то из жителей хочет заплатить другому за какую-либо услугу, то собирается все племя и все узнают о том, что этот bitcoin теперь принадлежит другому жителю. Такая система лучше, так, как если бы учетом этих стоунов занимался один житель, то тут влиял бы еще и такой фактор, как честность. Если этот житель по какой-либо причине исчезнет, то весь учет пропадет. И тут снова будет играть роль фактор честности. Найдутся те жители, которые захотят приобрести себе чей-то стоун обманом. Вернемся к предыдущей системе. В случае, если один из жителей пропадет. Учет останется, так как каждый из жителей знает о том, кому принадлежит определенный стоун. Даже если вдруг один из стоунов скатится в море, жители могут запомнить, что у этого жителя есть этот стоун, но он не находится на суше.

Таким образом можно провести аналогию с blockchain'ом. Жители – это компьютеры майнеров. Когда проводится какая-то операция то множество этих компьютеров это фиксируют, при этом обеспечивая безопасность этой транзакции.

В последнее время выражалось много недовольства по поводу безопасности информации и по поводу защиты интеллектуальной собственности. Blockchain решит эти проблемы и сделает привычное еще проще. Blockchain будет способствовать развитию творчества, оно перестанет быть просто хобби, зарабатывать на своем таланте будет проще. Понимая принцип работы Blockchain'a, можно определить потенциал этой системы и то, на сколько изменится ваша жизнь.

Список используемых источников

1. Fleming, Stephen. Blockchain technology: Introduction to blockchain technology and its impact on Business Ecosystem/Stephen Fleming. – Stephen Fleming, 2017
2. Sean, Bennet. Blockchain: a guide to understanding blockchain/ Bennet Sean. – Cryptomasher via PublishDrive, 2017.
3. Описание технологии blockchain i [Электронный ресурс]. — Код доступа: <https://www.investopedia.com/terms/b/blockchain>. – Дата доступа: 05.03.2018.

СОСТОЯНИЕ ЗАЩИТЫ ПЕРСОНИФИЦИРОВАННЫХ МЕДИЦИНСКИХ ДАННЫХ В БЕЛАРУСИ В 2018 ГОДУ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Ситник М.Ю.

Сечко Г.В. – доцент каф. ИСиТ, к.т.н., доцент

Представлен анализ защиты персонализированных данных в лечебных медицинских учреждениях

Анализируется терминология в области персональных данных пациента и предлагается именовать их персонализированными медицинскими данными. На примере 29-й минской поликлиники анализируется состояние защиты персонализированных медицинских данных в Беларуси в 2018 году. Показывается, что для минимальной защиты конфиденциальных персонализированных медицинских данных, нужно полностью отказаться от всевозможных бумажных носителей медицинских данных и перейти на электронные, в том числе исключить хранение бумажных документов в регистратуре поликлиники. При исключении бумажной документации в медучреждении защиту данных в ЛВС медицинского учреждения традиционными методами (защита от несанкционированного доступа, защита от хакерских атак и т. д.) можно считать достаточной, а, главное, дешёвой. Естественно, однако, что в этом случае стопроцентной гарантии исключения утечки данных для пациента через работников медучреждения нет. Для повышения уровня защиты персонализированных медицинских данных состоятельных пациентов предлагается использовать белорусские биометрические средств контроля доступа к рассматриваемым данным.

Характер медицинской деятельности связан с использованием большого количества данных о здоровье пациента, истории его болезней, истории обращения к врачу и других данных. Эти данные содержатся в

медицинских документах, которые оформляются сотрудниками медицинских учреждений на каждого пациента или их группу. Медицинская документация – это документы установленной формы, предназначенные для регистрации результатов лечебных, диагностических, профилактических, реабилитационных, санитарно-гигиенических и других мероприятий. Примерами медицинских документов для Беларуси являются «Медицинская карта амбулаторного больного», «Медицинская карта стационарного больного» «Медицинская карта амбулаторного больного инфекциями, передаваемыми половым путем», «Медицинская карта больного грибковым заболеванием, чесоткой», «Медицинская карта амбулаторного больного кожным заболеванием», рецепт на лекарство и другие.

Однако, как следует из паспортной части, например, «Медицинской карты амбулаторного больного», в медицинских документах содержатся не только медицинские, но и персональные данные пациента. Согласно Федеральному закону Российской Федерации «О персональных данных» от 27.07.2006 N 152-ФЗ (статья 3, пункт 1) персональные данные – любая информация, относящаяся к прямо или косвенно определенному или определяемому физическому лицу (субъекту персональных данных). Ссылка на закон РФ сделана только потому, что аналогичный закон Беларуси готовится к принятию только в 2019 году.

Комплект персональных данных пациента вместе с его медицинскими данными будем именовать персонифицированными медицинскими данными (ПМД, название «персональные данные пациента» – синоним). Именно эти данные – клад для мошенников [1]. Как утверждают независимые эксперты, на чёрном рынке ценность медицинской информации в 10 раз выше финансовой. Полученные незаконным методом сведения о здоровье могут использоваться в ущерб пациентов. Это и обман пенсионеров, когда им пытаются продать препараты-пустышки за баснословные суммы, и желание вырвать деньги с угрозами обнародования диагнозов у известных личностей, и другие случаи [2]. Отсюда вытекает задача защиты персональных данных пациентов, которым такая защита нужна. Рассмотрим состояние защиты персонифицированных медицинских данных в Беларуси в 2018 году на примере сорока минских поликлиник для взрослых и, включая 13-ю военную и 33-ю студенческую, и 16 детских поликлиник, включая поликлинику Минского района в Боровлянах. Пациенту, обратившемуся в поликлинику, оформляют вручную «Медицинскую карту амбулаторного больного» и, при наличии в поликлинике нужного софта и указания главврача поликлиники, в регистратуре заполняют «Электронную медицинскую карту (ЭМК) [3, 4]» пациента, которая затем автоматически включается в электронную картотеку пациентов (часть базы ПМД поликлиники). Необходимый для заполнения ЭМК софт в Беларуси разрабатывают 5 основных организаций, крупнейшая из которых – ЗАО МАПСОФТ, чей софт работает в 36 из 56 взятых на рассмотрение поликлиник, что составляет 64 % от охваченных автоматизацией поликлиник. Программа для ЭМК в комплексе ПО для медицины ЗАО МАПСОФТ, называется «Регистратура». Если несмотря на наличие программы «Регистратура» в закупленном комплексе ПО для медицины в поликлинике нет или нет указания главврача поликлиники на её применение по причине, например, отсутствия финансирования, то в поликлинике остается только бумажный вариант «Медицинской карты амбулаторного больного». При этом в Беларуси никто не требует у пациента согласия на обработку его персональных данных из ПМД, что предусмотрено статьей 6 (пункт 1 подпункт 1.1) закона РФ «О персональных данных» от 27.07.2006 N 152-ФЗ.

В начале февраля 2014 года появилась информация [3], что в Минске на базе 29-й поликлиники начали тестировать работу ЭМК. Собственно ЭМК, как рассказала главный врач поликлиники Раиса Григорьевна Званец в интервью информационному агентству «Минск новости», представляет собой электронный вариант амбулаторной карты. Причем, по заверению врача, оцифрованы были медицинские карты всех 39 тысяч пациентов поликлиники. Финансировал работы госбюджет, затем эксперимент с белорусской ЭМК в 29-й поликлинике был заморожен до настоящего времени из-за отсутствия дальнейшего финансирования. База оцифрованных ЭМК при этом устарела.

Но, предположим, что база ПМД базе 29-й поликлиники, все 39 тысяч ЭМК, попадут в ЛВС поликлиники. Обеспечит ли поликлиника их надёжную защиту? Мы полагаем, что нет. Во-первых, регистратура поликлиники до сих пор хранит бумажные варианты «Медицинских карт амбулаторного больного». При существующей зарплате медрегистратора в 300 рублей в месяц нет гарантии в том, что за 10 рублей наличными он не представит доступ к любой карте любому злоумышленнику. Во-вторых, при отказе от всевозможных бумажных носителей ПМД и переходе на электронные стопроцентной гарантии исключения утечки данных для пациента через работников медучреждения нет. Действительно, традиционные методы защиты информации в ЛВС медицинского учреждения обеспечивают некоторый уровень защиты, но не исключают утечку данных для пациента через работников медучреждения нет. К числу таких работников относится системный администратор поликлиники, главврач, начмед и некоторые другие. Полагаем, что материально обеспеченные пациенты могут достичь более высокого уровня защиты своих ПМД аутентификацией за умеренную плату своей личности по радужке с помощью описанного в [5, 6] архиватора. Предварительный расчёт показал, что при хранении в базе архиватора в течение трёх лет примерно 2000 изображений радужки стоимость аутентификации, включающая затраты на покупку и эксплуатацию архиватора, для одного пациента составит примерно 100 рублей

Воспользоваться аутентификацией по радужке могут и менее обеспеченные пациенты, если утечка информации может привести к значительным негативным последствиям для таких пациентов.

Следует отметить также, что обеспечение защиты ПМД в Беларуси даже после принятия белорусского Закона о персональных данных встретит ряд негативных моментов, имеющих в России, которая опережает Беларусь в данной области (даже в части упомянутого Закона). Например, в [7] указано, что:

1. Даже если у врача на столе стоит компьютер с ЭМК, ему по привычке проще пользоваться бумажными документами.

2. Обычно в медицинских организациях нет выделенного человека и/или подразделения по информационной безопасности, а в небольших медицинских учреждениях нет даже своих системных администраторов. При этом основные пользователи информационных систем (врачи и младший медицинский персонал) часто имеют низкий уровень осведомленности и компьютерной грамотности.

3. Недостаток финансирования. Когда у главврача стоит дилемма, что купить: томограф или средства защиты информации, выбор всегда будет в пользу томографа, потому что именно это прибор позволяет

обеспечить основную функцию лечебно-профилактического учреждения – защиту здоровья граждан. Здесь надо искать компромисс. Надо, чтобы менеджеры от медицины осознали важность проблемы защиты персональных данных своих пациентов».

ВЫВОДЫ. 1. Чтобы минимально защитить конфиденциальные ПМД, нужно отказаться от всевозможных бумажных носителей медицинских данных и перейти на электронные. К сожалению, в части внедрения электронных медицинских карт Беларусь намного отстаёт от России. Отставание имеется и в части законодательства.

2. При исключении бумажной документации в медучреждении защиту ПМД в ЛВС медицинского учреждения традиционными методами (защита от несанкционированного доступа, защита от хакерских атак и т. д.) можно считать достаточной, а, главное, дешёвой. Естественно, однако, что в этом случае стопроцентной гарантии исключения утечки данных для пациента через работников медучреждения нет.

3. Наиболее материально обеспеченные пациенты могут достичь более высокого уровня защиты своих ПМД аутентификацией за умеренную плату своей личности по радужке с помощью описанного в [5] архиватора. Воспользоваться аутентификацией по радужке могут и менее обеспеченные пациенты, если нарушение заданной характеристики безопасности ПМД, обрабатываемых традиционными методами, может привести к значительным негативным последствиям для таких пациентов.

Список использованных источников.

1. Защита персональных данных в медицинских учреждениях [Электронный ресурс]. – Режим доступа: www.bit-medic.ru/articles/zashita-personalnyh-dannyh/. – Дата доступа 25.03.2018.

2. О чём молчат сотрудники похоронных бюро? Беларускія навіны – [Электронный ресурс]. – Режим доступа: www.newsby.org/by/2011/08/11/text20890.htm. – Дата доступа: 25.03.2018.

3. Электронная медицинская карта. Введение в картотеку «e-Gov.by ... [Электронный ресурс]. – Режим доступа: e-gov.by/stroitelstvo-e-gov/elektronnaya-medicinskaya-karta-vvedenie-v-kartoteku. – Дата доступа 25.03.2018.

4. Электронная медицинская карта (ЭМК) [Электронный ресурс]. – Режим доступа: swan-it.ru/elektronnoe_zdravoohranenie/elektronnaya_meditinskaya_karta. – Дата доступа 25.03.2018..

5. . Гивойно, А. А. Защита медицинских данных пациентов / А. А. Гивойно, В. Н. Ростовцев // Доклады БГУИР. – 2016. – № 7 (101). – С. 79–83.

6. Безопасное архивирование данных с помощью биометрических технологий / А. А. Гивойно, С. В. Нестерович, Г. В. Сечко, Т. Г., Таболич Т. Г. // Веснік сувязі. – 2013. – № 6 (122). – С. 25–28.

7. Обзор: ИТ в здравоохранении 2014, Денег на ИБ не хватает - CNews [Электронный ресурс]. – Режим доступа: www.cnews.ru/reviews/publichealth2014/articles/deneg_na_ib_ne_hvataet. – Дата доступа 25.03.2018.

МНОГОПОТОЧНОСТЬ В UNITY СРЕДСТВАМИ РЕАКТИВНЫХ РАСШИРЕНИЙ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Соколов В.А. Скуратович Е.С.

Бакунова О.М., ст. преподаватель каф. ИСиТ, м.т.н.

В данной статье будут затронуты основные проблемы, возникающие при разработке многопоточных мобильных приложений на платформе Unity, а также способы их решения с использованием библиотеки и реактивных расширений UniRx.

С помощью языка C# можно создавать приложения, выполняющие несколько задач одновременно. Задачи, которые потенциально могут задержать выполнение других задач, выполняются в отдельных потоках; такой способ организации работы приложения называется многопоточностью или свободным созданием потоков. Приложения, которые используют многопоточность, более продуктивно реагируют на действия пользователя, так как пользовательский интерфейс остается активным, в то время как задачи, требующие интенсивной работы процессора, выполняются в других потоках. Многопоточные приложения на языке C# при использовании Mono разрабатываются с помощью ключевых слов: ThreadPool, Thread и асинхронных делегатов.

В качестве примера многопоточного приложения можно представить работу в ресторане. Предположим, что каждый сотрудник выполняет свои обязанности в одно время с другими сотрудниками. К примеру, один готовит еду, второй убирает столы и т.д. (и все это происходит одновременно). Это и есть наши потоки.

ThreadPool — реализация паттерна «пул объектов». Его смысл в эффективном управлении потоками: создании, удалении, назначении им какой-то работы. Возвращаясь к ресторанной аналогии, ThreadPool — это шеф-повар, который контролирует количество сотрудников на стройке и назначает каждому из них задания на день.

Класс, который позволяет создавать новые потоки внутри существующего приложения, называется Thread.

Асинхронные делегаты — асинхронный вызов метода с помощью делегата, который определен с такой же сигнатурой, что и вызываемый метод. Для асинхронного вызова метода необходимо использовать метод BeginInvoke. При таком подходе делегат берет из пула поток и в нем выполняет некоторый код.

Инструменты для синхронизации потоков

Язык C# предоставляет инструменты для синхронизации потоков. Эти инструменты представлены в виде Monitor и lock. Они используются для того, чтобы работа блока кода не осуществлялась одновременно несколькими потоками. Но есть одно различие. Использование этих инструментов может привести к deadlock'у (взаимоблокировке потоков). Это происходит следующим образом: поток A ждёт, когда поток B вернет управление, а поток B, ждёт, когда поток A выполнит заблокированный код. Поэтому синхронизацию потоков и многопоточность необходимо использовать осторожно.

Проблемы встроенных механизмов многопоточности в Unity.

Основной проблемой, с которой сталкиваются программисты при разработке однопоточных приложений — это UI-зависания, вызванные выполнением сложных операций в основном потоке. В Unity имеется механизм распараллеливания задач, представленный, как coroutine (корутин), но он работает в одном потоке, и, если запустить в корутине что-либо «тяжеловесное», то произойдет зависание. Если от программиста требуется параллельное выполнение функций в основном потоке, то можно использовать корутины. Однако, хотелось бы напомнить, что корутины — это итераторы, которые в Unity работают следующим образом:

- сначала идет регистрация корутина,
- далее, после каждого вызова Update и перед вызовом LateUpdate, Unity опрашивает все зарегистрированные корутины и обрабатывает код, который описан внутри метода, имеющий возвращаемый тип IEnumerator.

Помимо своих преимуществ, корутины также имеют и недостатки:

1. Нельзя получить возвращаемое значение
2. Обработка ошибок
3. Проблемы с callback'ами
4. Нельзя обрабатывать тяжеловесные методы в корутинах

Как мы уже отметили ранее, что корутины работают в основном потоке. По этой причине мы получаем зависание, запуская в них громоздкие методы. Все эти недостатки запросто устраняются с помощью реактивных расширений, которые в дальнейшем принесут много различных возможностей и упростят разработку.

Реактивные расширения — набор библиотек, которые позволяют работать с событиями и асинхронными вызовами наподобие Linq запросов. Цель подобных расширений — сделать процесс написания кода быстрее, в котором участвует асинхронное взаимодействие. В Unity применяется библиотека UniRx, которая реализует базовый функционал реактивных расширений для Unity на базе .NET Reactive Extensions. Почему же нельзя использовать эту родную реализацию? Дело в том, что стандартные RX в Unity не работают. Библиотека UniRx поддерживается на платформах PC, Mac, Android, iOS, WP8, WindowsStore и является кроссплатформенной.

Основой реактивных расширений являются интерфейсы IObservable и IObserver. Они предоставляют типизированный механизм для push-уведомления, также известный как шаблон проектирования «Наблюдатель».

- Интерфейс IObservable представляет класс, который отправляет уведомления (поставщик). Интерфейс IObserver представляет класс, который их получает (наблюдатель). Так же существует класс «Т», предоставляющий информацию для уведомлений. Реализация IObserver подготавливает к получению уведомлений от поставщика (реализация IObservable), передавая свой экземпляр методу поставщика IObservable.Subscribe. Этот метод возвращает объект IDisposable, который может использоваться для отказа от подписки наблюдателя до того, как поставщик завершит отправку уведомлений. Интерфейс IObserver определяет три следующих метода, которые должен реализовать наблюдатель:

- Метод OnNext, который вызывается поставщиком для предоставления наблюдателю новых данных или сведений о состоянии
- Метод OnCompleted, который вызывается поставщиком для подтверждения завершения отправки уведомлений наблюдателю.
- Метод OnError, который вызывается поставщиком для указания того, что данные являются недоступными, поврежденными, или у поставщика возникли другие ошибки.

Также в UniRx реализован Scheduler — главный компонент, посредством которого реализована многопоточность. Базовые временные операции (Timer, Interval) в UniRx реализованы с помощью MainThread. Это означает, что большинство операций, кроме Observable.Start, работают в основном потоке и потокобезопасно. Observable.Start по умолчанию использует ThreadPoolScheduler — это означает, что будет создан поток. С основными понятиями и теоретическими знаниями мы ознакомились, теперь рассмотрим пример создания наблюдателя

В данном примере мы попытаемся получить данные из какого-либо интернет-ресурса с помощью библиотеки UniRx. Для загрузки данных с помощью реактивных расширений нам необходимо создать наблюдателя и использовать класс ObservableWWW, который является оберткой над стандартным классом WWW Unity. Метод Get использует корутины и возвращает IObservable, к которому мы присоединим наблюдателя. Данный подход позволяет избежать проблем и ошибок.

Выводы.

Использование реактивных расширений при разработке приложений на Unity имеет множество преимуществ. Основным из них является упрощение синтаксиса для построения многопоточных приложений. Количество костылей с корутинами значительно уменьшается, приложение становится более гибким и быстрым. Также при построении многопоточного приложения с помощью UniRx необходимо помнить, что любая часть данных должна быть защищена от изменения их значений множеством потоков.

Список использованных источников.

1. Документация Unity [Электронный ресурс] – Режим доступа: <https://docs.unity3d.com/>. – Дата доступа: 16.03.2018.
2. Документация по C# [Электронный ресурс] – Режим доступа: <https://msdn.microsoft.com/ru-ru/>. – Дата доступа: 16.03.2018.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ КАДРОВОГО УЧЕТА В ПРОЕКТЕ «ЭЛЕКТРОННАЯ ШКОЛА»

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Соломенников А.С.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Коренская И. Н. – ст. преподаватель каф. ИСиТ*

Объектом исследования является возможность автоматизации работы отдела кадров в учреждениях образования находящихся на самостоятельном финансировании участвующих в проекте «Электронная школа».

Цель работы – разработка программного средства для кадрового учета в проекте «Электронная школа». Разработка данного программного средства позволит автоматизировать работу кадровой службы в учреждения образования находящихся на самостоятельном финансировании.

Исходя из поставленной цели, вытекают следующие задачи:

- изучить специфику работы кадровой службы;
- разработка и настройка пользовательского интерфейса;
- создание базы данных для хранения информации;
- разработать инструмент для генерации отчетной документации.

С сентября 2013 года, комитетом по образованию Мингорисполкома в городе Минске реализуется региональный пилотный проект «Апробация модели управляемого развития образовательных услуг для формирования и стимулирования самообразовательной деятельности обучающихся средствами информационно-коммуникационных технологий» (Электронная школа) [1].

Данный проект согласуется с целями и задачами:

- «Концепции информатизации системы образования Республики Беларусь на период до 2020 года»;
- «Государственной программы развития цифровой экономики и информационного общества на 2016-2020»;
- «Государственной программы «Образование и молодежная политика» на 2016-2020 годы»;
- «Программы перспективного развития отрасли «Столичное образование 2013-2018».

Однако степень развития «Электронной школы» на сегодня не достигла того уровня, при котором бы обеспечивалась полная комплексность информатизации школы с одновременным её удешевлением. В этих условиях сделана попытка разработки одного из элементов «Электронной школы» (программного средства для кадрового учета), который бы по своим функциональным характеристикам был бы не хуже имеющихся аналогов, а по стоимости был бы дешевле аналогов. Принципы, положенные в основу проектирования, – это максимальное удешевление стоимости разрабатываемого программного средства при одновременном превышении его функциональных характеристик над аналогичными характеристиками аналогов.

Список использованных источников:

Список использованных источников.

1. Центр информационных ресурсов системы регионального образования [Электронный ресурс]. – Электронная школа – Режим доступа: <http://iso.minsk.edu.by/main.aspx?guid=4801>. – Дата доступа: 12.10.2017.

ПРОГРАММНОЕ СРЕДСТВО «ИНТЕРНЕТ-КИНОТЕАТР» НА ОСНОВЕ ПИРИНГОВЫХ СЕТЕЙ

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Сорока А.А.

Леванцевич В.А. ст. преподаватель каф. ПОИТ

На сегодняшний день объемы передаваемой информации в сети интернет постоянно растут. С ростом объемов растет уровень интернет-пиратства. Все чаще правообладатели сталкиваются с тем, что их продукты используются без их ведома. Затраченные усилия и денежные вложения не окупаются, что ведет либо к ухудшению качества, либо уменьшению количества информации.

Рассматривая данную проблему, на примере интернет-видео, можно убедиться в том, что актуальность проблемы растет с каждым днем. Количество ресурсов, предоставляющих бесплатный доступ к видео, постоянно увеличивается, а существующие механизмы защиты слишком сложные и затратные. Данная ситуация принуждает пользователей использовать сторонние пиратские веб-сайты, программы или другие средства.

Помимо всего прочего существующие методы защиты накладывают определенные ограничения и на самих правообладателей. Средства, которые используются для обеспечения доступа и защиты ресурсов могли бы быть использованы для улучшения качества информации.

В связи с этим цель данного программного средства решение двух существующих проблем:

- организация передачи данных по технологии, уменьшающей затраты на содержание веб-серверов;
- защита лицензионного видео.

Для достижения поставленных целей необходимо решить следующие задачи:

- определить требования к разрабатываемому программному средству и составление спецификации, включающей их;
- осуществить выбор технологии и языка программирования для реализации программного средства;
- провести проектирование архитектуры программного средства;
- разработка пользовательских интерфейсов;
- разработка моделей данных;
- разработка алгоритмов работы программного средства;
- программирование и тестирование отдельных программных модулей;
- тестирование готового программного средства.
- проанализировать предметную область;

Как показал анализ программной области, основными недостатками существующих программных решений являются следующие:

- отсутствие многоплатформенных разработок;
- отсутствие поддержки работы на мобильных устройствах;
- отсутствие возможности интеграции со сторонними системами;
- высокая стоимость.

Для решения проблемы организации передачи данных было принято решение использовать технологию пиринговых сетей [1]. Преимущества данной технологии:

- нагрузка на сервер со временем распределяется по всей сети;
- выход из строя сервера не остановит работу сети;
- если сеть имеет некоторую вложенность с различной скоростью доступа в каждой, требуемый файл может быть получен с максимально допустимой скоростью передачи.

Для решения проблемы кроссплатформенности были выбраны инструменты, позволяющие производить разработку программного средства при использовании единой среды программирования. Данный подход обеспечивает высокую скорость разработки и в дальнейшем позволит достаточно эффективно сопровождать программное средство при минимальных затратах.

Для защиты видео была использована спецификация EME [2]. Цель данной спецификации – защитить пользователя: от сетевых атак и отслеживания, а также от защиты информации, хранящейся на пользовательском устройстве.

В результате исследования и анализа имеющихся решений, проблем и целей был установлен список требований, которым должно удовлетворять программное средство, а также базовый набор компонентов, которые помогут пользователям эффективно и полезно использовать данное программное средство. Программное средство должно выполнять следующие функции:

- управление лицензиями
- управление видеофильмами
- управление пользователями;
- управление состоянием сети;
- функция подбора фильмов на основе истории просмотра пользователя;
- функция просмотра зашифрованного видео.

В результате выполнения дипломного проекта было разработано программное средство, которое поможет повысить эффективность использования видеоматериалов в сети интернет. Это обусловлено удобной и интуитивно понятной архитектурой и структурой программы, элементами навигации, управления и другими функциональными блоками, отвечающими за основные и наиболее важные функции.

Разработанное программное средство выполняет все необходимые функции и полностью соответствует поставленным требованиям, изложенным в постановке задачи. Оно является полноценным программным продуктом, который объединяет в себе возможности пиринговых сетей, а также возможности современным систем защиты информации.

Список использованных источников.

1. Финкова, М.В. Пиринговые сети / М.В. Финкова. – СПб.: Наука и Техника, 2006. – 272 с.
2. Encrypted Media Extensions [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/encrypted-media/> – Дата доступа 28.02.2018.

СИСТЕМА УПРАВЛЕНИЯ КЛИМАТИЧЕСКИМИ ПАРАМЕТРАМИ СУШИЛЬНОЙ КАМЕРЫ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Филалко Е.А.

Журавлёв В.И. – доцент каф.ПЭ, к.т.н., доцент

В работе рассмотрены подходы к реализации систем управления климатическими параметрами сушильной камеры.

В настоящее время в промышленности и быту широко применяются сушильные камеры и шкафы. Поэтому возникает потребность в точном поддержании температуры и влажности внутри камеры. В сушильных камерах, имеющих большой объём, организовать точное регулирование достаточно трудно. Для реализации данной задачи необходимо измерять температуру и производить нагрев в разных точках камеры с принудительной циркуляцией воздуха под управлением микроконтроллера [1]. При данных условиях можно более точно поддерживать постоянную температуру в пределах 1°C.

В разработанной системе учтены и решены недостатки, которые выявили при анализе существующих прототипов и схемных решений, в результате чего разработку можно использовать как автоматизированную систему сушки, способной к круглосуточной работе. Центральной частью системы (рисунок 1) является управляющее устройство, выполненное на микроконтроллере ATmega32A [2], который координирует работу системы, управляет работой воздушных заслонок, вентиляторов, нагревателей, подачей воды на увлажнение, устройством индикации.

При помощи кнопок управления пользователь может выбрать режим управления и запустить процесс сушки. Микроконтроллер обрабатывает поступающие значения с датчиков, сравнивает их с введёнными значениями и при необходимости выдаёт на исполнительные механизмы сигналы для их включения или отключения. После перевода системы в режим «работа» на дисплее отображаются параметры с датчиков, информация о нагревателях, вентиляторах, состоянии задвижек. После запуска процесса сушки задвижки закрываются, включается нагревательные элементы и вентилятор. При необходимости увлажнения на электромагнитный клапан подаётся напряжение и через форсунки внутри камеры разбрызгивается вода. Присутствует тепловая защита нагревательных элементов, которая реализована при помощи биметаллического реле.

Основные преимущества системы:

- пользователь может использовать стандартную программу сушки или создать свою;
- систему эффективно использовать при модернизации оборудования;
- автоматическая блокировка при аварийных режимах работы.

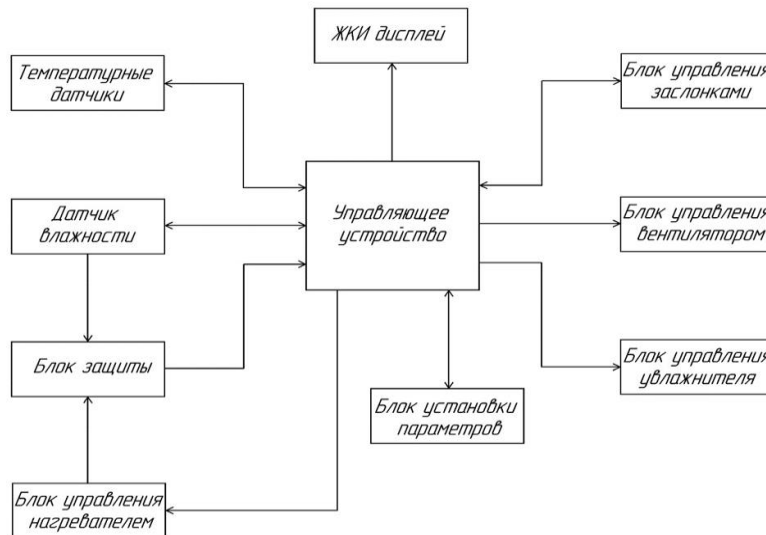


Рисунок 1 – Структурная схема системы управления

Список использованных источников.

1. Трамперт, В. Измерение, управление и регулирование с помощью AVR-микроконтроллеров. / В. Трамперт. – М.: МК-Пресс, 2006. – 208 с.
2. ATmega32, ATmega32A – 8-разрядные микроконтроллеры с 32 Кб внутрисистемно программируемой Flash памяти [Электронный ресурс]. – Режим доступа: <http://www.gaw.ru/html/cgi/txt/ic/Atmel/micros/avr/atmega32.htm> .–Дата доступа 15.03.2018.

КВАНТОВЫЙ КОМПЬЮТЕР – НОВОЕ ПОКОЛЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Халецкий Д.М.

Калитеня И.Л., ассистент каф. ИСuТ, м.т.н.

Вычислительная техника развивается и изменяется уже не одно десятилетие. В настоящее время вычислительная техника укрепила себя почти во всех сферах человеческой деятельности. Развитие и внедрение вычислительной техники

обусловлено укреплением зависимости человека от вычислительных мощностей. Одним из устройств вычислительной техники является компьютер. В работе будет рассмотрена новая ветка развития вычислительной техники – квантовый компьютер.

Вычислительные способности современных компьютеров, которые используют в своей архитектуре бинарные вычисления, растут, однако они почти достигли своих физических пределов. Классический компьютер использует в своей основе бинарный код, единицей измерения в котором является бит, который находится в одном из базовых состояний: 0 или 1. Именно из-за состояния бита его существование или несуществование можно вычислить и контролировать.

Квантовый вычисления используют в своей основе квантовые биты – кубиты. Кубиты, как и биты, имеют два основных состояния. В роли физического состояния кубитов могут выступать атомы, фотоны, ионы и ядра атомов. Главное отличие квантового компьютера, использующего кубиты, от классического использующего биты, для передачи и обработки объема данных задействованы явления квантовой суперпозиции и квантовой запутанности. То есть, вместо привычного состояния битов, находящихся одном из двух базовых состояний, кубит может находиться во всех возможных состояниях одновременно. Изменение состояния кубита ведет к изменению состояния других кубитов. Это существенно ускорит вычисления в некоторых сферах задач. Уже проведен ряд экспериментов физической реализации вычислений на малом числе кубитов.

В теории квантовый компьютер способен на решение задач, которые классический компьютер решал сотни или тысячи лет. Почти все алгоритмы современного шифрования используют в своей основе разложение больших чисел на множители – алгоритм RSA. Алгоритм перебирает все возможные способы разложения. Квантовый компьютер использующий алгоритм Питера Шора будет способен производить вычисления гораздо больших чисел на порядок быстрее классических компьютеров. В итоге все системы, основанные на алгоритме RSA, становятся полностью уязвимы для квантовых компьютеров. Квантовая и постквантовая криптографии, основанные на разных методах, призваны осуществить защиту информации.

Как и любая вычислительная система, квантовый компьютер имеет цель создания и сложности в проектировании. Сложность в проектировании квантового компьютера состоит скорее в инженерной и научной нехватке знаний. Основной проблемой является реализация квантовой телепортации – осуществление переноса состояния кубита на другой. Стандартной схеме используется три кубита: запутанная пара и телепортируемый кубит. При телепортации кубит принимает одно из значений, а оригинал разрушается – пример действия невозможности клонирования. С инженерной точки зрения, существует две проблемы: ошибки вычислений и физическая реализация. Ошибки в квантовых компьютерах делят на две главные группы. Первая группа принадлежит всем компьютерам – внешнее воздействие. Из-за шумов, воздействия магнитных волн и радиации может происходить произвольная смена кубитов. Решение этой проблемы была найдено группой ученых Google во главе с Джулианом Келли. Была создана особая схема девяти кубитов, которые осуществляют поиск ошибок в системе. Из первой группы следует вторая группа ошибок. Кубиты нестабильны по своей природе, информация мгновенно утрачивается. По подсчетам примерно 99% вычислительной мощности любых наработок будет тратиться на устранение ошибок. Внешнее воздействие нарушается связь между кубитами (процесс декогеренции), единственным решением является полная изоляция от внешних факторов, которой достигнуть пока физически невозможно.

Реальный квантовый компьютер ставит перед собой цель достичь: масштабируемости и возможности увеличения количества кубитов, кубиты должны иметь возможность реализоваться в начальное состояние, реализация вентилей Тьюринга, получение информации с кубитов. Реализация квантового компьютера вопрос времени. Однако архитектура, требования и языки программирования разрабатываются уже в настоящее время. Уже сейчас есть возможность проектирования программных симуляторов. Лучшим претендентом является язык разработанный на основе Haskell центром национальной разведки США – «Quipreg». К физической реализации квантового компьютера существует ряд требований:

1. Фиксированное выделение в пространстве большого числа двухуровневых кубитов, на которые можно осуществлять выборочное воздействие для организации их эволюции.
2. Возможность заготовки входного регистра кубитов исходного базисного состояния и помехоустойчивость вычислительных процессов.
3. Совокупность однокубитовых и двухкубитовых операций, при выборе физической системы, требует определенных нелинейных взаимодействий между управляемыми кубитами, обеспечивающих осуществление двухкубитовых операций.
4. Выполнение измерений и операций вычисления состояния квантовой системы с высокой точностью [1].

На данный момент существует три основных кандидата по реализации подхода создания квантового компьютера: ионные ловушки, квантовые ловушки, ядерный магнитный резонанс.

Целью создания квантового компьютера остается быстрота вычислений и обработки информации. Задачи, которые призван решить квантовый компьютер это:

1. Оптимизация вычислений.
2. Быстрый поиск по мировым базам данных.
3. Ускорение поиска новых средств и проверка уже существующих материалов в различных сферах человеческой деятельности.

Список использованных источников:

1. Валиев К.А. Квантовые компьютеры: надежды и реальность/ К.А. Валиев, А.А. Кокин Cryptomasher via PublishDrive, 2017 – Ижевск: РХД, 2001.

ТЕХНОЛОГИИ 3D ПЕЧАТИ. 3D ПРИНТЕРЫ И СФЕРЫ ИХ ПРИМЕНЕНИЯ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Халецкий Д.М.

Бакунова О.М., ст. преподаватель каф. ИСиТ, м.т.н.

Периферийные устройства объемной печати, приближенные к современным образцам, существуют несколько десятилетий. Создание обычных макетов и объемных моделей становятся все доступнее с совершенствованием периферийных устройств. В работе рассмотрены технологии 3D-печати, 3D-принтеры и сферы применения.

Скорость развития периферийных устройств печати растет с каждым годом. В данное время почти все отрасли современного производства и услуг содержат в себе технологии аддитивного производства. Инженером, который разработал и собрал первое периферийное устройство 3D-печати, был Чарльз Халл. Основанная им компания 3D System выпустила первый аппарат под названием FDM, позже в производство поступила усовершенствованная модель SLA-250. Технологии FDM и SLA применяются и в современных 3D-принтерах.

В основе любой 3D-печати лежит послойное создание объекта посредством спроектированной модели. За счет того, что создание идет слой за слоем, который не превышает толщины листа, все технологии объемной печати исключают полностью или подразумевают малое количество отходов. Со временем технологии печати объемных предметов будут увеличиваться в количестве. На данный момент существует более десятка технологий печати.

Первый тип, который был использован в первой версии 3D-принтера, является экструзия. Экструзия – процесс и метод изготовления изделий путем продавливания материала расплава. В данном типе объемной печати содержатся две технологии. Метод моделирования путем наплавления: выдавливание материала при застывании на заготовленную основу. Минусом данной технологии является скорость воссоздания модели. Основным материалом печати для данной технологии является пластик. Технология робокастинга: используется с керамическим шламом, образуя готовую форму из-за псевдопластичности.

Стереолитография или технология SLA использует в своей основе способ послойного отверждения жидкого полимера с помощью подсветки ультрафиолетом. В данном случае лазер не может напечатать всю модель по толщине, превышающую полимер, поэтому печать производится слой за слоем. Преимущество технологии в возможности печати больших моделей, минимальное количество отходного материала. Минусом данной технологии ограничения в выборе цветов и материала, а также недостаток в мобильности.

Следующей типологией объемной печати является послойное формирование на выровненном слое порошка. Данная типология содержит несколько технологий 3D-печати:

1. ZDP технология нанесения клея струйной печатью;
2. EBM или электронно-лучевая плавка в вакууме металлического порошка лучом;
3. SLM или селективное лазерное спекание чистых металлов лазерным лучом.

Так же в данной типологии присутствует DMSL (прямое лазерное спекание) и SHS (выборочное тепловое спекание тепловой головкой).

Подача проволочного материала EBF плавит материал под действием электронного излучения. Ламинирование LOM – создание объекта путем вырезки лазером в большом количестве слоев контуров сечений. Методы струйного моделирования и точечной подачи порошка сходные с аналогами в 2D печати изображений.

Существует отдельная ветка экспериментальных устройств объемной печати. Биопринтеры – устройством печати 3D-структуры путем нанесения капель живых клеток. Естественные процессы контролируемого роста, деления и модификации приводят объект печати в финальную форму.

В настоящее время широкое распространение в бытовые условия получили принтеры, использующие технологию HPМ(FDM) печати. Данная технология позволяет создавать детали и контракции конечных объектов из термопластика. Из-за почти полного отсутствия отходов вредных для здоровья человека, принтеры данной технологии позволительно использовать в офисах и быту. Для печати используется два вида материалов. Основной материал используется для создания самой детали, а вспомогательный для поддержки. После создания необходимого объекта вспомогательный материал удаляется механическим или химическим путем. Веткой в поколении данных принтеров являются принтеры ручной печати. Устройство рисования трехмерных объектов использует послойное наплавление нити пластика.

Список сфер, в которые невозможно внедрить 3D-печать, с каждым днем уменьшается. Прогресс стремительно осваивает новые рынки применения.

Самой большой сферой применения 3D-принтеров является промышленность.

В машиностроение технологии объемной печати позволяет разрабатывать новые, модернизировать, реставрировать или заменять детали. Использование 3D-печати обеспечивает доступность тестирования новых разработок. Преимуществами объемной печати в данной сфере является масштабируемость и скорость воссоздания. В данное время машиностроение использует 3D-принтеры для: создания прототипов и их тестирования, компонентов и оснастки моделей, литых моделей, создания готовых изделий. В данное время машиностроение использует 3D-принтеры для: создания прототипов и их тестирования, компонентов и оснастки моделей, литых моделей, создания готовых изделий.

Производители электроники используют технологии объемной печати для сокращения сроков изготовления опытных образцов и конечных изделий. С ускорением времени изготовления изделий, открывается возможность снизить затраты на изготовление и дорабатывать изделия путем тестирования, модернизировать

их, реставрировать или заменять детали. Использование 3D-печати обеспечивает доступность тестирования новых разработок. Преимуществами объемной печати в данной сфере является масштабируемость и скорость воссоздания.

Возможность создания масштабируемых детальных моделей в короткие сроки, открыла возможность использования технологии в сфере строительства, архитектуры и дизайна. Изготовления детализированных макетов – основная часть любого архитектурного и строительного процесса. Использование 3D-печати позволяет изменить исходный образец в кратчайшие сроки. В течении последних 10 лет активно развиваются и разрабатываются принтеры строительства зданий. Все прототипы примитивны по своей структуре, так как не позволяют полностью реализовать всю нужды первоначального проектирования. Однако уже сейчас имеется возможность создания объекта, путем послойного нанесения бетона, двухэтажного размера. Развитие данной технологии позволит заметно ускорить строительство объектов, откроет возможность проектирования и воспроизведения сложных архитектурных сооружений, существенно удешевит себестоимость конечного объекта, за счет минимального количества отходов материалов.

Сфера потребительского изготовления товаров включает в себя все основные преимущества технологии 3D-печати. Различные персональные товары уже сейчас, в большинстве случаев, изготавливаются с помощью объемной печати. 3D-принтер открывает возможность создания уникальных персональных предметов быта. Высокая скорость производства, дешевизна изготовления, точность и простота печати с течением времени вытеснит ручное производство.

Образование в сфере 3D-печати одно из перспективных направлений. 3D-принтеры стали появляться в учреждениях образования во всех государствах. Создание полномасштабных моделей существенно ускоряют понимание материала и предоставляет возможность обнаружить ошибку, в отличие от изображения на бумаге.

Пищевая промышленность – один из возможных путей развития 3D-печати. Сфера не получила широкого распространения (основа использования в дизайне и оформлении блюд), однако потенциал ее использования велик. Высказываются идеи, что благодаря уменьшению стоимости техники и возможности долговременного хранения материала для пищевого 3D-принтера (смеси, консервированные, с большим сроком хранения), откроется возможность удешевления и сохранности доставки продукции пищевой промышленности в разные точки планеты.

Само актуальной сферой развития 3D-печати является медицина.

Первой ступенью внедрения объемной печати в медицине выступила стоматология. Принтеры в этой сфере позволили добиться в невероятной точности изготовления изделий. Отпала необходимость в хранении шаблонов, слепков и данных каждого пациента. При сканировании ротовой полости достаточно сохранить построенную 3D-модель в базе данных и при необходимости распечатать нужный образец. С внедрением технологии 3D-печати устраняется необходимость в долгом изготовлении протезов и отдельных частей, как следствие скорость обслуживания клиентов в данной сфере возрастает.

Следующей ступень в развитии является создание протезов средствами 3D-печати.

Время изготовления индивидуального протеза может занимать значительно продолжительное время. Особо качественные изделия из титана имеют высокую цену производства. 3D-печать протезов существенно снижает их стоимость и скорость производства. Точность изготовления, дешевизна возможных материалов и скорость позволяют создавать протезы, которые в десятки раз дешевле титановых, изготовленных на заказ. Поставка на серийное производство различных протезов с более низкой ценой позволит менее обеспеченным людям приобрести устройство.

Последней ступень в развитии 3D-печати медицинской сферы стала биопечать.

Использование тканевой инженерии с 3D-печатью позволяют в настоящее время создавать точные модели различных препаратов, органов и частей тела. Пересадка напечатанных органов в настоящее время не доступна и исследования проводятся исключительно на животных. В 2014 году, с помощью напечатанной точной копии сердца, удалось спасти новорожденного ребенка, у которого был сложный порок сердца. Ученые использовали воспроизведенное сердце как объект исследования. Изготовление косных имплантов производится в настоящее время. Уже были изобретены глазные протезы с помощью 3D-печати. В 2015 году в Китае трехлетнему ребенку был имплантирован напечатанный на 3D-принтере череп. Были произведены операции по имплантированию части напечатанного позвоночника.

С течением времени методы 3D-печати станут наиболее практичными. С использованием данного метода любая отрасль сделает огромный рывок в производстве. С появлением устройств 3D-печати широкого диапазона расходуемых материалов, откроет возможность утилизации и переработки отходов материалов, которые являются одной из самых актуальных проблем.

ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА РАБОЧЕГО ВРЕМЕНИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Харлов А.А.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Коренская И. Н. – ст. преподаватель каф. ИСиТ*

Основой для реализации задачи по мониторингу рабочего времени служит необходимость со стороны работодателей и самих сотрудников контролировать количество отработанного рабочего времени. Контроль персонала необходим, так как организация производственного процесса во многом зависит от дисциплины и мотивации сотрудников. Особенно остро стоит вопрос четкого контроля, когда у работников гибкий график. Эти проблемы обсуждаются в докладе.

Информация об использовании системы учета рабочего времени положительно влияет на дисциплину коллектива, продуктивность работы и безопасность.

Главная задача системы – сохранение в базе сведений о количестве часов, затраченных на определенную задачу, а также их представление в удобном для пользователя виде.

Для удобства использования программного средства разработаны веб и мобильный клиенты. Во избежание дублирования кода, что непременно ведет к ошибкам в системе, необходимо реализовать всю бизнес-логику на удаленном сервере, с которым работают оба клиента.

Разработанное программное средство состоит из нескольких компонентов: база данных, веб-сервер, веб-клиент, мобильный клиент. Архитектура программного средства представлена на рисунке 1.

Для обмена данными между клиентом и сервером осуществляется по протоколу HTTPS с использованием сообщений в формате JSON.

В качестве СУБД для данного программного средства была выбрана Microsoft SQL Server. Microsoft SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Использование данной СУБД обусловлено её надежностью, целостностью, автоматическими обновлениями и производительностью.

Веб-сервер программного средства разработан с применением архитектурного стиля REST, который представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой системы. Разработка серверной части велась на платформе ASP.NET, которая является составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP.

Для разработки клиентской мобильной части был использован фреймворк Xamarin. Xamarin – фреймворк для кроссплатформенной разработки мобильных приложений) с использованием языка C#. Он представляет возможность применения любого языка программирования с полным доступом ко всем возможностям SDK платформы и механизму создания UI, получая на выходе приложение, которое имеет высокую производительность.

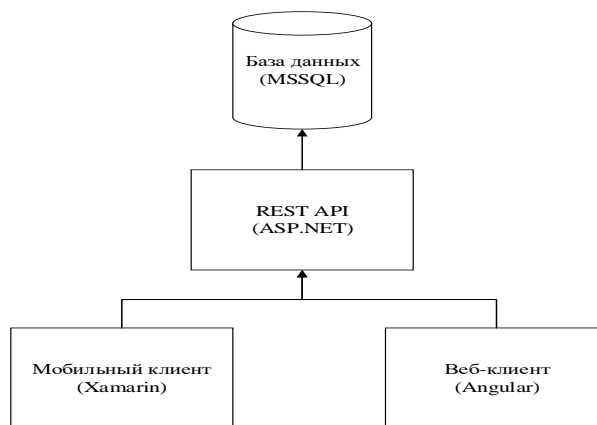


Рисунок 1 – Архитектура программного средства

Для реализации веб-клиента был выбран подход SPA (Single Page Application – одностраничное веб-приложение) и один из лидирующих фреймворков Angular. Среди преимуществ Angular можно выделить: декларативный стиль кода, высокая скорость разработки, модульность, связывание данных, развитое сообщество.

Разработанное программное средство выполняет следующие функции:

- добавление/изменение/удаление клиентов компании;
- создание и редактирование записей таймера;
- отображение статистики пользователя с возможностью сравнения с предыдущим периодом;
- добавление/изменение/удаление проектов;
- просмотр и экспорт отчетов;
- отображение истории таймера в мобильном приложении;
- создание новых записей таймера в мобильном приложении.

Данное программное средство имеет практическое применение и успешно внедрено на предприятии ООО «Синезис».

Список использованных источников:

- 1.Nagel, С. Professional C# 6 and .NET Core 1.0 /С. Nagel – Wrox – 2016. – 1536 pp.
- 2.Xamarin [Электронный ресурс]. – Режим доступа: <https://developer.xamarin.com/>. – Дата доступа: 18.12.2017.

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА АНАЛИЗА ЛИЦЕНЗИОННЫХ НАРУШЕНИЙ В РАЗРАБАТЫВАЕМЫХ ПРОДУКТАХ

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Хваль И.Н.

Леванцевич В.А. ст. преподаватель каф. ПОИТ

На данный момент сложно найти приложение, которое не использовало бы в своей основе программное обеспечение с открытыми исходным кодом. Использование подобных компонентов и библиотек позволяет ускорить процесс разработки приложения и уменьшить количество встречаемых в нём ошибок. В то же время данное программное обеспечение (а также его компоненты, будь это алгоритмы, отдельные файлы исходного кода, скомпилированные библиотеки или изображения) является таким же объектом авторского права, как и любое произведение науки, литературы и искусства, и его использование должно быть выполнено в соответствии с указанными его правообладателями набором требований – лицензией. Выше указанные проблемы обсуждаются в работе и рассматриваются пути их решения.

К сожалению, среди команд разработчиков сильно распространено заблуждение относительно того, что программные продукты с открытым исходным кодом можно использовать как в личных, так и коммерческих целях. В части случаев так и есть – некоторые лицензии действительно не налагают жёстких ограничений на их использование в создаваемых приложениях – но в то же время многие лицензии могут, например, запретить использование компонентов в коммерческих приложениях или потребовать обязательное открытие исходного кода итогового проекта.

В связи с этим в компаниях, специализирующихся на разработке программных продуктов, возникла потребность в системах для обнаружения используемых в приложении компонентов, определении используемых ими лицензий с последующей проверкой их на совместимость с проектом. Благодаря данной информации команда разработчиков может своевременно реагировать на появление в коде компонентов, чья лицензия потенциально может навредить итоговому приложению.

Целью данного дипломного проекта является разработка и внедрение программного средства для осуществления анализа разрабатываемых внутри компании программных продуктов на наличие в нём несовместимых компонентов с открытым исходным кодом.

Назначением рассматриваемой работы является предоставление следующих возможностей:

- возможность сопоставления активных проектов в системе управления проектами компании с соответствующими репозиториями в системах контроля и управления версиями;
- возможность сканирования выгруженных проектов при помощи системы анализа исходного кода проекта на лицензионную совместимость компонентов;
- возможность построения отчётов о результатах сканирования.

Конечными пользователями программного средства являются сотрудники отдела по определению лицензионных нарушений в разрабатываемых программных средствах, взаимодействующие с программным средством через интернет-браузер.

Анализ предметной области и существующих программных средств с аналогичным предназначением позволил оценить их достоинства и недостатки.

На основе результатов анализа были сформулированы функциональные требования к разработанному программному средству:

- получение и обработка информации о проектах из системы управления проектами;
- получение и обработка информации о репозиториях из систем контроля и управления версиями;
- формирование настроек сканирования репозитория;
- выгрузка данных из репозитория;
- проведение сканирования загруженных файлов в системе анализа данных в соответствии с сформированными настройками сканирования репозитория;
- получение отчётной информации, содержащей сведения о результатах сканирования.

Распространённые недостатки – высокая стоимость и избыточная функциональность итогового решения, зависимость результатов анализа от наполненности базы данных компонентов компании-поставщика.

На основе функциональных требований к программному средству и анализа архитектурных подходов по разработке систем, уже существующих на предприятии, было принято решение одностраничное интернет-приложение, серверная часть которого была реализована на языке Java. Данный подход положительно сказался на скорости разработки и поможет в дальнейшем расширении функциональности программного средства. На основе архитектурного решения были выбраны подходящие технологии и спроектирована наиболее подходящая для решения задач приложения база данных.

Результат выполнения дипломного проекта – создание программного средства, которое реализует поставленный бизнесом набор требований и обладающее возможностью для дальнейшего развития и добавления новой функциональности.

Список использованных источников

1. Мухопад, В.И. Экономика и коммерциализация интеллектуальной собственности. Учебник // В.И. Мухопад. – Мск: Инфра-М, 2016. – 572 с.
2. Уоллс, К. Spring 4 для профессионалов / К. Уоллс, – СПб: ДМК Пресс, 2015. – 754 с.

СИСТЕМА ИЗМЕРЕНИЯ ДАВЛЕНИЯ В ШИНАХ АВТОМОБИЛЯ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Хведченя А.В.

Стешенко П.П. доцент каф. ПЭ, к.т.н., доцент

Рассматривается система автоматического измерения давления в шинах, используемая в автомобилях. Предложена структурная и разработана электрическая схема устройства. Представлен алгоритм ее работы.

Система контроля давления в шинах предназначена для предупреждения об опасном изменении давления в шинах [1].

При нормальном давлении в шине протектор контактирует с дорожным покрытием по всей своей ширине, что способствует наилучшему сцеплению колеса с дорогой и равномерному износу протектора. Кроме того, при снижении давления в шине увеличивается податливость боковин и их способность сильнее деформироваться под действием боковых инерционных сил, возникающих при любом движении автомобиля не по прямой. Боковой увод существенно отражается на управляемости и устойчивости машины.

Существует две системы измерения давления в шинах [2].:

- система косвенного измерения давления в шинах;
- система прямого (непосредственного) измерения давления в шинах.

Система косвенного измерения давления в шинах работает благодаря дополнительной модификации антиблокировочной системы торможения (ABS). Данная система TPMS определяет снижение давления внутри шины путем просчета количества оборотов колеса за определенную дистанцию: при снижении давления колеса уменьшается его диаметр и скорость вращения. Эта информация передается бортовому компьютеру, после чего загорается индикатор TPMS. При замене шин или выполнении ремонта ходовой части автомобиля косвенная система TPMS требует перенастройки параметров – калибровку. К минусам данной системы можно отнести и то, что TPMS не может сравнивать скорость движения каждого колеса по отдельности, поскольку движения автомобиля очень редко происходит по прямой линии. Система контроля разработана таким образом, что скорость каждых двух колес, которые расположены по диагонали, суммируется, из полученных результатов вычисляется разница и делится на среднюю скорость каждого колеса. То есть, когда водитель видит сигнал индикатора, он еще не знает, в каком именно колесе понизилось давление. Чтобы это определить, ему необходимо выйти из салона и самостоятельно осмотреть каждое колесо.

Система косвенного контроля давления шин имеет следующие недостатки:

- не позволяет определить резкое падение давления при пробое колеса;
- система не указывает на одновременное падение давления в колесах;
- на работоспособность системы значительно влияют такие факторы, как степень пробуксовки колес, загрузка автомобиля и состояние автомобильной резины;
- чтобы индикатор засветился, давление в колесах должно упасть не менее, чем на 25-30%;
- не определяется давление в колесах до начала движения автомобиля;
- большая длительность калибровки

Косвенная система контроля давления указывает водителю только на изменение параметров управления автомобилем [3].

Преимущество системы непрямого (косвенного) измерения давления в шинах – это низкая цена, так как не требуется установка дополнительных элементов на колеса с их последующим обслуживанием или заменой.

Нами предложена система прямого измерения давления в шинах. Модули непосредственного измерения монтируются на внутренней стороне колесного диска и закрепляются на вентиле для накачивания шин (или вместо него). Каждый модуль измеряет текущее давление шины и по радиоканалу передает данные в специализированный контроллер или электронную систему управления автомобиля. Модули питаются от несъемных элементов питания. Датчики непосредственного измерения, напрямую отслеживают давление каждой шины. Система включает следующие блоки (рисунок 1):

- блоки преобразования сигнала с датчиками давления;
- блок управления с дисплеем.

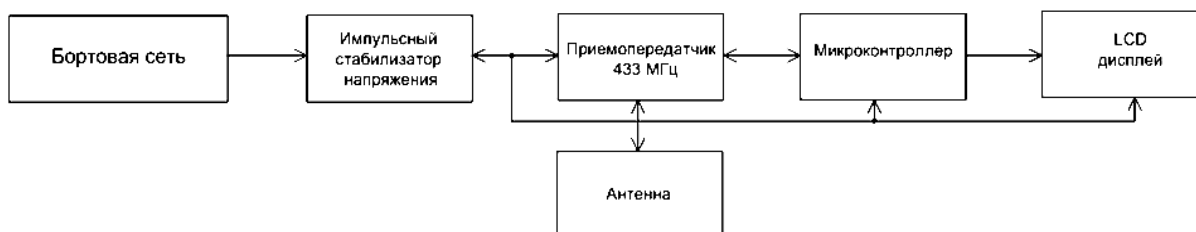


Рисунок 1 – Структурная схема измерения давления в шинах автомобиля

Стабилизатор напряжения предназначен для преобразования автомобильного бортового напряжения 12В в требуемое напряжение 5В для питания схемы.

Модуль приемопередатчика, работающего на частоте 433 МГц, является входным и выходным трактом радиоканала блока управления [4].

Датчик давления преобразует величину давления в электрический сигнал. Приемная антенна осуществляет прием сигналов от датчиков давления и передачу их в блок управления. В качестве приемной антенны возможно использование антенны центрального замка автомобиля.

Источник питания представляет из себя гальванический элемент на основе диоксида литий-марганца (Li/MnO₂). Он поддерживает работоспособность датчика в течение 5-10 лет.

Для блока управления представлен алгоритм работы всей системы и разработана печатная плата в программе DipTrace.

Список использованных источников.

1. Система контроля давления в шинах [Электронный ресурс]. – Режим доступа: http://systemsauto.ru/electric/tires_pressure.html. – Дата доступа: 27.10.2017г.

2. Система контроля давления в шинах [Электронный ресурс]. – Режим доступа: <http://www.automaster.net.ua/drukujpdf/artikul/50224>. – Дата доступа: 10.11.2017г.

3. Система контроля давления в шинах [Электронный ресурс]. – Режим доступа: http://myauto26.ru/electric/tires_pressure.php. – Дата доступа: 16.11.2017г.

4. Радиочастотные микросхемы в решениях для систем контроля давления в шинах [Электронный ресурс]. – Режим доступа: <http://www.rlocman.ru/review/article.html?di=130443>. – Дата доступа: 16.11.2017г.

АВТОРИЗАЦИЯ НА ОСНОВЕ ПОЛИТИКИ ASP.NET CORE

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Черник Д.В.

Бакунова О.М. – ст. преподаватель каф. ИСиТ, м.т.н.

Бакунов А.М. – ст. преподаватель каф. ИСиТ, м.т.н.

Калетня И.Л. ассистент каф. ИСиТ, м.т.н.

Механизм авторизации программного обеспечения гарантирует, что текущему пользователю разрешен доступ к данному ресурсу, выполнить задание или выполнить задание на данном ресурсе. В ASP.NET Core существует два способа настройки уровня авторизации. Вы можете использовать механизм предоставления ролей или же использовать политики авторизации. Авторизация на основе ролей основана на предыдущих версиях платформы ASP.NET, в то время как авторизация на основе политики является новой для ASP.NET Core.

Роли использовались в приложениях ASP.NET с релиза технологии. Технически говоря, роль - простая строка. Ее значение, однако, рассматривается как метаданные для механизма безопасности приложения (проверяется на наличие в объекте IPrincipal) и используется приложениями для сопоставления набора прав и разрешений конкретному аутентифицированному пользователю. В ASP.NET вошедший в систему пользователь идентифицируется объектом IPrincipal, а в ASP.NET Core ему соответствует класс ClaimsPrincipal. Этот класс предоставляет набор идентификаторов, и каждый идентификатор представлен объектами Identity, в частности ClaimsIdentity. Это означает, что любой зарегистрированный пользователь имеет список претензий (claims), которые являются сведениями о его статусе. Имя пользователя и его роль – два общих типа претензий для пользователей приложения ASP.NET Core. Однако наличие или отсутствие роли зависит от хранилища идентификационных данных. Например, если вы используете социальную аутентификацию, вы никогда не увидите роли.

Авторизация следует за аутентификацией. Аутентификация — это поиск личности пользователя, тогда как авторизация — это определение прав, которыми пользователь обладает для доступа к конечным точкам приложения. Роли пользователя обычно хранятся в базе данных и извлекаются, когда учетные данные пользователя проверяются, и в какой момент работы приложения информация о роли связывается с учетной записью конкретного пользователя. Интерфейс Identity имеет метод IsInRole, который должен быть реализован для корректной проверки учетной записи пользователя. Класс ClaimsIdentity делает это, проверяя, что утверждение Role доступно в коллекции претензий, полученных в результате процесса аутентификации. В любом случае, когда пользователь пытается вызвать метод защищенного контроллера, его роль должна быть доступна для проверки. В противном случае пользователю запрещается вызов любых защищенных методов.

Атрибут Authorize — это декларативный способ защиты контроллера или некоторых его методов.

Указанный без аргументов атрибут проверяет только аутентифицирован ли пользователь. Однако атрибут также поддерживает некоторые дополнительные параметры, такие как Roles. Свойство Roles указывает, что пользователям в любой из перечисленных ролей будет предоставлен доступ. Чтобы потребовать несколько ролей, вы можете применить атрибут Authorize несколько раз или написать собственный фильтр.

При желании атрибут Authorize также может принимать одну или несколько схем аутентификации через свойство ActiveAuthenticationSchemes.

Свойство `ActiveAuthenticationSchemes` — это строка с разделителями-запятыми, в которой перечислены компоненты промежуточного ПО аутентификации, которым механизм авторизации будет доверять в текущем контексте. Другими словами, он утверждает, что доступ к классу контроллера разрешен только в том случае, если пользователь аутентифицирован по выбранной схеме и имеет любую из перечисленных ролей. Строковые значения, переданные в свойство `ActiveAuthenticationSchemes`, должны совпадать с промежуточным ПО проверки подлинности, зарегистрированным при запуске приложения.

В ASP.NET 2.0 промежуточное ПО проверки подлинности заменяется службой с несколькими обработчиками. В результате схема аутентификации является меткой, которая выбирает необходимый обработчик.

Информация, предоставляемая атрибутом `Authorize`, используется встроенным фильтром авторизации. Поскольку он отвечает за проверку того, может ли пользователь выполнить запрошенную операцию, этот фильтр запускается перед любым другим фильтром ASP.NET Core. Если пользователь не авторизован, фильтр останавливает конвейер обработки и отменяет запрос.

Общая схема движения HTTP запроса в ASP.NET приложении имеет следующий вид (рисунок 1):

Роли — это простой способ группировать пользователей приложения на основе того, что они могут или не могут сделать. Но они не очень выразительны; по крайней мере, недостаточно для удовлетворения потребностей большинства современных приложений.

Роли по существу являются плоскими понятиями. Как только количество вариантов использования приложения возрастает, количество требуемых ролей значительно возрастает. Из-за минусов системы авторизации на базе ролей и в результате эволюции платформы, была создана авторизация на основе политики.

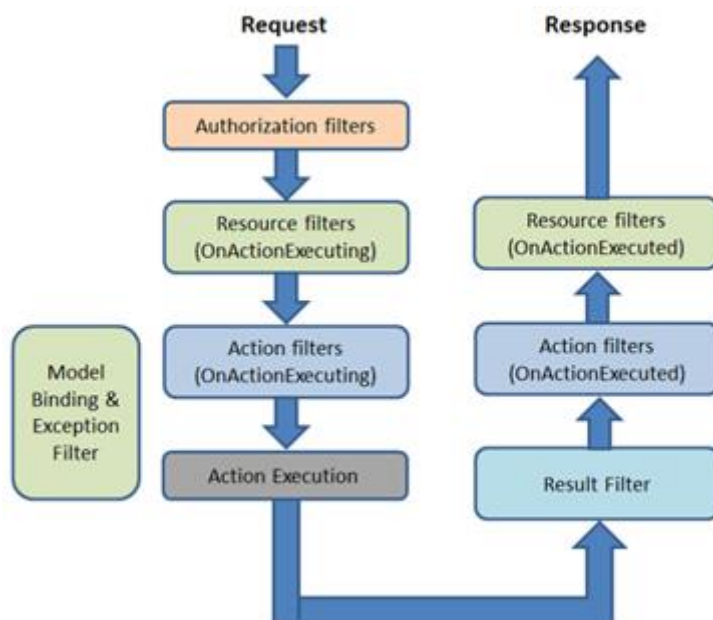


Рисунок 1 - Схема движения HTTP запроса

В ASP.NET Core, основанный на политиках механизм авторизации предназначен для разделения слоев авторизации и логики приложения. Проще говоря, политика — это сущность, разработанная как совокупность требований, которые сами по себе являются условиями, которым должен удовлетворять текущий пользователь.

Простейшая политика представляет собой набор требований, наиболее общими из которых являются: пользователь аутентифицирован, пользователь связан с данной ролью. Другое общее требование заключается в том, чтобы пользователь имеет конкретное претензии(claim) или конкретное требование с определенным значением. В наиболее общих терминах претензии— это утверждение о пользователе, который пытается получить доступ к методу, что претензия установлена и имеет значение true.

Для создания политики необходимо использовать объект `AuthorizationPolicyBuilder`, который предназначен для конфигурирования и сбора требований, используя различные методы расширения, а затем создает экземпляр политики. Требования действуют на статус аутентификации и схемы, роль и любую комбинацию претензий, прочитанных через файл Cookie или токен-носитель.

Если ни один из предопределенных методов расширения для определения требований вас не устраивает, вы всегда можете прибегнуть к определению собственного метода расширения для объекта `AuthorizationPolicyBuilder`.

Однако недостаточно просто создать объект политики - вы также должны зарегистрировать вашу политику для механизма авторизации используя промежуточное ПО. Вы делаете это, добавляя промежуточное ПО как службу в методе `ConfigureServices` для класса `Startup`. Каждая политика, добавленная к промежуточному ПО, имеет имя, которое используется для ссылки на политику в атрибуте `Authorize` в классе контроллера.

Атрибут `Authorize` позволяет вам устанавливать политику декларативно, но политики также могут быть запрограммированы программно прямо в методе действия. Если программная проверка разрешений не удалась, вы можете вернуть объект `ForbiddenResult`. Другой вариант - возвращение объекта `ChallengeResult`.

Также вы имеете возможность выполнить программную проверку политик в режиме Razor. Для этого вы должны сначала ввести зависимость от службы авторизации. Использование службы авторизации в представлении может помочь скрыть элементы пользовательского интерфейса, которые не должны находиться в пределах досягаемости текущего пользователя, учитывая текущий контекст. Однако имейте в виду, что просто скрывать варианты в представлении недостаточно. Вы всегда должны применять политики в контроллере.

Базовые требования политики в основном охватывают претензии(claims), аутентификацию и обеспечивают универсальный механизм настройки на основе утверждений, но вы также можете создавать пользовательские требования. Требование к политике состоит из двух элементов: класса требований, содержащего только данные, и обработчика полномочий, который проверяет данные пользователя. Пользовательские требования расширяют ваши возможности настройки определенной политики.

Требование должно иметь по крайней мере один обработчик авторизации. Обработчик имеет тип `AuthorizationHandler<T>`, где T - тип требования.

Обработчик требований авторизации считывает претензии, связанные с конкретным пользователем, и проверяет на соответствие с собственными претензиями. Если всех соответствий не найдено, обработчик возвращает отрицательный результат. Положительный результат возвращается только в том случае, если претензии существует и содержит указанное значение. Предполагается, что пользовательские претензии будут частью информации, связанной каким-то образом с пользователем, например, столбцом в одной из таблиц базы данных.

В свою очередь обработчик авторизации вызывает метод `Succeed`, передавая текущее требование, чтобы уведомить, что требование было успешно проверено. Если это требование не прошло валидацию, обработчик не должен ничего делать и может просто вернуться к дальнейшей работе. Однако, если обработчик хочет определить отказ требования независимо от того, что другие обработчики поэтому же требованию могут завершиться успешно, он вызывает метод `Fail` на объекте контекста авторизации.

Также, вам необходимо зарегистрировать новый обработчик с системой DI в рамках типа обработчика `IAuthorization`.

Каждое требование может иметь несколько обработчиков. Когда несколько обработчиков зарегистрированы в системе DI для одного и того же требования достаточно, чтобы по крайней мере один был завершен успешно.

Авторизация на основе политики — это новый подход, который обеспечивает более богатую и выразительную модель. Это связано с тем, что политика представляет собой набор требований на основе требований и пользовательской логики на основе любой другой информации, которая может быть введена из контекста HTTP или внешних источников. Эти требования связаны с одним или несколькими обработчиками, которые отвечают за фактическую оценку требования.

Список используемых источников

1. Документация ASP.NET Core. [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/aspnet/core>. – Дата доступа: 20.03.2018.
2. Документация .NET [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/library> – Дата доступа: 23.03.2018.
3. Документация и исходный код. [Электронный ресурс]. – Код доступа: <https://github.com/aspnet> . – Дата доступа: 15.03.2018.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ АДМИНИСТРАТИВНОЙ ПАНЕЛЬЮ MAGENTO

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Черный М.С.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Сечко Г. В. – доцент каф. ИСиТ, к.т.н., доцент*

Объектом исследования является программное обеспечение выполняющее функцию управления административной панелью Magento с мобильного устройства. В административной панели Magento располагается информация о пользователях ресурса, продуктах, заказах и транзакциях. Также ее главной сильной стороной является обилие встроенных функций: речь идет о валюте, языках, скидках и купонах и многом другом.

Для любой системы онлайн-торговли очень важно обеспечить непрерывность функционирования (отсутствие простоев и замедлений в работе). По сути, это является ключевым требованием, которое крайне важно, как с точки зрения маркетинга и PR, так и с точки зрения бизнеса в целом.

Целью проекта является разработка программного обеспечения для управления административной панелью Magento, с целью своевременного обнаружения неполадок в работе сервера и получения актуальной информации о текущем состоянии аппаратного обеспечения, товарах, пользователях и отчетах за выбранный период.

Для выполнения поставленной задачи было разработано клиент-серверное программное средство, состоящее из iOS приложения, а также серверной части, представленной в виде Magento и Gateway серверов. Архитектура программного средства представлена на рисунке 1.

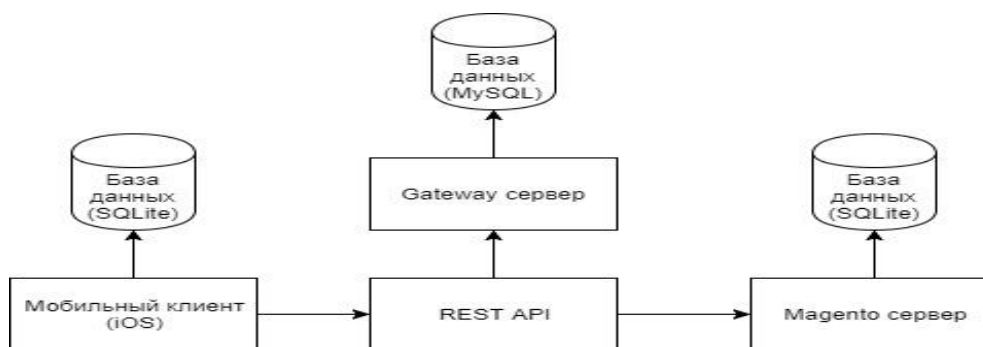


Рисунок 1 – Архитектура программного средства

Каждый из описанных элементов обладает собственной базой данных. Коммуникация между клиентской и серверной частью осуществляется по протоколу HTTPS с использованием сообщений в формате JSON.

Веб-сервер программного средства разработан с применением архитектурного стиля REST, который представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой системы. Разработка серверной части велась на языке программирования PHP.

В качестве СУБД для мобильного приложения была выбрана SQLite, доступ к базе данных осуществляется через CoreData.framework разработанного компанией Apple Inc.

Core Data – это мощный и гибкий фреймворк для хранения и управления графом модели. Основные критерии, обуславливающий данный выбор:

- высокая производительность;
- возможность расширяемости;
- быстрдействие и энергоэффективность.

Для разработки мобильного приложения использовалась IDE XCode с использованием языка Swift. Swift – это надёжный и интуитивно понятный язык программирования, при помощи которого можно создавать приложения для iOS, Mac, Apple TV и Apple Watch.

Разработанное программное средство может работать как на планшетах, так и на смартфонах под управлением операционной системы iOS.

Мобильное приложение реализует следующие функции:

- предоставление информации о пользователях, продуктах, заказах;
- рассылку SMS сообщений и push-notification уведомлений;
- предоставление данных о аппаратном обеспечении и исключительных ситуациях;
- генерацию отчетов по заданным критериям.

Список использованных источников.

1. Коннолли, Т. Базы данных: проектирование, реализация и сопровождение [текст] / Т. Коннолли. – М. 2001.
2. Нахавандипур, В. IOS. Разработка приложений для iPhone, iPad и iPod / В. Нахавандипур; Пер. с англ. О. Сивченко. – СПб.: Питер, 2013.

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ВЗАИМООТНОШЕНИЯМИ С КЛИЕНТАМИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Черняевский С.В.

Бакунова О.М. - ст. преподаватель каф. ИСиТ, м.т.н.

Система управления взаимоотношениями с клиентами или CRM-система - это программное обеспечение для организаций, предназначенное для автоматизации процессов взаимодействия с клиентами. Целью внедрения данной системы является грамотное построение и оптимизация бизнес-процессов организации, тем самым повышение уровня продаж и улучшение обслуживания клиентов. Достигается это путем создания хранилища информации о клиентах, сохранения истории взаимоотношений с ними, планирования и анализа маркетинговых мероприятий, оптимизации работы сотрудников путем фиксации конкретных задач по клиенту.

Характерными признаками необходимости внедрения CRM-системы являются: наличие постоянно растущего списка клиентов, наличие расширенного спектра товаров и услуг, сложный процесс доведения продукта или услуги до конечного клиента, постоянно меняющиеся условия рынка и высокая конкуренция, потребность в прозрачности управления организацией.

На рынке существует достаточное количество готовых решений, которые можно внедрить в организацию. Вместе с этим присутствуют организации со специфичными бизнес-процессами и определенными

требованиями к системе, что делает невозможным использование готовых решений и требует разработки собственной CRM.

Разработка системы включает в себя основные этапы характерные для любого программного обеспечения. При этом на каждом этапе существуют нюансы, свойственные для разработки только данного типа систем. Первый и один из самых важных этапов, который проводится до проектирования будущей системы - это определение целей внедрения данной системы, а также описание текущих бизнес-процессов организации. После описания и оптимизации структуры бизнес-процессов организации наступает этап проектирования будущей системы. Дальнейшим этапом является непосредственно разработка CRM-системы. При реализации модулей системы разработчикам необходимо обратить внимание на безопасность и отказоустойчивость системы. Далее производится подготовка инфраструктуры будущей системы. Последующим этапом является этап внедрения.

Создание CRM системы сопряжено с многими трудностями и требует довольно больших затрат со стороны организации. При этом, если у бизнеса есть специфика, которая не позволяет использовать готовые решения, разработка данной системы оправдана и окупает затраты на ее реализацию, внедрение и поддержку за счет повышения уровня продаж, улучшения обслуживания клиентов и тем самым повышения их лояльности, а также сокращения издержек на управление процессами взаимодействия с клиентами, предоставляя возможность для дальнейшего роста и развития организации.

Список используемых источников:

1. Гринберг, Пол. CRM со скоростью света. Привлечение и удержание клиентов в реальном времени через Интернет/Пол Гринберг – СПб.: «Символ-Плюс» 2006. – 528с.
2. Эдриан Пейн Руководство по CRM. Путь к совершенствованию менеджмента клиентов/Эриан Пэйн – СПб.: «Гревцов Паблишер» 2007. – 384с.
3. Трофимов, Сергей. CRM для практиков/Сергей Трофимов – М.: «АвтоКод» 2006. – 304 с.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ПЕРСОНАЛЬНЫМИ ФИНАНСАМИ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Чернянин Р.И.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Иванков А. С. – зам. нач. отдела разработки*

Обсуждаются результаты разработки мобильного приложения для управления персональными финансами, позволяющего вести учет и статистику расходов пользователя. Приложение накапливает информацию в виде, позволяющем анализировать персональные финансовые операции за любой промежуток времени.

В процессе проектирования был проведен анализ существующих аналогов и выявлены их преимущества и недостатки. На основе этого анализа было определено, какие функции необходимо реализовать в данном программном средстве. В экономическом разделе был проведен расчет экономической целесообразности разработки данного программного средства, из которого следует что рентабельность инвестиций проекта составит 175%.

В работе с использованием всех возможностей языка программирования Java были реализованы следующие функции:

- добавление/изменение/удаление финансовых операций;
- добавление/изменение/удаление расписания;
- изменение настроек приложения и системных данных (категории и валюты);
- просмотр отчетов о финансовых операциях;
- экспорт данных о финансовых операциях в файл.

При проектировании баз данных было использовано приложение DB Browser for SQLite. Браузер баз данных для SQLite представляет собой визуальный инструмент с открытым исходным кодом для создания, разработка и редактирование файлов баз данных, совместимых с SQLite. Он использует привычный табличный интерфейс, поэтому не нужно изучать сложные SQL-команды.

Для разработки самого приложения использовалась среда разработки Android Studio. Это интегрированная среда разработки для работы с платформой Android. Данная среда имеет ряд преимуществ над конкурентами:

- разработчиком данной IDE является компания Google, что означает более удобную интеграцию всех необходимых функций для новых версий Android;
- встроенный SDK (software development kit);
- удобный конструктор интерфейсов;
- информативные логи.

Таким образом, при помощи современных средств разработки было создано мобильное приложение для управления персональными финансами, которое успешно выполняет поставленные задачи.

Список использованных источников.

1. SqLitebrowser [Электронный ресурс]. – Режим доступа: <http://sqlitebrowser.org/>. – Дата доступа: 17.01.2018.

2. Awesomedevlop [Электронный ресурс]. – Режим доступа: <http://awesomedevlop.blogspot.com.by/2016/12/android-studio.html>. – Дата доступа: 17.01.2018.

ПРОГРАММНОЕ СРЕДСТВО ПО ОРГАНИЗАЦИИ ТРАНСЛЯЦИИ ПОТОКОВОГО ВИДЕО ПОСРЕДСТВОМ WOWZA STREAMING ENGINE

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Шилкин А.О.

*Пачинин В. И. – зав. кафедрой ИСиТ, к.т.н., доцент
Образцова О. Н. – доцент каф. ИСиТ, к.т.н., доцент*

В работе представлена разработка программного средства обеспечивающего организации трансляции потокового видео посредством wowza streaming engine.

Прямой эфир - это процесс передачи информации в реальном времени, трансляция видео и аудио сигналов с места проведения записи в эфир. Основной особенностью прямого эфира является то, что получаемые данные невозможно редактировать.

В настоящее время метод передачи данных используется повсеместно во всех видах электронных средствах массовой информации. Это различные развлекательные передачи, викторины, интервью, новости. На телевидении в прямом эфире сегодня выходит не так много программ. Чаще всего это новостные сюжеты в жанре прямого отчета, прямые трансляции футбольных матчей, спортивных соревнований, музыкальных конкурсов, важных политических событий.

Среди всех положительных черт прямого эфира, в первую очередь нужно выделить оперативность. Аудитория получает самую свежую информацию «из первых рук». Это подогревает интерес зрителя и вызывает у него желание узнать о случившемся первым.

Главная задача системы – показ потокового видео для пользователей, которые приобрели подписку на просмотр в системе.

В основе работы программного средства будет положена микросервисная архитектура. Это модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам. Программные комплексы, разработанные в соответствии с сервис-ориентированной архитектурой, реализуются как набор веб-служб, взаимодействующих по протоколу REST. Интерфейсы компонентов в сервис-ориентированной архитектуре инкапсулируют детали реализации (операционную систему, платформу, язык программирования) от остальных компонентов, таким образом обеспечивая комбинирование и многократное использование компонентов для построения сложных распределённых программных комплексов, обеспечивая независимость от используемых платформ и инструментов разработки, способствуя масштабируемости и управляемости создаваемых систем.

Приложение состоит из следующих серверов:

- сервер Admin microservice – является интерфейсом для управления, контентом, пользователями, подписками;
- сервер Content microservice – предназначен для управления контентом, предоставления данных обычному пользователю, все данные представлены в json;
- сервер User microservice – предназначен для регистрации, авторизации пользователей, все данные представлены в json;
- сервер Subscription microservice – предназначен для оформления подписки, проверки наличия подписки для предоставления контента, все данные представлены в json;
- сервер Frontend – является интерфейсом для обычного пользователя;
- сервер Wowza Streaming Engine – предназначен для транслирования прямых трансляций;
- хранилище файлов – предназначен для хранения картинок и видео файлов;
- сервер Memcached – хранилище для кеша.

Для каждого микросервиса база данных располагается на отдельном сервере. Благодаря этому, нагрузка распределяется на несколько баз данных, вместо одной.

Передача данных осуществляется через протокол TCP/IP.

На рисунке 1 представлена диаграмма развертывания программного средства, демонстрирующая конфигурацию и связь узлов программной системы.

Для обмена данными между клиентом и сервером осуществляется по протоколу HTTPS с использованием сообщений в формате JSON.

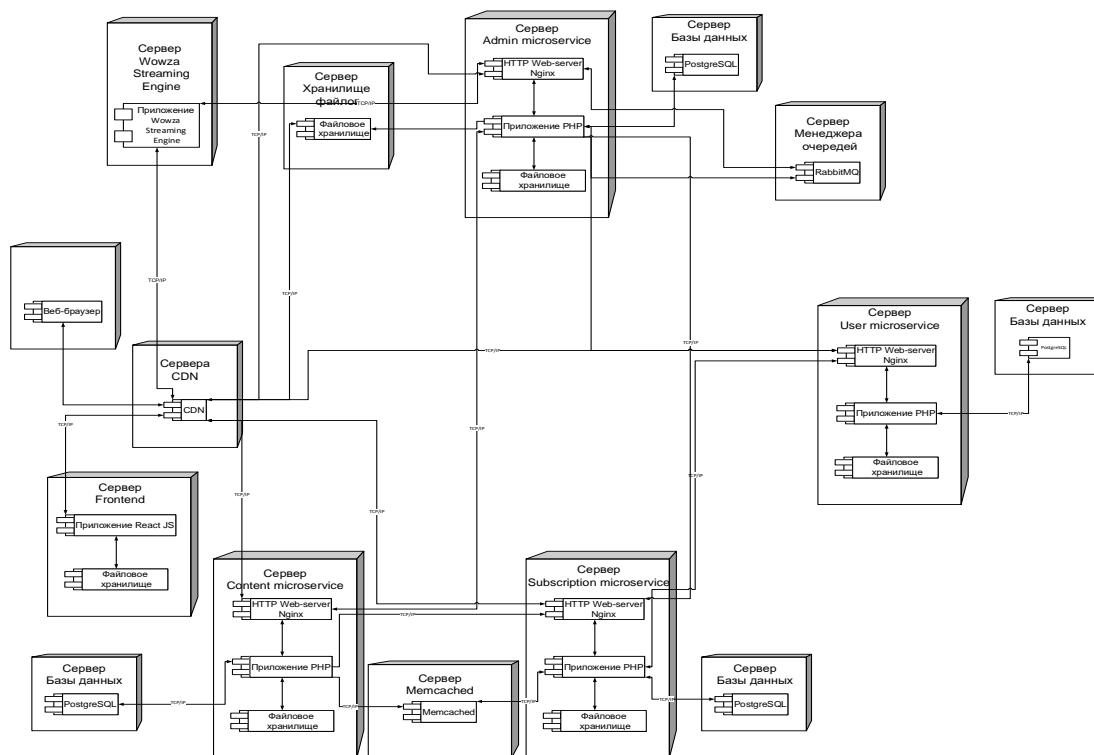


Рисунок 1 - Диаграмма развертывания

В качестве СУБД для данного программного средства была выбрана PostgreSQL. PostgreSQL не просто реляционная, а объектно-реляционная СУБД. Это даёт ему некоторые преимущества над другими SQL базами данных с открытым исходным кодом, такими как MySQL, MariaDB и Firebird.

Фундаментальная характеристика объектно-реляционной базы данных — это поддержка пользовательских объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Это делает PostgreSQL невероятно гибким и надежным. Среди прочего, он умеет создавать, хранить и извлекать сложные структуры данных. В некоторых примерах ниже вы увидите вложенные и составные конструкции, которые не поддерживаются стандартными СУБД.

Существует обширный список типов данных, которые поддерживает PostgreSQL. Кроме числовых, с плавающей точкой, текстовых, булевых и других ожидаемых типов данных (а также множества их вариаций), PostgreSQL может похвастаться поддержкой uuid, денежного, перечисляемого, геометрического, бинарного типов, сетевых адресов, битовых строк, текстового поиска, xml, json, массивов, композитных типов и диапазонов, а также некоторых внутренних типов для идентификации объектов и место-положения логов. Справедливости ради стоит сказать, что MySQL, MariaDB и Firebird тоже имеют некоторые из этих типов данных, но только PostgreSQL поддерживает их все:

- сетевые адреса;
- многомерные массивы;
- геометрические данные;
- данные в формате JSON.

– Веб-сервер программного средства разработан с применением архитектурного стиля REST, который представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой системы.

– Для реализации веб-клиента был выбран подход SPA (Single Page Application – одностраничное веб-приложение) и один из лидирующих фреймворков ReactJs. Среди преимуществ ReactJs можно выделить: декларативный стиль кода, высокая скорость разработки, модульность, связывание данных, развитое сообщество.

Разработанное программное средство выполняет следующие функции:

- управление пользователями: добавление, редактирование, удаление, блокирование, смена пароля;
- управление группами: для каждой группы можно выставлять права доступа и привязывать пользователя к группе;
- управление подписками: отмена подписки, просмотр статуса, просмотр транзакций;
- добавление, редактирование, удаление каналов, программ, сезонов, эпизодов;
- начинать и заканчивать запись прямых трансляций;
- видео должно быть зашифровано технологией AES-128;
- трансляция при завершении должна конвертироваться в видео по запросу.

Список использованных источников:

1. Ньюмен, С. Создание микросервисов. / Сэм Ньюмен – СПб.: Питер, 2015. – 304 с.
2. Мэттью, Нейл. PostgreSQL. Основы / Нейл Мэттью, Ричард Стоунз – Москва: Символ, 2014. – 640 с.

АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННОГО ПРОЦЕССА ИЗГОТОВЛЕНИЯ ЭЛЕКТРОННЫХ ИЗДЕЛИЙ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Шпилевский В.В

Скудняков Ю.А. - доцент каф. ПЭ, к.т.н., доцент

Приводится описание производственного технологического процесса по автоматизации изготовления электронных изделий для достижения высокой производительности, снижения производственных издержек и повышения качества выпускаемой продукции.

В современных производственно-технологических процессах не обойтись без их автоматизации и оптимизации. Все это необходимо для повышения производительности, снижения производственных издержек и повышения качества выпускаемой продукции. В качестве примера рассмотрим производственный технологический процесс по выпуску электронных изделий. Вся электронная продукция состоит в обязательном порядке из платы с радиоэлементами. При изготовлении электронных изделий применяются специализированные установки автоматической пайки. Принцип работы данной установки в следующем: сначала изготавливаются трафареты печатных плат, по которым наносится паяльная паста, после этого пишется программа для установки каждого элемента на нужную позицию и заносится в компьютер установки. Когда программа готова, в установку размещается трафарет, по которому наносят паяльную пасту (процедура нанесения пасты автоматическая). Платы с нанесенной пастой поступают далее по ленте и с помощью вакуумного пинцета устанавливаются на нужные места. После этого плата с радиоэлементами поступает в камеру тепла, где нагревается до определенной температуры и в течении необходимого времени сушится (температура и время зависят от типа и характеристик паяльной пасты). При выходе из установки плата подвергается контролю качества. Сам контроль заключается в осмотре платы на предмет незатвердевшей пасты и, самое главное, в отсутствии мелких шариков, которые возникают при нарушении режимов сушки (они приводят к коротким микросхем и других элементов). Но в автоматических установках можно использовать только SMD-элементы (элементы, предназначенные для поверхностного монтажа), а все остальное устанавливается вручную. Таким образом, использование специализированных установок автоматической пайки позволяет повысить производительность производственного технологического процесса и качество выпускаемой продукции. Снижение производственных издержек тоже очень велико, пропадает необходимость в постоянной закупке паяльных станций (цена хорошей станции составляет примерно \$1000) и комплектов сменных жал (цена одного жала составляет примерно \$10). Наносится меньший вред окружающей среде и здоровью людей, так как в производстве используются припои, содержание свинца в которых 61% (ПОС-61), а также флюс и спиртно-нефрасовая смесь для отмывки плат от флюса. Для повышения уровня автоматизации и оптимизации производственного технологического процесса производства электронной техники следует использовать современные гибкие производственные системы (ГПС), которые можно отнести к природоподобным системам, моделирующим деятельность человека и являющимся высшей, наиболее развитой формой автоматизации производственного процесса [1]. К основным принципам организации ГПС можно отнести: 1) принцип совмещения высокой производительности и универсальности, использование которого позволяет обеспечить универсальность и автоматизацию в программно-управляемом и программно-перенастраиваемом оборудовании. При этом ГПС сравнимы по производительности с автоматическими линиями, а по гибкости - с универсальным оборудованием; 2) принцип модульности, когда ГПС строится на базе гибких производственных модулей, разработанных для основных видов производств изделий электронной техники; 3) принцип иерархичности ГПС – предусматривает построение многоуровневой структуры. На самом нижнем уровне находятся гибкие автоматизированные модули, на высших уровнях - гибкие автоматизированные линии, участки, цехи, предприятия в целом. Модульность и иерархичность позволяют разрабатывать ГПС для самого высокого организационного структурного уровня; 4) принцип преимущественной программной настройки, где оборудование ГПС, как основное, так и вспомогательное, при смене изделий перенастраивается путем ввода новых управляющих программ модулей; 5) принцип обеспечения максимальной предметной замкнутости производства на возможно более низком уровне структуры ГПС – позволяет свести к минимуму затраты на транспорт и манипулирование. Одновременно достигается снижение количества операций при общем повышении гибкости ГПС; 6) принцип совместимости технологических, программных, информационных, конструктивных, энергетических и эксплуатационных элементов.

Список использованных источников:

1. Александрова, А.Т. Гибкие производственные системы электронной техники/ А.Т. Александрова, Е.С. Ермаков. - М.:Высш.шк.,2003.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА АНАЛИЗА И УЧЁТА ИЗМЕНЕНИЙ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Ярмолович О.М., Зайкина И.С.

Моженкова Е.В. – ассистент каф. ИСиТ, м.т.н.

В работе рассмотрены особенности оздания и функционирования систем анализа и учета изменений аппаратно программных комплексов организаций в Республике Беларусь.

При создании автоматизированной системы на предприятии одним из первых и наиболее важных шагов проектирования является анализ деятельности, которая подвергается автоматизации. В анализ деятельности входит описание бизнес-процессов, происходящих на предприятии и принятие решения о разработке новой корпоративной информационной системы, либо выбора наиболее подходящего решения на ИТ-рынке [1].

Компания ОДО «Виталюр» на сегодняшний день занимает лидирующие позиции в ритейле Республики Беларусь. С каждым годом сеть магазинов предприятия увеличивается. Количество сотрудников за последний год выросло на полтысячи и сейчас составляет около пяти тысяч человек. На предприятии существует целый ряд оборудования, которое необходимо учитывать, проводить инвентаризацию оргтехники и других материальных ценностей. Каждый месяц предприятие закупает новое оборудование и списывает старое, перемещает и складировать технику, поэтому со временем процесс учета оборудования становится более трудоемким. Можно потерять много времени, перебирая документы каждой единицы техники, каждого сотрудника, отслеживая внутренние перемещения оборудования, поиски недостатков по отделам, кабинетам, этажам и офисам. Рост парка оборудования привел к необходимости внедрения средства автоматизации учета и изменения программно-аппаратных комплексов (ПАК) на предприятии [2].

Проведенный анализ существующих аналогов программных средств (ПС) выделил их основные недостатки:

- фирмы, осуществляющие разработку и внедрение систем учёта и анализа ПАК предприятия, находятся за пределами Республики Беларусь, что затрудняет процесс внедрения ПС и обучения персонала;
- нет возможности доработки и изменения функциональности ПС отделом АСУ предприятия;
- необходимость ввода избыточной информации для корректной работы ПС;
- отсутствие необходимой функциональности ПС;
- нет возможности интеграции со сторонними приложениями.

Поэтому руководством компании, было принято решение разработать собственное ПС, которое позволит учесть специфику работы предприятия, и обеспечит выполнение следующих функций:

- хранение, анализ и обработка сведений о программно-аппаратных комплексах предприятия;
- отслеживание перемещения компьютерной техники на другое рабочее место, добавление или замена устройств/комплектующих;
- аудит программного обеспечения;
- планирование инвентаризации;
- формирование аналитических отчетов;
- синхронизация пользователей с Active Directory;
- отправка оповещений на электронную почту сотрудника;
- интеграция данных с другими ПС.

Учет компьютерной техники организации (тип устройства, технические характеристики, инвентарный и серийный номера, поставщик, стоимость, дата покупки, срок и дата окончания гарантии, ответственное лицо и др.). Учет ремонтов и модернизации, замен расходных материалов и частей, профилактических работ (дата, вид и описание произведенных работ). Учет программного обеспечения (его описание, регистрационный ключ, количества копий, стоимости). В том числе системой предусмотрен количественный и суммовый учет номенклатуры по материально-ответственным лицам. За каждым местом хранения есть возможность закрепить ответственного сотрудника и сотрудника по его обслуживанию. Присутствует механизм внешних печатных форм и отчетов. Реализован поиск по базам данных (несколько критериев, один критерий) [3].

Внедрение ПС на предприятии позволит:

- снизить временные затраты на проведение инвентаризации и оформление отчетных документов;
- оптимизировать трудозатраты персонала;
- минимизировать ошибки «человеческого фактора»;
- повысить эффективность поиска информации за счет сокращения времени обработки характеристик комплексов предприятия и получения из них оперативных данных;
- упорядочить данные;
- осуществлять мониторинг функционирования корпоративной вычислительной сети, оперативно устранять поломки и сбои в работе.

Список использованных источников.

1. Репин, В.В. Бизнес-процессы. Моделирование, внедрение, управление/ В.В. Репин, – Издательство МИФ, М.- 2012. – 512 с.
2. Теория и технология систем управления/ Ю.М. Кривин – ДМК Пресс, 2012. – 424 с.
3. Федоров, Ю.Н. Справочник инженера по АСУТП Проектирование и разработка/ Ю.Н. Федоров – Инфра-Инженерия, 2016. – 485 с.

ПРОВЕРКА ГИПОТЕЗЫ ДЛЯ ВЫБОРА ТЕМПЕРАТУРЫ ТЕРМОВЫДЕРЖКИ КОМПЛЕКТУЮЩИХ ЭВМ

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Яровенко К.А.

Сечко Г.В. – доцент каф. ИСиТ, к.т.н., доцент

Проверяется гипотеза о том, что при повышении температуры на 10°C наработка комплектующего ЭВМ до отказа уменьшается в 2 раза. С помощью теории линий Аррениуса показывается, что гипотеза справедлива только на малых величинах электрической нагрузки комплектующего.

Для отбраковки заведомо дефектных комплектующим ЭВМ (К/ЭВМ), эксплуатация которых приводит к их ранним отказам (ранними обычно именуют отказы, возникающие на начальной стадии эксплуатации технических объектов), большинство производителей применяют в технологическом процессе их изготовления операции приработки, важное место среди которых занимает термовыдержка [5] (эксплуатация изделия на этапе его изготовления в камере тепла в течение заданного времени, называемого длительностью термовыдержки, при заданной температуре (температуре термовыдержки)). Одной из важнейших технологических задач при разработке документации на операцию термовыдержки является правильный выбор её длительности. В американских стандартах длительность термовыдержки назначается в размере 24, 72 и 168 часов (сутки, трое суток, неделя). Для оптимизации длительности термовыдержки обычно используют простейший метод, который заключается в следующем. Из предложенного ряда берут какую-либо длительность (например, 72 часа), и при обнаружении недостаточной длительности термовыдержки повышают температуру на 10°C . Существует гипотеза, что при повышении температуры на 10°C наработка К/ЭВМ до отказа уменьшается в 2 раза. Например, К. Хиллман, специалист компании DFR Solutions, однозначно заявляет в [2]: «Принято считать, что срок службы увеличивается в два раза при каждом снижении температуры на 10°C ». В нашем случае это значит, что длительность термовыдержки можно уменьшить до 36 часов. Если такая длительность по-прежнему недостаточна, то температуру снова повышают на 10°C , получая длительность 18 часов и т. д. Метод справедлив, если гипотеза верна.

В настоящей работе для проверки справедливости вышеизложенной гипотезы предлагается использовать результаты работы [1], где по данным справочника [3] с помощью программного комплекса МАТКАД строятся линии Аррениуса для различных электрорадиоэлементов (ЭРЭ), в т. ч. и К/ЭВМ, и на их основе определяются энергии активации деградационных процессов при различных коэффициентах электрической нагрузки. При этом под электрической нагрузкой ЭРЭ K_H понимается отношение фактически протекающего через него тока либо фактического падения напряжения на нём к номинальным значениям указанных параметров.

Проверить справедливости вышеизложенной гипотезы с помощью результатов работ, изложенных в [1], выберем работу [4]. Автор [4] исследует безотказность преобразователей напряжения (источников вторичного электропитания, ИВЭП), большинство ЭРЭ которых составляют мощные кремниевые транзисторы. По результатам исследования в [4] получены следующие результаты: наработка ИВЭП до отказа $MTBF = 44707$ час при температуре $T = 50^{\circ}\text{C}$ и $MTBF = 16305$ час при температуре 75°C . По результатам работ, изложенных в [1], имеем следующие энергии активации деградационного процесса в ИВЭП: 0,382 эВ при $K_H = 0,1$; 0,340 эВ при $K_H = 0,3$; 0,310 эВ при $K_H = 0,5$; 0,267 эВ при $K_H = 0,7$.

Подставляя при выбранном K_H $MTBF = 5,1$ года и $T = 50^{\circ}\text{C}$, а также соответствующую выбранному K_H энергию активации в уравнение Аррениуса, пересчитанное для $MTBF$ [1], несложно найти неизвестный коэффициент этого уравнения, получив тем самым зависимость $MTBF$ от температуры. Затем, подставляя в найденную зависимость $T = 50^{\circ}\text{C}$ (для проверки справедливости гипотезы) и $T = 75^{\circ}\text{C}$ (для определения, при каком среднем K_H в [4] получен результат $MTBF = 1,86$ года), можно проверить и гипотезу, и результат, полученный в [4]. Выполненная проверка показала, что при коэффициентах нагрузки, отличных от 0, результат, полученный при проверке, значительно отличается от результатов из [3, 4]. Это отличие составляет: 53 % при $K_H = 0,3$, 1258 % при $K_H = 0,5$ и 9237 % при $K_H = 0,7$. Похоже, что эксперимент по построению автором [4] зависимости $MTBF$ от T проводился только для коэффициента электрической нагрузки, примерно равной 0,1. Результаты расчёта по гипотезе отличаются от результатов проверки тем больше, чем выше коэффициент электрической нагрузки. Тем не менее, гипотеза, несмотря на свою неточность при некоторых K_H , имеет большую популярность в технической литературе.

Список использованных источников:

1. Ранние отказы устройств вычислительной техники / Е. В. Моженкова, Г. В. Сечко, А. Н. Соловьянчик, Т. А. Черепко // Технические средства защиты информации: Тезисы докладов VIII-й белорусско-российский НТК (Браслав, 24-28 мая 2010 года). – Минск.: БГУИР, 2010. – С. 97.
2. Хиллман К. Распространенные ошибки при проектировании электроники // Новости электроники. – 2008. – № 1. – С. 17–23.
3. Надежность изделий электронной техники производственно-технического назначения и народного потребления: Справочник. М.: ВНИИэлектронстандарт, 1983. – 54 с.
4. Жданкин, В. Надёжность преобразователей напряжения и её количественная оценка / В. Жданкин // Современные технологии автоматизации. – 1997. – № 4. – С. 116–119.

НАУЧНОЕ ИЗДАНИЕ

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

(Минск, 21 апреля 2018 года)

ТЕЗИСЫ ДОКЛАДОВ
по направлению 8:

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

В авторской редакции

Ответственный за выпуск *В. И. Пачинин*
Компьютерная верстка *В. И. Пачинин*

Подписано в печать 12.04.2018. Формат 60x84 1/8. Бумага офсетная. Гарнитура «Ариал».
Отпечатано на ризографе. Усл. печ. л. 13,72. Уч.-изд. л. 17,5. Тираж 89 экз. Заказ 71.

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. Бровка, 6

