

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**

**ГОУВПО "ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ" (ДонНТУ)**

**ФАКУЛЬТЕТ
КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ**

**ИНФОРМАТИКА,
УПРАВЛЯЮЩИЕ СИСТЕМЫ,
МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ
МОДЕЛИРОВАНИЕ
(ИУСМКМ – 2017)**

**Сборник материалов VIII Международной
научно-технической конференции
в рамках
III Международного Научного форума
Донецкой Народной Республики**

25 мая 2017г.

г. Донецк, ДонНТУ – 2017

УДК 004

Информатика, управляющие системы, математическое и компьютерное моделирование в рамках III форума «Инновационные перспективы Донбасса» (ИУСМКМ – 2017): VIII Международная научно-техническая конференция, 25 мая 2017, г. Донецк: / Донец. национал. техн. ун-т; редкол. Ю.К. Орлов и др. – Донецк: ДонНТУ, 2017. – 802 с.

ISBN: 978-5-9908741-2-1

В сборнике опубликованы результаты научных исследований и технических разработок в области теоретической и прикладной математики, компьютерной и программной инженерии, автоматизированного проектирования, информационных управляющих систем и технологий, математического и компьютерного моделирования, дизайна, систем искусственного интеллекта, системного анализа и управления.

Results of scientific research and development works are published in collected papers in following fields: theoretical and applied mathematics, software engineering, computer-aided design, information management systems and technologies, mathematical and computer modeling, artificial intelligence systems, media industry and design, computer engineering, system analysis and management.

Редакционная коллегия:

Маренич К.Н., д.т.н., профессор, ректор ДонНТУ; Кожухов И. Б., д. ф-м. н., проф. каф. «Высшая математика №1» НИУ «Московский институт электронной техники» (МИЭТ); Струнилин В.Н., декан факультета компьютерных наук и технологий (ФКНТ); Аноприенко А.Я., к.т.н., проф., зав. каф. КИ; Карабчевский В.В., к.т.н., доц., зав. каф. КМД; Привалов М.В., к.т.н., доц., зав. каф. АСУ; Павлыш В.Н., д.т.н., проф., зав. каф. ПМ; Миненко А.С., д.ф-м.н., проф., зав. каф. ИИСА; Федяев О. И., к.т.н., доц. зав. каф. ПИ; Орлов Ю.К., доц. каф. ИИСА; Ручкин К.А., доц. каф. ИИСА; Григорьев А.В. к.т.н., доц. каф. ПИ.

Адрес редакционной коллегии:

83001, г. Донецк, ул. Артема, 58,

Донецкий национальный технический университет,

11 учебный корпус, факультет компьютерных наук и технологий,
кафедра искусственного интеллекта и моделирования, ком.10.408.

E-mail: iuskm@donntu.org

<http://iuskm.donntu.org>

© Донецкий национальный технический университет, 2017

СОДЕРЖАНИЕ

СЕКЦИЯ № 1 ТЕОРЕТИЧЕСКАЯ И ПРИКЛАДНАЯ МАТЕМАТИКА.....	13
Кожухов И.Б. О конгруэнциях полигонов над вполне простыми полугруппами.	14
Свентковский В.А. О построении плоских выпуклых k -угольников ($k=5.6.7.8$), которые можно разделить на n любых заданных равных прямоугольных треугольников.	18
Щучкин Н.А. Конечно порожденные полуабелевы n -группы	26
Матвеев М.О., Ефименко К.Н. Квантовые компьютеры или новый уровень обработки информации	32
Павлов А.Л. Задача кошки для простейшего уравнения соболевского типа в классе обобщенных функций медленного роста.....	38
Грановский Я.И. К спектральной теории векторнозначных операторов штурма-лиувилля с суммируемыми потенциалами и точечными взаимодействиями	42
Оридорога Л.Л. Об отрицательных квадратах функций, не являющихся радиально положительно определёнными.....	46
Лиманский Д.В. Условия подчиненности для тензорного произведения дифференциальных полиномов	49
Саксон А.В., Шевцова А.Д., Анохина И.Ю., Перинская Е.В. Сравнительный анализ CMS систем для создания сайта-визитки	52
Будыка В.С. Самосопряженность матричного оператора Дирака с точечными взаимодействиями.....	56
Прокопенко Е.В., Масло С.В., Пахомов Е.В. Применение линейного программирования для формирования техногенного месторождения	58
Андрюхин А.И, Полетаев В.А. Рефлексивная референция и анализ квайнов ...	63
Рощина Е.В., Анохина И.Ю. Жизненный цикл групп в социальных сетях	73
Кузнецов П.А., Масло С.В., Лисицин А.Н. Разработка алгоритма для системы автоматического управления охлаждением шахтного газа-метана.....	78
Черепов В.Г., Зензеров В.И. Моделирование работы гидропривода горных машин.....	83
Ткаченко Д.Н. Применение теории дифференциальных уравнений в построении полиномиальных решений уравнений Эйлера-Пуассона.....	88
Клищевский А.Р., Лазебная Л.А. Применение линейного программирования для нахождения минимального времени взаимодействия звеньев транспортной логистической цепи	92

Мазнев А.В., Белоконь Т.В. О первых интегралах уравнений Пуассона при наличии трех инвариантных соотношений в задаче о движении гиростата.....	94
Ананьева А.Ю. Экстремальные расширения неотрицательного оператора Якоби....	98
СЕКЦИЯ № 2 ПРОГРАММНАЯ ИНЖЕНЕРИЯ.....	101
Лебезова Э. М. Анализ эффективности методологий разработки программного обеспечения в условиях ограниченного времени и роста неопределённости	102
Чернышова А.В., Кубашевский Д.В. Подсистема защиты авторского права в программном обеспечении	110
Грищенко О.С., Чернышова А.В. Защита информации в приложениях для мгновенной передачи сообщений.....	115
Полетаев В.А., Кузьмичева А.С. Использование средств рефлексии языка Java при реализации протоколов прикладного уровня.	120
Лебедев В.Е., Чернышова А.В. Клиент-серверное программное обеспечение безопасного облачного хранилища файлов	125
Пикалёв Я.С., Дорохина Г.В. Анализ систем распознавания номерных знаков автомобилей.....	130
Ушаков А.И. Дорохина Г.В. Разработка клиент-серверного приложения компьютерной системы психофизиологической диагностики	135
Коломойцева И.А., Сторожук Н.О. Программная система для бисероплетения	140
Коломойцева И.А., Власюк Д.А. Программная система для перевода текста с помощью камеры мобильного устройства.....	150
Серёженко О.А., Коломойцева И.А. Анализ словарей ассоциаций в мета-поисковой системе поиска цитат	157
Ефремченко И.А., Губенко Н.Е. Особенности модели аутентификации на основе ЦВЗ для разработки систем графического пароля.....	163
Титаренко М.Г., Коломойцева И.А. Информационная система «GrabTheTrade»	167
Губенко Н.Е., Сирант А.В. Сравнительный анализ и выбор алгоритмов хеширования для организации парольной защиты игровых приложений.....	173
Медгаус С.В., Федяев О.И. Исследование программных моделей нейронных сетей прямого распространения.....	181
Ворожбитов Э.Э., Павлий В.А. Разработка клиентского приложения для шифрования и дешифрования файлов с использованием сети Фейстеля.....	188

СЕКЦИЯ № 3 СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ	193
Кухта С.С., Бобелюк М.Б., Мезенцев Д.А., Харитонов А.Ю. 3D-фрезерный станок с ЧПУ под управлением LabVIEW	194
Скаженник В.Б. Проектирование горнотехнических систем на основе компьютерного моделирования.....	198
Аврамцов В.С., Григорьев А.В. Методы создания графических баз данных в САПР.....	203
Калинина Е.С., Григорьев А.В. Онтологии в САПР, анализ состояния.....	208
Зимонин И.О., Григорьев А.В. Разработка приложения при помощи совместной работы библиотеки ObjectARX и Javascript API с платформой AutoCAD на примере САПР трубопроводов.....	218
СЕКЦИЯ № 4 ИНФОРМАЦИОННЫЕ УПРАВЛЯЮЩИЕ СИСТЕМЫ И ТЕХНОЛОГИИ	231
Бизянов Е.Е., Гутник А.А. Подсистема интеллектуального анализа технико-экономической информации.....	232
Бизянов Е.Е., Кононенко Н.Н. Архитектура интеграции разнородных источников данных	235
Чередникова О.Ю., Капралова А.Ю. Выбор системы автоматизации	239
Москалюк Т.Г, Харитонов А.Ю. Разработка программного обеспечения звукоанализирующего пульсометра в среде NI LabVIEW	243
Вудвуд Е.Ю., доц. Васяева Т.А. Анализ статистических данных для системы прогнозирования продаж	247
Раскидкин В.В., Достлев Ю.С. Интерактивная поддержка в системе анализа достоверности первичной информации системы автоматического управления раскромом проката на металлургическом предприятии	255
Лысенко В.С., Землянская С.Ю. Модель СППР для малого предприятия	260
Хороших С.Л., Савкова Е.О. Исследование оптимальных методов загрузки транспорта	266
Задорожная Е.Г., Савкова Е.О. Метаэвристические алгоритмы формирования оптимального графика прохождения лечебно-оздоровительных процедур.....	270
Гимадеев К.Д., Чередникова О.Ю. Автоматизация работы приемной комиссии учебного заведения при проведении вступительных испытаний	275
Выключок А.В., Землянская С.Ю. Разработка методики оценки компетентности сотрудников IT-отдела с помощью систем нечёткого вывода.....	281

Двойкин А. А., Привалов М. В., Воронова А. И. Архитектура веб-ориентированной системы распределённой обработки медицинских изображений	287
Чаленко К.Ю., Светличная В.А. Анализ и определение задач логистики в интернет-магазине	293
Коношенко В. О., Привалов М. В. Проектирование архитектуры компьютеризированной подсистемы управления работой службы такси	301
Слюнкин Д.В., Васяева Т.А. Прогнозирование технического состояния оборудования системы контроля и управления доступом	307
СЕКЦИЯ № 5 МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ.....	312
Дяченко О.Н., Зинченко Ю.Е., Дяченко В.О. Исследование эффективности компактного тестирования на основе методов циклического кодирования	313
Плитка А.Ю. Ручкин К.А. Исследование и разработка метода распознавания объекта на изображении с помощью алгоритма муравьиных колоний.....	319
Ганенко А.И. Анализ применения Марковских цепей при моделировании прогноза генотипа потомства.....	323
Должиков Н.В., Левшов А.В. Влияние затенения на солнечную электростанцию	328
Карабчевский В.В., Кудимов М.И. Автоматизация решения задач трехмерного моделирования как инструмент разработки учебных пособий и обучающих систем	335
Толстых В.К., Бодряга В.Е. Идентификация температуропроводности в непрерывном металлическом слитке.....	341
Чепига А.А. Расчет переходного процесса при включении электропривода в однофазной электрической цепи переменного тока	345
Орловский Д.В. Описание и выбор метода для проектирования скс для постановки предварительного диагноза по биохимическому анализу слюны.....	351
СЕКЦИЯ № 6 СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА.....	359
Ольшевский А.И., Марченко В.В. Модуль визуализации движения автотранспорта по курсу «Основы правил дорожного движения»	360
Безуглый М.А., Секирин А.И. Модифицированный генетический алгоритм составления расписания в условиях учебного заведения	365
Свиридов Д.В., Бочаров Д.М. Исследование программ и интернет-ресурсов для составления комплекса физических тренировок	371

Семенова А.П., Миненко А.С. Информационные технологии при моделировании и распознавания лица человека по его мимическим изображениям	375
Бычкова Е.В., Нестеренко В.С. Программное средство создания резервных копий данных	380
Бурлаева Е.И., Ермоленко Т.В. Анализ методов классификации текста.	384
Ковалёв Д.В., Бычкова Е.В. Система распознавания лиц при помощи систем видеонаблюдения. Алгоритм поиска лиц	392
Черкасс Д.В., Кравец Т.Н. Методы анализа социальных групп на базе их поведение в социальных сетях.....	397
Бычкова Е.В., Шумский А.А. Программное средство распознавания печатного текста.....	402
Столбунская А.С., Паламарчук И.Д., Бочаров Д.М. Автоматизация процесса обучения при помощи виртуальных лабораторий	406
Горлов Е.Д., Бочаров Д.М. Разработка информационной системы учета рабочего времени сотрудников предприятия	410
Потопахин А.А., Ручкин К.А. Анализ программных средств и методов распознавания динамических жестов	416
Нечаев А.В., Харитонов Ю.Е. Структура и функциональность классов для моделирования искусственной жизни	420
Михайлович Ф., Харитонов Ю.Е. Применение теории адаптивного резонанса в системах выдачи рекомендаций	424
Семенова А.П., Кудрявых А.Ю. Разработка программного обеспечения для просмотра и редактирования графических файлов	428
Лысюк Р.Р., Кравец Т.Н. Алгоритмы анализа социальных сетей для выявления социальных характеристик пользователей	432
Тимошков А.В. Автоматическое построение структурированной тематической аннотации для русского языка	437
Купоросов А.А., Бочаров Д.М. Программное средство для нахождения оптимального решения комбинаторных задач путём применения оптимальной комбинации генетических операторов.....	439
Шуляк Б.А., Харитонов Ю.Е. Экономико-математическое моделирование методом имитации отжига	445
Потёмкин А.С., Миненко А.С. Метод нахождения центра зрачка на изображении глаза	450
Минков О.В., Ручкин К.А. Исследование и разработка алгоритма распознавания сложных накладывающихся геометрических объектов	455

Соколов Н.Г., Ручкин К.А. Обзор существующих методов оффлайн-распознавания последовательностей рукописных символов	460
Нечепуренко М.С., Ольшевский А.И. Исследование и разработка алгоритма построения деревьев Штейнера для случая ортогональной метрики в трёхмерном пространстве	464
Потёмкин А.С., Миненко А.С. Использование компьютерного зрения для распознавания языка жестов	470
Вивденко В.С., Бочаров Д.М. Обзор распространенных игровых движков для создания 2D приложений.....	475
Ткаченко В.В., Ручкин К.А. Исследование алгоритмов обнаружения объектов на изображении	479
Яременко П.А. Применение алгоритмов программного распознавания автомобильных номеров с использованием Android-устройства.	482
Иванов А.А., Копытова О.М. Об одном алгоритме адаптивного тестирования	488
Семенова А.П., Стрюковская А.Д. Разработка программного обеспечения для загрузки транспортного средства с помощью эволюционных алгоритмов	494
Струков В.А., Сорокин Р.А. Разработка автоматизированной обучающей системы по курсу «Основы алгоритмизации и программирования»	499
Гетманова А.Д., Михневич П.Д., Ручкин К.А. Выбор программного обеспечения с открытым исходным кодом для создания Web-сайта.....	503
Сорокин Р.А., Оверченко Я.Ю. Разработка системы машинного перевода текстов в англо-русском направлении.....	507
Ручкин К.А., Егурнов Н.О. Обзор систем процедурной генерации.....	511
Лукашов С.А. Основные алгоритмы роевого интеллекта, их сравнение и сферы применения	515
Лукашов С.А. Использование онтологий для создания онлайн-словаря	520
Иванов А.А. Использование адаптивного тестирования при дистанционном обучении.....	523
Вергелес А.А., Копытова О.М. О влиянии переброски дуг на поведение ОД-к-автомата	528
СЕКЦИЯ № 7 МЕДИАИНДУСТРИЯ И ДИЗАЙН	533
Сидорчук В.И., Харитонов В.В. Проблема адаптации сайта для слабовидящих людей.....	534
Дворянин Д.М., Харитонов В.В. К вопросу о понятиях UX/UI дизайна.....	539
Тилинина Н.Ю. Адаптация дизайна для людей с нарушенным цветовосприятием	545

Тахтаров Ю.О., Харитонова В.В. Сравнительный анализ очков виртуальной реальности	550
Писарев Д.В., Харитонова В.В. Методика создания 3D-графики в game-индустрии	553
Кладько Д.С., Харитонова В.В. Особенности психо-физиологического восприятия цвета детьми дошкольного и младшего школьного возраста на примере интерфейса обучающей системы "Космо"	558
Штанько А.А., Харитонов А. Ю. Разработка структуры веб-ориентированной обучающей системы для среды LabVIEW	564
Фролов А.Ю, Грунский И.С. Исследование основных этапов внутренней оптимизации сайта	569
Павлий В.А., Воронков В.А. Разработка информационной системы управления персональными сайтами студентов с возможностью их верификации	574
Лях Е.В. Разработка структуры для web-ресурса «Интернет-магазин аксессуаров в ДНР».....	578
Кандаурова А.О., Харитонова В.В. Использование цвета в разработке айдентики бренда.....	582
Жиленкова Е.Ю., Харитонова В.В. Использование трендов web-дизайна 2017 года при разработке персонального сайта	586
Харитонова В.В. Составляющие дизайнерской компетентности будущих разработчиков медиасистем.....	591
СЕКЦИЯ № 8 КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ.....	594
Прищепа В.В., Краснокутский В.А., Меренкова Л.Л. Разработка программного модуля «Нагрузка преподавателя». Оценка эффективности использования ORM Entity Framework.	595
Юрьев Н.Н., Бельков С.Д., Суббота Н.С., Васяева Т.А. Система контроля и управления доступом.....	601
Краснокутский В.А., Серпуховитин Ю.Е. Устройство измерения скорости вращения диска испытательного стенда.....	605
Плахин С.В., Краснокутский В.А. Вопросы организации аппаратного и программного обеспечения струевых методов исследования быстрых химических реакций	610
Скиба В.Е., Краснокутский В.А., Меренкова Л.Л. Исследование и разработка оптимальных запросов в рамках решения задачи по автоматизации процесса формирования нагрузки кафедры.....	615

Койбаш А.А., Кравченко А.Г. Эффективность использования многоядерных систем для прогнозирования траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники.	622
Габибов Р.Ю., Николаенко Д.В. Исследование персональных сетей при проектировании компьютерных систем.....	626
Игнатов А.Ю., Теплинский С.В. Особенности разработки приложений для универсальной платформы Windows.....	631
Омельянченко С.Н., Мальчева Р.В. Реализация пошагового онлайн-умножителя двоичных чисел	635
Иванов В.Ю., Завадская Т.В., Меренкова Л.Л. Исследование методов формирования автоматизированного расписания.....	639
Чаган А.Н., Ярошенко Н.А. Структура системы зондового диагностирования цифровых плат.....	644
Захарченко К.С., Теплинский С.В., Дорожко Л.И. Особенности определения кодировки символов в C#.....	649
Седневцев М.А., Теплинский С.В. Особенности поиска удаленных файлов и каталогов в файловой системе NTFS.....	653
Лычагин Р. В., Аноприенко А. Я., Иваница С. В. Анализ существующих методов выравнивания изображения при решении проблемы распознавания лиц	657
Лада А.А., Мальчева Р.В. Исследование структуры графической обучающей системы на примере игры.....	662
Марченко А.С., Зинченко Ю.Е. Разработка онлайн системы тестирования знаний в области FPGA-технологий.....	665
Лушачевский В.А., Аноприенко А.Я. Модернизация алгоритмов определения рук при помощи Kinect и Leap Motion	670
Дерксен Е.А., Завадская Т.В., Меренкова Л.Л. Исследование эффективности методов оптимизации баз данных в рамках автоматизированной системы приемной комиссии ДонНТУ.....	675
Азаров А.Б., Зубков Д.А. Константинов В.С., Зинченко Ю.Е., Коваль А.В. Эмуляция периферийных устройств для удаленного доступа к FPGA-комплексам	678
Зимцев П.П., Коротач А.С., Чередникова О.Ю. Исследование возможностей SD-WAN сетей.....	683
Крахмаль М.В., Кривошеев С.В. Применение параллельных архитектур вычислительных систем для визуализации 3d-модели.....	688

Кравченко М.К., Кривошеев С.В. Применение параллельных архитектур вычислительных систем для расчета параметров движения транспортного средства.....	700
Иорданов Р.В., Завадская Т.В. Метаэвристический алгоритм летучих мышей для построения маршрута пути робота.	692
Горин Н.А., Струнилин В.Н. Разработка системы дистанционного управления для мобильного робота.	699
СЕКЦИЯ № 9 СИСТЕМНЫЙ АНАЛИЗ И УПРАВЛЕНИЕ.....	703
Degtyarev S.P. On the possible seminorms on certain weighted Holder spaces in mathematical models of the thin films.	704
Ванжа Т.В., Орлов Ю.К. Научно-технический риск в системе управления предприятием угольной промышленности.....	708
Миненко А.С., Радевич Е.В. Приближенный анализ процесса кристаллизации металла.....	712
Гуляев В.А., Орлов Ю.К. Исследование влияния работы кислородно-компрессорного цеха на деятельность металлургического завода.....	718
Гримало П.А., Орлов Ю.К. Разработка модели оценки и прогнозирования факторов травматизма в горном производстве	726
Рославцева Л.А., Орлов Ю.К. Разработка моделей системного управления процесса сушки макаронных изделий.....	732
Бровкина А.С., Орлов Ю.К. Современные медицинские информационно-справочные системы.....	738
Дробитько А.В., Тарасова И.А. Модернизация системы автоматического розлива на предприятиях пищевой промышленности.....	744
Тарасов Н.Ю., Удовика Ю.Д., Бутенко Д.А., Лёвкин А.В., Колесников В.Ю., Елисеев В.И. Создание программы для робота и реализация его функционирования.....	749
Горчаков А.В, Ольшевский А.И. Проектирование автоматизирован-ного учета движения двигателей по участкам цеха ремонта машин	752
Мартынова Д.С., Гудаев О.А. Построение систем управления документами в электронных библиотеках	758
Резникова Г.А., Тарасова И.А. Разработка системы контроля и мониторинга состояния промышленной безопасности на угледобывающих предприятиях. ..	764
Удовика Ю.Д., Бутенко Д.А., Тарасов Н.Ю., Колесников В.Ю., Лёвкин А.В., Елисеев В.И. Создание программного робота «Художника» с использованием Lego Mindstorms.	770

Цыганков С.Н., Гудаев О.А. Онтологические модели описания организации деятельности в формате JSON-LD.....	774
Милая А.С., Савкова Е.О. Применение генетического алгоритма для формирования альтернативных решений в задачах принятия решения.	779
Кацель С.Л., Кравец Т.Н. Исследование программной реализации теста Порсолта (или принудительное плавание) для установления уровня депрессивности лабораторных грызунов	784
Байбуза А.В., Орлов Ю.К. Разработка системы управления запасами	789
Небесная В.С., Ермоленко Т.В. Автоматическая генерация программного кода на основе естественно языковых спецификаций.....	797

Секция № 1
Теоретическая и
прикладная
математика



УДК 004.274

О КОНГРУЭНЦИЯХ ПОЛИГОНОВ НАД ВПОЛНЕ ПРОСТЫМИ ПОЛУГРУППАМИ

Кожухов И.Б.

Национальный исследовательский университет «МИЭТ» (Москва)

E-mail: kozuhov_i_b@mail.ru

Аннотация

Кожухов И.Б. *О конгруэнциях полигонов над вполне простыми полугруппами. Изучаются связи между конгруэнциями полигонов над вполне простыми полугруппами и полигонов над группами.*

Ключевые слова: *конгруэнция, полигон, полугруппы, строение конгруэнций полигонов.*

Полигоном над полугруппой S (см. [1]) называется множество X , на котором действует полугруппа S , т.е. определено отображение $X \times S \rightarrow X$, $(x, s) \mapsto xs$, удовлетворяющее условию $x(st) = (xs)t$ при $x \in X$, $s, t \in S$. Хорошо известно, что полигон является алгебраической моделью автомата (при этом X – множество состояний, а S – полугруппа входных сигналов).

Конгруэнции алгебраических систем несут определённую информацию, иногда весьма значительную, о строении самой системы. Полигоны над полугруппами в этом смысле не являются исключениями, и имеется довольно большое число работ, посвящённых конгруэнциям полигонов.

Вполне простой полугруппой называется полугруппа S , не имеющая идеалов, отличных от S , и содержащая хотя бы один примитивный идемпотент. Согласно классической теореме Сушкевича – Риса (см. [2, теорема 3.5]) вполне простые полугруппы – это в точности рисовские матричные полугруппы $S = M(G, I, \Lambda, P)$, где G – группа, I и Λ – множества, $P = \|p_{\lambda i}\|_{\lambda \in \Lambda, i \in I}$ ($p_{\lambda i} \in G$) – сэндвич-матрица, элементы из S имеют вид $(g)_{i\lambda}$ ($g \in G, i \in I, \lambda \in \Lambda$) и умножаются по правилу $(g)_{i\lambda} \cdot (h)_{j\mu} = (gp_{\lambda j}h)_{i\mu}$. Полигоны над вполне простыми полугруппами были описаны в [3]. Ниже будет приведено это описание.

Для любой группы G и её подгруппы H (не обязательно нормальной) обозначим через G/H множество правых смежных классов Hg . Оно

является G -полигоном, где умножение на элементы из G определено по правилу $Hg \cdot g' = Hgg'$. Полигон над группой G с единицей e называется унитарным, если $xe = x$ для любых элементов x этого полигона. Если полигон X (над любой полугруппой) является объединением своих попарно не пересекающихся подполигонов X_γ ($\gamma \in \Gamma$), то мы говорим, что X является копроизведением полигонов X_γ , и записываем это так:

$$X = \coprod_{\gamma \in \Gamma} X_\gamma.$$

Пусть X – множество, G – группа, $(H_\gamma)_{\gamma \in \Gamma}$ – семейство её подгрупп, $Q = \coprod_{\gamma \in \Gamma} (G/H_\gamma)$ – G -полигон (хорошо известно, что это общий вид любого унитарного полигона над группой), $\kappa_\lambda : Q \rightarrow X$, $\pi_i : X \rightarrow Q$ ($i \in I$, $\lambda \in \Lambda$) – отображения такие, что $q\kappa_\lambda\pi_i = q \cdot p_{\lambda i}$ при всех $q \in Q$, $i \in I$, $\lambda \in \Lambda$. Определим на множестве X действие полугруппы $S = M(G, I, \Lambda, P)$ по правилу $x \cdot (g)_{i\lambda} = (x\pi_i \cdot g)\kappa_\lambda$. Нетрудно видеть, что тогда X будет являться полигоном над полугруппой S . В [3] было доказано, что любой полигон над вполне простой полугруппой изоморфен полигону, построенному таким образом.

Пусть X – полигон над полугруппой $S = M(G, I, \Lambda, P)$. Положим $A = XS$. Нетрудно видеть, что A также является полигоном над S . Будем обозначать символом Con решётку конгруэнций. Решётка $\text{Con } A$ конгруэнций полигона A является подрешёткой решётки $\text{Con } X$ конгруэнций полигона X . Решётка $\text{Con } Q$ конгруэнций G -полигона Q имеет определённые связи с решёткой $\text{Con } A$, установление которых позволит прояснить строение решёток конгруэнций полигонов над вполне простыми полугруппами, учитывая, что решётки конгруэнций полигонов над группами являются более простыми объектами (в частности, все конгруэнции полигонов над группами (необязательно унитарных) были описаны в [4]).

Каждой конгруэнции ρ S -полигона A поставим в соответствие конгруэнцию $\tilde{\rho}$ G -полигона Q следующим образом: $\tilde{\rho}$ – конгруэнция на Q , порождённая всеми парами вида $(a\pi_i, a'\pi_i)$, где $(a, a') \in \rho$, а $i \in I$. Далее, рассмотрим такие конгруэнции $\sigma \in \text{Con } Q$, которые удовлетворяют следующему условию:

$$\forall q, q' \in Q \quad \forall \lambda \in \Lambda \quad (q, q') \in \sigma \Rightarrow (q\kappa_\lambda, q'\kappa_\lambda) \in \rho. \quad (1)$$

Теорема. $\tilde{\rho}$ – наибольшая конгруэнция полигона Q , удовлетворяющая условию (1).

Доказательство. Докажем вначале, что конгруэнция $\tilde{\rho}$ удовлетворяет условию (1). Пусть $(q_1, q_2) \in \tilde{\rho}$. Тогда имеет место цепочка равенств

$$\begin{aligned} q_1 &= u_1 g_1, \\ v_1 g_1 &= u_2 g_2, \\ &\dots\dots\dots \\ v_{n-1} g_{n-1} &= u_n g_n, \\ v_n g_n &= q_2, \end{aligned}$$

где для $j = 1, 2, \dots, n$ выполнено следующее: $g_i \in G$, $u_j = a_j \pi_{i_j}$, $v_j = a'_j \pi_{i_j}$ для некоторых $a_j, a'_j \in A$, $i_j \in I$ таких, что $(a_j, a'_j) \in \rho$. Пусть $\lambda \in \Lambda$. Умножив каждое из равенств цепочки на κ_λ , получим:

$$\begin{aligned} q_1 \kappa_\lambda &= (u_1 g_1) \kappa_\lambda, \\ (v_1 g_1) \kappa_\lambda &= (u_2 g_2) \kappa_\lambda, \\ &\dots\dots\dots \\ (v_{n-1} g_{n-1}) \kappa_\lambda &= (u_n g_n) \kappa_\lambda, \\ (v_n g_n) \kappa_\lambda &= q_2 \kappa_\lambda, \end{aligned}$$

а значит,

$$\begin{aligned} q_1 \kappa_\lambda &= a_1 \cdot (g_1)_{i_1 \lambda}, \\ a'_1 \cdot (g_1)_{i_1 \lambda} &= a_2 \cdot (g_2)_{i_2 \lambda}, \\ &\dots\dots\dots \\ a'_{n-1} \cdot (g_{n-1})_{i_{n-1} \lambda} &= a_n \cdot (g_n)_{i_n \lambda}, \\ a'_n \cdot (g_n)_{i_n \lambda} &= q_2 \kappa_\lambda. \end{aligned}$$

Следовательно, $(q_1 \kappa_\lambda, q_2 \kappa_\lambda) \in \rho$. Таким образом, $\tilde{\rho}$ удовлетворяет условию (1)

Осталось доказать, что любая конгруэнция $\sigma \in \text{Con } Q$, удовлетворяющая условию (1), содержится в $\tilde{\rho}$. Пусть $(q, q') \in \sigma$. Возьмём любое $\lambda \in \Lambda$ и положим $a = q\kappa_\lambda$, $a' = q'\kappa_\lambda$. Ввиду условия (1) $(a, a') \in \rho$. Возьмём $i \in I$. По определению конгруэнции $\tilde{\rho}$ мы имеем $(a\pi_i, a'\pi_i) \in \tilde{\rho}$. Положим $q_1 = a\pi_i$, $q_2 = a'\pi_i$. Тогда $(q_1, q_2) \in \tilde{\rho}$. Имеем: $q_1 = a\pi_i = q\kappa_\lambda\pi_i = q \cdot p_{\lambda i}$ и аналогично $q_2 = q' \cdot p_{\lambda i}$. Отсюда $q = q_1 \cdot p_{\lambda i}^{-1}$, $q' = q_2 \cdot p_{\lambda i}^{-1}$. Так как $\tilde{\rho}$ – конгруэнция, то $(q, q') \in \tilde{\rho}$. Тем самым доказано, что $\sigma \subseteq \tilde{\rho}$. Это означает, что $\tilde{\rho}$ – наибольшая конгруэнция, удовлетворяющая (*).

Замечание. Из доказательства теоремы видно, что если $\sigma \in \text{Con } Q$ такова, что

$$(q, q') \in \sigma \Rightarrow (q\kappa_\lambda, q'\kappa_\lambda) \in \rho \text{ при некотором } \lambda \in \Lambda,$$

то

$$(q, q') \in \sigma \Rightarrow (q\kappa_\lambda, q'\kappa_\lambda) \in \rho \text{ при всех } \lambda \in \Lambda.$$

Выводы. Выявленные связи между конгруэнциями полигонов A и Q поможет в дальнейшем получить ещё большую информацию о строении конгруэнций полигонов над вполне простыми полугруппами и исследовать полигоны с условиями на их решётки конгруэнций.

Список литературы

1. Kilp, M., Knauer, U., Mikhalev, A.V. Acts, monoids and categories. W. de Gruyter, N.Y.—Berlin, 2000.
2. Клиффорд, А., Престон, Г. Алгебраическая теория полугрупп, т. 1. М., «Мир», 1972.
3. Avdeyev, A.Yu., Kozhukhov, I.B. Acts over completely 0-simple semigroups. Acta Cybernetica, 2000, v. 14, № 4, p. 523-531.
4. Халиуллина, А.Р. Конгруэнции полигонов над группами. *Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика*, **13:4(2)** (2013), с. 133–137.

УДК 514.112.4

О ПОСТРОЕНИИ ПЛОСКИХ ВЫПУКЛЫХ k -УГОЛЬНИКОВ ($k = 5, 6, 7, 8$), КОТОРЫЕ МОЖНО РАЗДЕЛИТЬ НА n ЛЮБЫХ ЗАДАННЫХ РАВНЫХ ПРЯМОУГОЛЬНЫХ ТРЕУГОЛЬНИКОВ

Свентковский В.А.

Московский автомобильно-дорожный государственный технический университет (МАДИ)
g-mail: vladimirsventkovskiy @gmail

Аннотация


Свентковский В.А. О построении плоских выпуклых k -угольников ($k=5,6,7,8$), которые можно разделить на n любых заданных равных прямоугольных треугольников. Проводятся доказательства неизвестного факта комбинаторной геометрии.

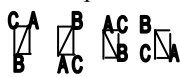
Ключевые слова: выпуклые k -угольники, комбинаторная геометрия.

Введение. Работа посвящена доказательству неизвестного факта комбинаторной геометрии, а именно: что из любого числа n , начиная с некоторого n_0 , из n произвольных одинаковых прямоугольных треугольников, можно сложить выпуклый k -угольник, при $k=5,6,7,8$. Методом построения этих k -угольников выбран метод отрезания треугольников – уголков, подобных исходному треугольнику, от некоторого прямоугольника со сторонами, кратными катетам исходного прямоугольного треугольника.

Теорема. Пусть ABC – произвольный прямоугольный треугольник. Тогда существуют натуральные числа k_5, k_6, k_7, k_8 , такие, что при любом $n \geq k_i$, $i=5,6,7,8$, существует выпуклый i -угольник, составленный из n треугольников, равных треугольнику ABC .




Доказательство. В дальнейшем условимся считать, что задан произвольный прямоугольный треугольник ABC с катетами $AC = p$, $BC = q$. Причем катет $AC = p$ будем всегда располагать горизонтально.

Тогда катет BC - всегда расположится вертикально. . Возможны четыре

расположения треугольника ABC : 

Алгоритм 1. В общем виде алгоритм построения многоугольника заключается в отрезании одного, двух, трех или четырех прямоугольных

треугольников-уголков, подобных треугольнику ABC с целым коэффициентом подобия, (уголков, составленных из треугольников, равных треугольнику ABC) от некоторого прямоугольника со сторонами, кратными катетам треугольника ABC , длины Mp – по горизонтали, длины Nq – по вертикали. Для краткости такой прямоугольник будем обозначать (M,N) . Пусть коэффициент подобия уголка равен x . Тогда уголок состоит из x^2 треугольников ABC . Соответствующий уголок обозначим (x, x) . Уголок может отрезаться, а может не отрезаться от прямоугольника (M, N) . Если уголок не отрезается, то условимся считать $x = 0$. Пусть коэффициенты подобия уголков обозначены так: a – для левого верхнего, b – для правого верхнего, c – для правого нижнего, d – для левого нижнего угла прямоугольника (M, N) . Тогда фигура, полученная отрезанием уголков состоит из $n = 2MN - a^2 - b^2 - c^2 - d^2$ треугольников ABC . Эта фигура определяется шестеркой чисел, то есть вектором: $\vec{F} = (M, N, a, b, c, d)$. Задавая вектор \vec{F} с конкретными значениями M, N, a, b, c, d , мы задаем и алгоритм построения фигуры (k -угольника), состоящей из $n = 2MN - a^2 - b^2 - c^2 - d^2$ треугольников ABC . Для построения фигуры необходимо выполнение условия, чтобы уголки не налезали друг на друга, и чтобы фигура соответствовала представленной геометрической форме. Для заданного вида n , доказательством того, что можно построить k -угольник, составленный из n треугольников, равных заданному треугольнику ABC , будем считать предъявление алгоритма, представленного в форме конкретного вектора $\vec{F} = (M, N, a, b, c, d)$ и проверку соответствия фигуры его геометрической форме, то есть проверку соответствующих неравенств. Проверка неравенств осуществляется элементарно, в уме и зрительно.

Пятиугольники. Используем такие три геометрические формы: $\phi 1$ , $\phi 2$ , $\phi 3$ . Алгоритм построения семиугольника с проверкой неравенств, формируем для каждого вида n отдельно: $n = 4K - 1$, $n = 4K - 2$, $n = 4K - 3$, $n = 4K$. Для формы 1 проверяем, что $M, N > a > 0$, для формы 2, что $M = N > a = c > 0$, для формы 3 проверяем, что $M, N > a, b > 0, a + b = M, c = d = 0$,
1). $n = 4k - 1 = 2 \cdot 2 \cdot K - 1^2 - 0^2 - 0^2 - 0^2 = 2MN - a^2 - b^2 - c^2 - d^2$.
 $\Rightarrow M = 2, N = K, a = 1$,

$b = c = d = 0$. $\vec{F} = (M, N, a, b, c, d) = (2, N, 1, 0, 0, 0)$. смотри ф.1. $N \geq 2$;
 $n_{\min} = 2 \cdot 2 \cdot 2 - 1 = 7$.

2) $n = 4K - 2 \equiv 2 \cdot 2 \cdot K - 1^2 - 1^2 - 0^2 - 0^2$. Отсюда
 $M = 2, N = K, a = b = 1, c = d = 0$

$\vec{F} = (2, K, 1, 1, 0, 0)$. Смотри ф.3. $K \geq 2$; $n_{\min} = 5$.

3).

$n = 4K - 3 = 2K^2 - 2(K - 1)^2 - 1 = 2KK - (K - 1)^2 - 0^2 - (K - 1)^2 - 1^2$
 $\vec{F} = (K, K, K - 1, 0, K - 1, 1)$. смотри ф.2. $K \geq 2$. $n_{\min} = 5$.

4) $n = 4K$ распадается на 2 случая: $n = 8K$ и $n = 8K + 4$.

4.1) $n = 8K \equiv 2 \cdot 4 \cdot (K + 1) - 2^2 - 2^2$. Отсюда

$M = 4, N = K + 1, a = b = 2, c = d = 0$,

$\vec{F} = (4, K + 1, 2, 2, 0, 0)$, смотри ф.3. $N = K + 1 \geq 3, K \geq 2$; $n_{\min} = 16$;

4.2) $n = 8K + 4 = 2(K + 2)(K + 2) - K^2 - 0^2 - K^2 - 2^2$. Отсюда






$M = N = K + 2, a = K$,

$b = 0, c = K, d = 2$. $\vec{F} = (K + 2, K + 2, K, 0, K, 2)$. смотри ф.2.

$N = K + 2 \geq 3, K \geq 1$. $n_{\min} = 12$.

Итак доказано, что при $n \geq 16$, пятиугольник строится.

Шестиугольники

Геометрические формы: Ф.4 , Ф.5 , Ф.6 , Ф.7 , Ф.8 .

Алгоритм 1 построения шестиугольника с заданным n , формируем для каждого вида n отдельно, в зависимости от остатка, при делении числа n на 4. Проверяем неравенства. Для формы 4:

$c = 0, M, N > a, b, d > 0, a + d < N, b + c < M$ Для ф. 5:

$a = c = 0, M, N > b, d > 0$,

Для ф.6: $c = d = 0, a, b > 0, M > a + b, N > a, b > 0$. Для ф. 7:

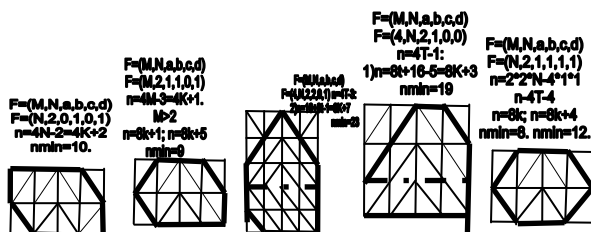
$c = 0, a, b, d > 0, M = b + c$,

$N > a + d, b > 0$. Для ф. 8:

$a, b, c, d > 0, M > a + b, c + d, N = a + d = b + c$.



Алгоритм описан, и кратко проиллюстрирован на нижеследующих рисунках, Из рисунков ясно, от каких прямоугольников и площадей

отрезаются уголки и сколько штук треугольников в уголках, прямоугольнике и пятиугольнике.



Итак, доказано, что шестиугольники строятся при любом $n. > 22$.

Семиугольники:

Используем такие две геометрические формы: ф.9 , ф.10 .

Алгоритм строим в зависимости от последней цифры числа n , задаем вектором $\vec{F} = (M, N, a, b, c, d)$ и проверкой неравенств, соответствующих геометрической форме. Проверяем неравенства: Для ф.9:
 $a, b, c, d > 0, a + b < M, a + d, b + c < N$. Для ф.10:
 $a, b, c > 0, d = 0, a + b, c < M, a, b + c < N$.

Алгоритм построения семиугольника с проверкой неравенств формируем для каждого вида n отдельно: $n = 10k + 1, n = 10k + 2, n = 10k + 3, n = 10k + 4, n = 10k + 5, n = 10k + 6, n = 10k + 7, n = 10k + 8, n = 10k + 9, n = 10k + 0$.

$$1) n = 10K + 1 \equiv 10(K + 2) - 19 \equiv 2 \cdot 5 \cdot (K + 2) - 1^2 - 1^2 - 1^2 - 4^2. \Rightarrow M = 5, N = K + 2,$$

$$a = b = c = 1, d = 4. \vec{F} = (5, K + 2, 1, 1, 1, 4). \text{ Смотри ф.9. } n = 10N - 19, N = K + 2 \geq 6, K \geq 4, n_{min} = 41.$$

$$2) n = 10K + 2 \equiv 10(K + 2) - 18 \equiv 2 \cdot 5 \cdot (K + 2) - 1^2 - 2^2 - 2^2 - 3^2. \text{ Отсюда}$$

$$M = 5, N = K + 2, a = 1, b = c = 2, d = 3. \vec{F} = (5, N, 1, 2, 2, 3).$$

Смотри ф.9

$$n = 10N - 18. N = K + 2 \geq 5, K \geq 3, n_{min} = 32.$$

$$3) n = 10K + 3 \equiv 10(K + 3) - 27 \equiv 2 \cdot 5 \cdot (K + 3) - 1^2 - 3^2 - 1^2 - 4^2.$$

Отсюда

$$M = 5, N = K + 3, a = 1, b = 3, c = 1, d = 4. \vec{F} = (5, K + 3, 1, 3, 1, 4)$$

Смотри ф.9. $n = 10N - 27, N = K + 3 \geq 6, K \geq 3; n_{\min} = 33.$

$$4) \quad n = 10K + 4 = 10 \cdot (K + 3) - 26 = 2 \cdot 5 \cdot (K + 3) - 1^2 - 3^2 - 4^2 - 0^2.$$

Отсюда $M = 5, N = K + 3, a = 1, b = 3, c = 4, d = 0.$

$$\vec{F} = (5, K + 3, 1, 3, 4, 0). \text{ Смотри ф.10.}$$

$$n = 10N - 26, N = K + 3 \geq 8, K \geq 5; n_{\min} = 54.$$

$$5) \quad n = 10K + 5 = 10 \cdot (K + 2) - 15 = 2 \cdot 5 \cdot (K + 2) - 1^2 - 1^2 - 2^2 - 3^2.$$

Отсюда

$$M = 5, N = K + 2, a = 1, b = 1, c = 2, d = 3. \vec{F} = (5, K + 2, 1, 1, 2, 3).$$

Смотри ф.9. $n = 10N - 15, N = K + 2 \geq 5, K \geq 3; n_{\min} = 35.$

$$6) \quad n = 10K + 6 = 10 \cdot (K + 2) - 14 = 2 \cdot 5 \cdot (K + 2) - 1^2 - 3^2 - 2^2 - 0^2.$$

Отсюда

$$M = 5, N = K + 2, a = 1, b = 3, c = 2, d = 0$$

$$\vec{F} = (5, K + 2, 1, 3, 2, 0). \text{ Смотри ф.10.}$$

$$n = 10N - 14, N = K + 2 \geq 6, K \geq 4; n_{\min} = 46.$$

$$7) \quad n = 10K + 7 = 10 \cdot (K + 3) - 23 = 2 \cdot 5 \cdot (K + 3) - 1^2 - 3^2 - 2^2 - 3^2.$$

Отсюда

$$M = 5, N = K + 3, a = 1, b = 3, c = 2, d = 3, \vec{F} = (5, K + 3, 1, 3, 2, 3)$$

. Смотри ф.9.

$$n = 10N - 23, N = K + 3 \geq 6, K \geq 3; n_{\min} = 37.$$

$$8) \quad n = 10K + 8 = 10 \cdot (K + 3) - 22 = 2 \cdot 5 \cdot (K + 3) - 1^2 - 2^2 - 1^2 - 4^2$$

Отсюда

$$M = 5, N = K + 3, a = 1, b = 2, c = 1, d = 4, \vec{F} = (5, K + 3, 1, 2, 1, 4)$$

. Смотри ф.9.

$$n = 10N - 22, N = K + 3 \geq 6, K \geq 3; n_{\min} = 38.$$

$$9) \quad n = 10K + 9 = 10 \cdot (K + 2) - 11 = 2 \cdot 5 \cdot (K + 2) - 1^2 - 3^2 - 1^2 - 0^2$$

Отсюда

$$M = 5, N = K + 2, a = 1, b = 3, c = 1, d = 0, \vec{F} = (5, K + 2, 1, 3, 1, 0)$$

. Смотри ф.10.

$$n = 10N - 11, N = K + 3 \geq 5, K \geq 2 \quad n_{\min} = 29.$$

10) $n = 10K$. Возможны 2 случая:

10.1) $K = 2k$ -четное число. $n = 20k$. $n = 20k = 20 \cdot (k + 3) - 60 =$

$$2 \cdot 10 \cdot (k + 3) - 1^2 - 1^2 - 3^2 - 7^2. \quad \text{Отсюда} \quad M = 10, N = k + 3,$$

$$a = 1, b = 1, c = 3, d = 7$$

$$\vec{F} = (10, K + 3, 1, 1, 3, 7). \quad \text{См.} \quad \text{ф.9.} \quad n = 20k.$$

$$N = k + 3 \geq 9, k \geq 6; n_{\min} = 120.$$

10.2) $K = 2k + 1$ -нечетное число. $n = 20k + 10$.

$$n = 20k + 10 = 20(k + 3) - 50 = 2 \cdot 10(k + 3) - 3^2 - 4^2 - 5^2 - 0^2. \quad \text{Отсюда}$$

$$M = 10, N = k + 3, \quad a = 3, b = 4, c = 5, d = 0,$$

$$\vec{F} = (10, k + 3, 3, 4, 5, 0). \quad \text{Смотри} \quad \text{ф.10.} \quad n = 20k + 10.$$

$$N = k + 3 \geq 10, k \geq 7, \quad n_{\min} = 150. \quad \text{Поэтому, при } n = 10K \geq 150$$

семиугольник строится. Итак, доказано, что семиугольники строятся при любом $n > 149$.

Восьмиугольники.

Здесь используем только одну геометрическую форму: F11  :

Алгоритм строим в зависимости от последней цифры числа n для каждого вида n , задаем вектором $\vec{F} = (M, N, a, b, c, d)$ и проверкой неравенств, соответствующих геометрической форме: $a, b, c, d > 0$; $a + b, c + d < M$; $a + d, b + c < N$.

1) $n = 10K + 1$ Возможны 2 случая.

1.1) $n = 20k + 1; \quad K$ -четно.

$$n = 20k + 1 = 20(k + 1) - 19 = 2 \cdot 10 \cdot (k + 1) - 4^2 - 1^2 - 1^2 - 1^2. \Rightarrow \Rightarrow$$

$$M = 10, N = k + 1, a = 4, b = c = d = 1, \vec{F} = (10, k + 1, 4, 1, 1, 1).$$

$$N = k + 1 \geq 6, k \geq 5; n_{\min} = 101.$$

1.2) $n = 20k + 11; K$ -нечетно.

$$n = 20k + 11 = 20(k + 3) - 49 = 2 \cdot 10 \cdot (k + 3) - 4^2 - 4^2 - 4^2 - 1^2. \Rightarrow$$

$$M = 10, N = k + 3, \quad a = 4, b = 1, c = 1, d = 1. \quad \vec{F} = (10, N, 4, 4, 4, 1)$$

$$\Rightarrow n = 20N - 49. N = k + 3 \geq 9, k \geq 6; n_{\min} = 131.$$

$$2) n = 10K + 2 = 10(K + 2) - 18 = 2 \cdot 5(K + 2) - 2^2 - 2^2 - 1^2 - 3^2. \Rightarrow$$

$$M = 5, N = K + 2, a = b = 2, c = 1, d = 3. \vec{F} = (5, K + 2, 2, 2, 1, 3)$$

$$\Rightarrow n = 10N - 18. N = K + 2 \geq 6, K \geq 4, n_{\min} = 42$$

$$3) n = 10K + 3 = 10(K + 1) - 7 = 2 \cdot 5(K + 1) - 1^2 - 1^2 - 1^2 - 2^2. \quad \text{Отсюда}$$

$$M = 5, N = K + 1,$$

$$a = b = c = 1, d = 2. \vec{F} = (5, K + 1, 1, 1, 1, 2)$$

$$\Rightarrow n = 10N - 7, N = K + 1 \geq 4, K \geq 3, n_{\min} = 33.$$

$$4) n = 10K + 4 = 10(K + 2) - 16 = 2 \cdot 5(K + 2) - 2^2 - 2^2 - 2^2 - 2^2.$$

$$\text{Отсюда } M = 5, N = k + 2,$$

$$a = b = c = d = 2.$$

$$\vec{F} = (5, N, 2, 2, 2, 2)$$

$$\Rightarrow n = 10N - 16, N = K + 2 \geq 5, K \geq 3, n_{\min} = 34.$$

$$5) n = 10K + 5 = 10(K + 2) - 15 = 2 \cdot 5 \cdot (K + 2) - 1^2 - 3^2 - 1^2 - 2^2.$$

$$\text{Отсюда } M = 5, N = k + 2,$$

$$a = 1, b = 3, c = 1, d = 2.$$

$$\vec{F} = (5, N, 1, 3, 1, 2) \Rightarrow n = 10N - 15.$$

$$N = K + 2 \geq 5, K \geq 3, n_{\min} = 35.$$

$$6) n = 10K + 6 = 10(K + 1) - 1^2 - 1^2 - 1^2 - 1^2. \quad \text{Отсюда } M = 5,$$

$$N = K + 1, a = b = c = d = 1.$$

$$\vec{F} = (5, N, 1, 1, 1, 1) \Rightarrow n = 10N - 4 = 10N - 4. \quad N = K + 1 \geq 3, K \geq 2,$$

$$n_{\min} = 26.$$

$$7) n = 10K + 7 = 10(K + 2) - 13 = 10(K + 2) - 1^2 - 2^2 - 2^2 - 2^2. \quad \text{Отсюда}$$

$$M = 5, N = k + 2,$$

$$a = 1, b = c = d = 2. \vec{F} = (5, N, 1, 2, 2, 2) \Rightarrow n = 10N - 13.$$

$$N = K + 2 \geq 5, K \geq 3, n_{\min} = 37.$$

$$8) n = 10K + 8 = 10(K + 2) - 12 = 2 \cdot 5(K + 2) - 1^2 - 3^2 - 1^2 - 1^2.$$

$$\text{Отсюда } M = 5, N = k + 2,$$

$$a = 1, b = 3, c = d = 1. \vec{F} = (5, N, 1, 3, 1, 1) \Rightarrow n = 10N - 12.$$

$$N = k + 2 \geq 5, k \geq 3, n_{\min} = 38.$$

$$9) n = 10K + 9.$$

$$9.1) \quad K \text{ -четно.} \quad K = 2k,$$

$$n = 20k + 9 = 20(k + 3) - 51 = 2 \cdot 10(k + 3) - 1^2 - 3^2 - 4^2 - 5^2,$$

$$\text{Отсюда } M = 10, N = k + 3, a = 1, b = 3, c = 4, d = 5.$$

$$\vec{F} = (10, N, 1, 3, 4, 5) \Rightarrow n = 20N - 51. \quad N = k + 3 \geq 10, k \geq 7,$$

$$n_{\min} = 149.$$

$$9.2) \quad K \text{ -нечетно.} \quad K = 2k + 1.$$

$$n = 20k + 19 = 20N - 61 = 2 \cdot 10(k + 4) - 4^2 - 4^2 - 5^2 - 2^2, \quad \text{Отсюда}$$

$$M = 10, N = k + 4, a = b = 4, c = 5, d = 2. \vec{F} = (10, N, 4, 4, 5, 2)$$

$$\Rightarrow n = 20N - 61, N = k + 4 \geq 10, k \geq 6, n_{\min} = 139.$$

$$10) n = 10K = 10(K + 1) - 1^2 - 2^2 - 1^2 - 2^2. \text{ Отсюда } M = 5, N = k + 1,$$

$$a = 1, b = 2, \vec{F} = (5, K + 1, 1, 2, 1, 2) \Rightarrow n = 10N - 10. \quad N = k + 1 \geq 4,$$

$$k \geq 3, n_{\min} = 30.$$

Итак, доказано, что восьмиугольники строятся при любом $n > 148$. Имеем: $k_5 = 16, k_6 = 23, k_7 = 150, k_8 = 149$. Поэтому $n_0 = 150$. Теорема доказана.

Выводы. В статье обнаружено, что любое натуральное число, большее 149 имеет естественный геометрический смысл, а именно, – это число представляется четырьмя геометрическими фигурами: пятиугольником, шестиугольником, семиугольником и восьмиугольником. Алгебраическая форма этих фигур $n = 2MN - a^2 - b^2 - c^2 - d^2$ с целыми M, N, a, b, c, d . Естественно ожидать, что теорема может найти применение в комбинаторной геометрии и теории чисел, а также в изготовлении следующей развивающей игры для детей и взрослых: “Из n заданных прямоугольных треугольников сложить пятиугольник, или шестиугольник, или семиугольник, или восьмиугольник”.

Список литературы

1. К. Чандрасекхаран. Введение в аналитическую теорию чисел.
2. Математика и ее приложения. Выпуск 1. Санкт Петербург. 2008г. ‘Некоторые задачи и проблемы комбинаторной геометрии’. (Свентковский В. А.)
- 3.. Ю.И. Манин. Введение в современную теорию чисел .Москва. Издат-во. МЦНМО2013

УДК 512.572

КОНЕЧНО ПОРОЖДЕННЫЕ ПОЛУАБЕЛЕВЫ n -ГРУППЫ

Щучкин Н.А.

Волгоградский государственный социально-педагогический университет
(Волгоград)

E-mail: nikolaj_shchuchkin@mail.ru

Аннотация

Щучкин Н.А. *Конечно порожденные полуабелевы n -группы. Изучаются связи между абелевыми группами и полуабелевыми n -группами. Приведен признак разложимости конечно порожденной полуабелевой n -группы.*

Ключевые слова: *полуабелевыми n -группы, разложимость, денормализация, индекс.*

Если на множестве G задана n -арная операция f ($n \geq 2$), которая удовлетворяет обобщенному закону ассоциативности

$$\begin{aligned} & f(f(a_1, \dots, a_n), a_{n+1}, \dots, a_{2n-1}) = \\ & = f(a_1, \dots, a_i, f(a_{i+1}, \dots, a_{i+n}), a_{i+n+1}, \dots, a_{2n-1}) \end{aligned} \quad (1)$$

для всех $i=1, \dots, n-1$, и для любых $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n, b$ из G однозначно разрешимо каждое из уравнений $f(a_1, \dots, a_{i-1}, x_j, a_{i+1}, \dots, a_n) = b$ для всех $j=1, \dots, n$, то так полученную алгебру $\langle G, f \rangle$ называют n -группой [1], [2]. При $n=2$ получим обычную бинарную группу.

Коммутативность (или перестановочность элементов) в теории n -групп имеет несколько обобщений групповой коммутативности. Отметим два из таких обобщений.

n -Группа $\langle G, f \rangle$ называется *полуабелевой*, если в ней при действии n -арной операции f можно переставлять крайние элементы, т.е. верно тождество

$$f(x_1, x_2, \dots, x_{n-1}, x_n) = f(x_n, x_2, \dots, x_{n-1}, x_1).$$

Если же в n -группе $\langle G, f \rangle$ при действии n -арной операции f можно переставлять любые элементы, т.е. верно тождество

$$f(x_1, \dots, x_n) = f(x_{\sigma(1)}, \dots, x_{\sigma(n)})$$

для любой подстановки $\sigma \in S_n$, то ее называют *абелевой*. Ясно, что любая абелева n -группа является полуабелевой, обратно неверно.

Имеется тесная связь между группами и n -группами. Отметим частный случай основных результатов работ [3], [4] для полуабелевых n -групп. На любой полуабелевой n -группе $\langle G, f \rangle$ можно определить абелеву группу $\langle G, + \rangle$, где сложение действует по правилу $a + b = f(a, \underbrace{c, \dots, c}_{n-3}, \bar{c}, b)$ для фиксированного элемента c из G . Тогда для элемента $d = f(\underbrace{c, \dots, c}_n)$ и отображения $\varphi(x) = f(c, x, \underbrace{c, \dots, c}_{n-3}, \bar{c})$, которое является автоморфизмом группы $\langle G, + \rangle$, верны тождества

$$\varphi(d) = d, \quad \varphi^{n-1}(x) = x \text{ для любого } x \in G, \quad (2)$$

$$f(a_1, \dots, a_n) = a_1 + \varphi(a_2) + \dots + \varphi^{n-2}(a_{n-1}) + a_n + d. \quad (3)$$

Элемент c является нулем в группе $\langle G, + \rangle$. Группу $\langle G, + \rangle$ обозначают $ret_c \langle G, f \rangle$ и называют ретрактом n -группы $\langle G, f \rangle$. Известно [3], [4], что любые два ретракта одной и той же n -группы изоморфны. Отметим, что для любой абелевой n -группы автоморфизм φ будет тождественным.

Верно и обратное: в любой абелевой группе $\langle G, + \rangle$ для выбранных автоморфизма φ и элемента d с условиями (2) задается полуабелева n -группа $\langle G, f \rangle$, где f действует по правилу (3). В этом случае n -группу $\langle G, f \rangle$ называют (φ, d) -определенной на группе $\langle G, + \rangle$ и обозначают $der_{\varphi, d} \langle G, + \rangle$. Заметим (см. [5]), что n -группа $\langle G, f \rangle$, (φ, d) -определенная на абелевой группе $\langle G, + \rangle$, является абелевой тогда и только тогда, когда $\varphi = I_G$ – тождественный автоморфизм. n -Группа $\langle G, f \rangle$, $(I_G, 0)$ -определенная на группе $\langle G, + \rangle$, где 0 – нуль группы $\langle G, + \rangle$, называется *производной* от этой группы.

Пусть $\langle G, f \rangle = \text{der}_{\varphi, d} \langle G, + \rangle$ – полуабелева n -группа. Для каждого автоморфизма φ' группы $\langle G, + \rangle$, сопряженного автоморфизму φ , на группе $\langle G, + \rangle$ рассмотрим эндоморфизм $\mu_{\varphi'}(x) = x + \varphi'(x) + \dots + \varphi'^{n-2}(x)$. Обозначим через $\text{Im } \mu_{\varphi'}$ образ этого эндоморфизма. Пусть φ' получен из φ сопряжением с помощью автоморфизма θ , т.е. $\varphi' = \theta \circ \varphi \circ \theta^{-1}$. Для каждого автоморфизма θ имеем смежный класс $\theta(d) + \text{Im } \mu_{\varphi'}$ по подгруппе $\text{Im } \mu_{\varphi}$.

Набор

$$\{ \theta(d) + \text{Im } \mu_{\varphi'} \mid \theta \in \text{Aut} \langle G, + \rangle \} \quad (9)$$

всех таких смежных классов назовем определяющим набором множеств для n -группы $\langle G, f \rangle$.

Теорема 1. (Теорема 3, [6]) *Для абелевой группы $\langle G, + \rangle$ полуабелевы n -группы $\langle G, f \rangle = \text{der}_{\varphi, d} \langle G, + \rangle$ и $\langle G, f' \rangle = \text{der}_{\psi, q} \langle G, + \rangle$ изоморфны тогда и только тогда, когда автоморфизмы φ и ψ сопряжены в группе автоморфизмов группы $\langle G, + \rangle$ и определяющие наборы множеств этих n -групп совпадают с точностью до перестановки.*

Как и в группах, назовем n -группу разложимой, если она изоморфна прямому произведению нескольких n -групп. В противном случае n -группу называют неразложимой.

В теории конечно порожденных абелевых групп неразложимыми являются бесконечные и конечные примарные группы, которые служат компонентами прямого разложения конечно порожденной абелевой группы.

Для конечно порожденных полуабелевых n -групп ситуация иная. n -Арным аналогом циклической группы служит n -группа, ретракт которой является циклической группой, такую n -группу называют полуциклической [7]. Отметим, что циклическая n -группа (т.е. n -группа, порожденная одним элементом) является полуциклической. Известно (Предложение 14, [8]), что конечная полуциклическая n -группа изоморфна прямому произведению примарных неразложимых n -групп (n -арный аналог соответствующей теоремы из теории конечных абелевых групп). Следующая доказанная теорема также является аналогом теоремы из теории абелевых групп о неразложимости бесконечной циклической группы.

Теорема 2. *Бесконечная полуциклическая n -группа неразложима.*

Среди неразложимых полуабелевых n -групп, кроме бесконечных и конечных примарных полуциклических, имеются другие неразложимые n -группы. В [6] приведен пример конечной неразложимой полуабелевой тернарной ($n = 3$) группы, которая не является полуциклической. Здесь мы укажем пример для бесконечного случая.

Рассмотрим тернарную группу $\langle G, f \rangle$, (φ, d) -определенную на прямой суммы $\langle Z, + \rangle \oplus \langle Z_2, + \rangle$, где $\langle Z_2, + \rangle$ — аддитивная группа кольца классов вычетов по модулю 2, автоморфизм φ задан по правилу $\varphi((x, y)) = (x, x+y)$, $d = (2, 1)$. Тернарная группа $\langle G, f \rangle$ является неразложимой. Какие же конечно порожденные полуабелевы n -группы будут неразложимыми? На этот вопрос мы ответим в следующей теореме. Автоморфизм ψ конечно порожденной абелевой группы $\langle G, + \rangle$ назовем разложимым, если группу $\langle G, + \rangle$ можно представить в виде прямой суммы $\langle G, + \rangle = \langle G_1, + \rangle \oplus \dots \oplus \langle G_k, + \rangle$ Своих собственных подгрупп $\langle G_i, + \rangle$ (не обязательно циклических) так, чтобы ограничение ψ_{G_i} этого автоморфизма на каждую подгруппу $\langle G_i, + \rangle$ было бы автоморфизмом этой подгруппы. В противном случае назовем такой автоморфизм неразложимым.

Теорема 3. *Конечно порожденная полуабелева n -группа $\langle G, f \rangle$ является разложимой тогда и только тогда, когда автоморфизм φ ретракта $\langle G, + \rangle = \text{ret}_c \langle G, f \rangle$ этой n -группы, заданный по правилу*

$$\varphi(x) = f(c, x, c, \bar{c})^{(n-3)},$$
сопряжен (в группе автоморфизмов этого ретракта) некоторому разложимому автоморфизму.

Следствие 1. *Неразложимыми конечными полуабелевыми n -группами могут быть только примарные.*

Следствие 2. *Неразложимыми конечно порожденными абелевыми n -группами могут быть только бесконечные полуциклические либо конечные примарные полуциклические.*

Доказана следующая

Теорема 4. *Конечно порожденная полуабелева n -группа изоморфна прямому произведению неразложимых конечно порожденных полуабелевых n -групп, частью бесконечных, частью конечных примарных.*

Следствие 3. *Конечная полуабелева n -группа изоморфна прямому произведению неразложимых конечных примарных полуабелевых n -групп, среди которых могут быть полуциклические.*

Аналогом теоремы о разложимости конечно порожденной абелевой группы для n -групп является

Следствие 4. *Конечно порожденная абелева n -группа изоморфна прямому произведению абелевых полуциклических n -групп, частью бесконечных, частью конечных примарных.*

Также n -арным аналогом теоремы о разложимости конечной абелевой группы является

Следствие 5. *(Теорема 10, [9]) Конечная абелева n -группа изоморфна прямому произведению примарных абелевых полуциклических n -групп.*

В работе [10] изучены свободные n -группы в классе полуабелевых n -групп. Рассмотрим строение свободной конечно порожденной полуабелевой n -группы. Для каждого элемента x_i из конечного множества $\{x_1, \dots, x_k\}$ определим прямую сумму $\langle A_i, + \rangle = \sum_{j=1}^{n-1} \langle (x_{ij}), + \rangle$ бесконечных циклических групп $\langle (x_{ij}), + \rangle$. Рассмотрим прямую сумму $\langle F, + \rangle = \langle (a), + \rangle + \sum_{i=1}^k \langle A_i, + \rangle$, где $\langle (a), + \rangle$ — бесконечная циклическая группа. На каждой группе $\langle A_i, + \rangle$ выбираем автоморфизм φ_i , действующий по правилу: для любого $t_1x_{i1} + t_2x_{i2} + \dots + t_{n-1}x_{in-1} \in A_i$ имеем

$$\varphi_i(t_1x_{i1} + t_2x_{i2} + \dots + t_{n-1}x_{in-1}) = t_{n-1}x_{i1} + t_1x_{i2} + \dots + t_{n-2}x_{in-1}.$$

Тогда на группе $\langle F, + \rangle$ имеем автоморфизм φ , действующий по правилу: для любого $sa + \sum_{i=1}^k z_i \in \langle F, + \rangle$ получим $\varphi(sa + \sum_{i=1}^k z_i) = sa + \sum_{i=1}^k \varphi_i(z_i)$.

Очевидно, элемент $d=a$ и автоморфизм φ группы $\langle F, + \rangle$ удовлетворяют (3), значит, на группе $\langle F, + \rangle$ определяем полуабелеву n -группу $\langle F, f \rangle = \text{der}_{\varphi, a} \langle F, + \rangle$, где n -арная операция f действует по правилу (4).

Доказано (теорема 5, [11]), что n -группа $\langle F, f \rangle$ является свободной в классе полуабелевых n -групп с порождающим множеством

$$X = \{-a + x_{ij} \mid i=1, \dots, k\} \cup \{0\}.$$

Свободная конечно порожденная полуабелева n -группа $\langle F, f \rangle$ построена на свободной абелевой группе $\langle (a), + \rangle + \sum_{i=1}^k \sum_{j=1}^{n-1} \langle (x_{ij}), + \rangle$, а построенный автоморфизм φ этой группы будет разложимым, значит, и $\langle F, f \rangle$ будет разложимой (согласно теореме 3), т.е. верна

Теорема 8. *Свободная конечно порожденная полуабелева n -группа $\langle F, f \rangle$ изоморфна прямому произведению одной бесконечной циклической n -группы и k неразложимых полуабелевых n -групп, $(\varphi, 0)$ -определенных на прямой сумме $n-1$ бесконечных циклических групп.*

Заключение. Хорошо известные факты из теории абелевых групп позволяют изучать полубелевы n -группы. Здесь, в частности, изучена разложимость конечно порожденной полуабелевой и абелевой n -групп.

Список литературы

1. Dörnte W. Untersuchungen unter einen verallgemeinerten Gruppenbegriff. Math. Zeitschr. 1928. №29. p. 1-19.
2. Post E.L. Polyadic groups. Trans. Amer. Math. Soc. 1940. №48. p. 208-350.
3. Глушкин Л.М. Позиционные оперативы. Мат. сб. 1965. Т.68 (110). №3. с. 444-472.
4. Hosszu M. On the explicit form of n -group operations. Publ. Math. 1963. V.10. p. 88-92.
5. Glasek K., Michalski J. and Sierocki I. On evaluation of some polyadic groups. Variag Holder-Pichir-Tempsky, Wiena. 1985. p. 159-171.
6. Щучкин Н.А. Строение конечных полуабелевых n -арных групп. Чебышевский сборник. 2016. Т.17. Выпуск 1(57). с. 260-275.
7. Гальмак А.М. n -Арные группы. Часть 1. Гомель. Гомельский гос. университет им. Ф. Скорины. 2003.
8. Щучкин Н.А. Полуциклические n -арные группы. Известия ГГУ им. Ф. Скорины. 2009. №3(54). с. 186-194.
9. Бощенко А.П., Щучкин Н.А. Конечные абелевы n -арные группы. Чебышевский сборник. 2011. Т.12. Выпуск 2(38). с. 5-14.
10. Shchuchkin N.A. Free semiabelian n -ary groups. Quasigroups and Related Systems. 2015. V.23. p. 309-317.

УДК 004.274

КВАНТОВЫЕ КОМПЬЮТЕРЫ ИЛИ НОВЫЙ УРОВЕНЬ ОБРАБОТКИ ИНФОРМАЦИИ

М.О. Матвеев¹, К.Н. Ефименко²

Донецкий национальный технический университет, г. Донецк
студент гр. РЭС-16¹, доцент кафедры прикладной математики²
max_mat@inbox.ru

Аннотация

Матвеев М.О., Ефименко К.Н. Квантовые компьютеры или новый уровень обработки информации. Рассмотрены общие теоретические основы принципов функционирования и методов создания квантовых компьютеров. Проанализированы особенности вычислительного процесса и возможные области применения квантовых вычислительных систем.

Ключевые слова: квантовый компьютер, Q-бит, направления разработки, структура, организация вычислений.

Общая постановка проблемы. Квантовый компьютер – это гипотетическое вычислительное устройство, использующее для обработки данных квантово-механические эффекты: квантового параллелизма, квантовой суперпозиции и квантовой запутанности [1-4]. Такие устройства оперируют кубитами (Q-биты – квантовые биты), которые могут одновременно принимать значение и логического нуля, и логической единицы. Поэтому с ростом количества используемых кубитов число обрабатываемых одновременно значений увеличивается в геометрической прогрессии [3].

Идея квантовых вычислений впервые была предложена Р. Фейнманом и Ю. И. Митиним. При моделировании на компьютере квантовых процессов было определено, что для решения многочастичных квантовых задач объем памяти компьютера недостаточен, так как при решении задачи с 1000 электронными спинами в памяти должно храниться 2^{1000} переменных. Поэтому все квантовые задачи, которые сейчас рассчитываются на классических компьютерах являются очень грубыми приближениями. Была высказана мысль о том, что квантовые задачи должен решать квантовый компьютер: природе задачи должен соответствовать способ ее решения. В 1995 году американский математик Шор переложил для квантового компьютера алгоритм вычисления простых множителей больших чисел. Шор показал, что если классический компьютер для нахождения множителей числа из 1000 двоичных знаков должен сделать 2^{1000} операций, то квантовому компьютеру для этого понадобится всего 1000^3 операций [4].

Существует несколько теоретических моделей квантового компьютера, которые базируются на том, что квантовая система из N двухуровневых кубитов имеет 2^N линейно независимых состояний, и таким образом, квантовое вычислительное устройство может выполнять параллельно 2^N операций [1].

Основные направления разработки квантового компьютера.

Квантовый компьютер, основанный на ядерно-магнитном резонансе – импульсный ядерный магнитно-резонансный спектрометр высокого разрешения. Спины ядер, входящих в состав атомов, образующих исследуемую молекулу считаются кубитами. Каждое ядро имеет свою частоту резонанса в данном магнитном поле. При воздействии импульсом на резонансной частоте одного из ядер оно начинает эволюционировать, остальные же ядра «молчат». Для того чтобы заставить эволюционировать второй атом, на него подается импульс другой частоты. Таким образом, процесс вычислений управляется импульсами переменного магнитного поля. Например, 1000^3 операций в алгоритме Шора для 1000-разрядного числа – это миллиард воздействий на отдельные спины и на их пары. При этом в молекуле есть прямая связь между спинами, и поэтому она является идеальной заготовкой для квантового компьютера, а сам спектрометр – готовый «процессор» для этого компьютера. Однако в настоящее время удается работать с системами с общим числом спинов не более пяти-семи, в то время как для решения полномасштабных задач их необходимо порядка 1000. Считается принципиально невозможным применение такой модели квантового компьютера.

Квантовый компьютер, основанный на использовании ионных ловушек, которые удалось «растянуть» и получить одномерный ионный кристалл, удерживаемый и в осевом, и в радиальном направлении внешними полями. У каждого иона кристалла берутся два уровня энергии, образующих один кубит. Между собой эти ионы связаны через колебания внутри одномерного кристалла, который имеет набор резонансных частот. Несмотря на большое количество экспериментов по квантовому вычислению с использованием таких кристаллов удалось получить только цепочку из 30 ионов. Но дальнейший прогресс в увеличении числа ионов связывают с созданием трехмерной лазерной стоячей волны – трехмерной совокупности точек с минимумами потенциальной энергии для поляризованных атомов.

Квантовый компьютер, основанный на твердом теле, которое может быть представлено как сверхпроводником, так и кремнием, на котором реализуется современная микроэлектроника. В нужных местах твердого тела на расстояниях порядка 100 ангстрем располагают атомы фосфора. Если на таком расстоянии расположить два атома фосфора, то облака внешних электронов немного пересекутся, что необходимо для их взаимодействия, и атомы смогут обмениваться состояниями. Один атом управляет электронами другого. Над

этим атомами делаются 50-ангстремные электроды, с помощью напряжения на электроде меняют резонансную частоту спина ядра атома фосфора, тем самым изменяя состояние атома. Предложено работать не на одном атоме, а на серии атомов; под этими электродами должна быть последовательность атомов, чтобы они действовали параллельно, тогда сформируется относительно большой сигнал, который легче регистрировать [2,4].

Структура квантового компьютера и организация вычислений. Квантовые методы выполнения вычислительных операций, а также передачи и обработки информации, уже начинают воплощаться в реально функционирующих экспериментальных устройствах, что стимулирует многообещающую работу на передовом крае науки и компьютерной техники [2].

Исследования по квантовой теории информации и квантовым вычислениям, проводимые компаниями IBM, Google, Microsoft и др. позволяют представить структуру квантового компьютера в следующем виде (рис. 1).

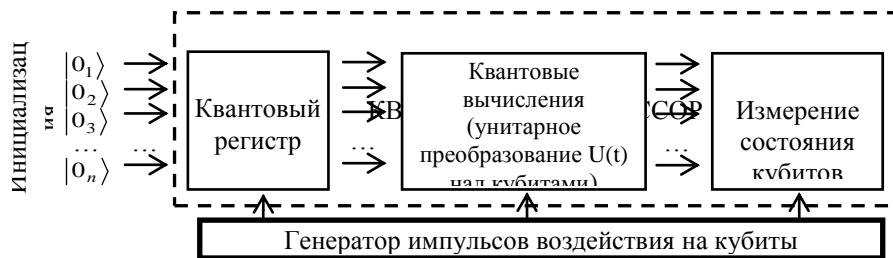


Рисунок 1 – Структура квантового компьютера

Основу структуры составляет квантовый регистр – совокупность некоторого числа N кубитов. Перед вводом данных в квантовый компьютер выполняется инициализация – все кубиты регистра приводятся в базисные состояния, то есть в состояния $|0_1\rangle, |0_2\rangle, |0_3\rangle, \dots, |0_n\rangle$. Затем каждый кубит подвергается выборочному воздействию, например, с помощью импульсов внешнего электромагнитного поля, управляемых классическим компьютером, которое переведет основные базисные состояния определенных кубитов в неосновные состояния. При этом состояние всего регистра перейдет в суперпозицию базисных состояний вида $|x\rangle = |x_1, x_2, \dots, x_n\rangle$, задающую бинарное представление числа n .

При вводе информации в квантовый компьютер состояние входного регистра, с помощью соответствующих импульсных воздействий

преобразуется в когерентную суперпозицию базисных ортогональных состояний $|\psi(0)\rangle$. В таком виде информация далее подвергается воздействию квантового процессора, выполняющего последовательность квантовых логических операций, определяемую унитарным преобразованием, действующим на состояние всего регистра. К моменту времени t в результате преобразований исходное квантовое состояние становится новой суперпозицией вида $|\psi(t)\rangle = U(|\psi(0)\rangle)$ которая и определяет результат преобразования информации на выходе.

Совокупность всех возможных операций на входе данного компьютера, формирующих исходные состояния, а также осуществляющих унитарные локальные преобразования, соответствующие алгоритму вычисления, способы подавления потери когерентности – так называемой декогерентизации (decoherence) квантовых состояний и исправления случайных ошибок, играют здесь ту же роль, что и программное обеспечение в классическом компьютере [2-4].

Вследствие того, что работа квантового компьютера будет основываться на принципе суперпозиции и вместо битов использоваться квантовые биты, которые одновременно могут находиться во всевозможных состояниях, то квантовые компьютеры для определенных классов задач будут в миллионы раз мощнее классических. Уже сейчас описаны десятки алгоритмов работы квантового компьютера и разрабатываются специальные языки программирования.

Упрощенная схема вычисления на квантовом компьютере выглядит так: берётся система кубитов, на которой записывается начальное состояние. Затем состояние системы или её подсистем изменяется посредством унитарных преобразований, выполняющих те или иные логические операции. В конце измеряется значение являющееся результатом работы. Такая концепция квантового процессора и квантовых логических вентилях была предложена в 1989 году Дэвидом Дойчем, который в 1995 году нашёл универсальный логический блок, позволяющий выполнять любые квантовые вычисления [4].

Определено, что для построения любого вычисления достаточно двух базовых операций. Квантовая система даёт результат, только с некоторой вероятностью являющийся правильным. Но за счёт небольшого увеличения операций в алгоритме можно сколь угодно приблизить вероятность получения правильного результата к единице. С помощью базовых квантовых операций можно симулировать работу обычных логических элементов, из которых сделаны обычные компьютеры. Поэтому любую задачу, которая решена сейчас, квантовый компьютер решит, и почти за такое же время [1].

Большая часть современных ЭВМ работают по такой же схеме: n бит памяти хранят состояние и каждый такт времени изменяются процессором. В квантовом случае система из n кубитов находится в состоянии, являющемся

суперпозицией всех базовых состояний, поэтому изменение системы касается *всех* 2^n базовых состояний одновременно. Теоретически новая схема может работать намного (в экспоненциальное число раз) быстрее классической. Например, практически квантовый алгоритм Гровера поиска в базе данных показывает квадратичный прирост мощности по сравнению с классическими алгоритмами.

В квантовом компьютере для реализации канала связи можно использовать эффект «квантовой телепортации», основанный на эффекте запутывания квантовых состояний двух частиц. Запутанные состояния возникают при взаимодействии двух квантовых частиц и последующем их разъединении, в результате которого состояние первой частицы строго коррелировано с состоянием второй. То есть подобная корреляция была заложена в момент взаимодействия, после чего частицы были пространственно разъединены. Таким образом, квантовый канал связи – это генератор коррелированных пар и разнесенные в пространстве квантовые частицы. При этом информация, которая была заложена в момент корреляции, сохраняется, что можно использовать для составления протокола квантовой телепортации.

Выводы. В настоящее время над квантовыми вычислениями обычно работают небольшие коллективы исследователей не только в корпорациях IBM и Intel, но и во многих крупных научных центрах и университетах по всему миру. В России аналогичные исследовательские работы ведутся во многих местах, например в МГУ, Институте Ландау и в Физико-технологическом институте. Тем не менее, универсальный квантовый компьютер представляет собой пока только гипотетическое устройство, возможность построения которого требует серьезного развития квантовой теории и фундаментальных открытий современной физики. На данный момент практическую реализацию получили лишь отдельные экспериментальные системы, исполняющие фиксированный алгоритм небольшой сложности [2].

Современные исследования посвящены не только разработке квантового компьютера, но и определению оптимальных областей его применения. Главным является то, что квантовый компьютер сможет очень быстро оптимизировать информацию и работать с большими объемами накопленных данных, даже в случае недостаточного понимания того, как их можно использовать. Квантовый компьютер сможет одновременно просчитает все возможные варианты решения поставленной задачи и моментально выдать тот, который является оптимальным. Это позволит оптимизировать множество различных процессов, начиная от медицины (например, улучшить раннее диагностирование рака) и заканчивая авиацией (например, делать более сложные автопилоты). Квантовый компьютер сможет работать со сложнейшими симуляциями и решать задачи, на решение которых обычный компьютер просто не способен. Создание полноценного квантового

компьютера, возможно, будет способствовать появлению искусственного интеллекта.

Таким образом, квантовые компьютеры являются одним из перспективных направлений развития вычислительной техники и формируют новый уровень обработки информации.

Список литературы

1. Максимов, Н. В. Архитектура ЭВМ и вычислительных систем : учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. – 5-е изд. – Москва : ФОРУМ : ИНФРА-М, 2013. – 512 с.
2. Портал TAdviser [Электронный ресурс] / Интернет-ресурс. – Режим доступа : http://www.tadviser.ru/index.php/Статья:Квантовый_компьютер_и_квантовая_связь. – Загл. с экрана.
3. Свободная энциклопедия «Википедия» [Электронный ресурс] / Интернет-ресурс. – Режим доступа : http://ru.wikipedia.org/wiki/Квантовый_компьютер. – Загл. с экрана.
4. Валиев, К. Квантовые компьютеры / К. Валиев // Открытые системы. СУБД, №5-6. – 2000. – С. 52-60.

УДК 517 955

ЗАДАЧА КОШИ ДЛЯ ПРОСТЕЙШЕГО УРАВНЕНИЯ СОБОЛЕВСКОГО ТИПА В КЛАССЕ ОБОБЩЕННЫХ ФУНКЦИЙ МЕДЛЕННОГО РОСТА

А.Л. Павлов

Государственное учреждение «Институт прикладной математики
и механики», г. Донецк

E-mail: alex4909@gmail.com

Аннотация.

Павлов А.Л. Задача Коши для простейшего уравнения соболевского типа в классе обобщенных функций медленного роста. Приведены необходимые и достаточные условия существования решения задачи Коши для уравнения $\Delta^k \partial_t^2 u - u = 0$ в пространстве медленно растущих обобщенных функций.

Ключевые слова: задача Коши.

Рассматривается задача Коши в полупространстве :

$$\Delta^k \partial_t^2 u - u = 0, t \geq 0, x \in R^n, \quad (1)$$

$$u|_{t=0} = g_0, \quad \partial_t u|_{t=0} = g_1, \quad (2)$$

где $\Delta = \partial_{x_1}^2 + \dots + \partial_{x_n}^2$ – оператор Лапласа, $k \in N$.

Уравнение (1) является одним из простейших уравнений, не разрешенных относительно старшей производной по выделенной переменной. Исследование таких уравнений начато С.Л. Соболевым.

В настоящее время имеется огромное количество работ, посвященных задачам для уравнений и систем уравнений соболевского типа [1, 2].

Под решением задачи (1), (2) часто понимают обобщенную функцию $u(t)$, зависящую гладко от параметра t и удовлетворяющую (1), (2). Такие решения будем называть сильными обобщенными решениями. Слабым обобщенным решением задачи (1), (2) будем называть обобщенную функцию u , сосредоточенную в полупространстве \bar{R}_+^{n+1} и удовлетворяющую уравнению

$$\Delta^k \partial_t^2 u - u = \Delta^k g_0 \otimes \delta_t' + \Delta^k g_1 \otimes \delta_t, \quad (3)$$

где δ_t – дельта-функция на R , δ_t' – ее производная.

В настоящей работе рассматриваются сильные обобщенные решения задачи (1), (2) со значениями в пространстве медленно растущих обобщенных функций $S'(R^n)$, а слабые обобщенные решения из пространства $D'_+(R, S'(R^n))$, которое является подпространством пространства обобщенных функций $D'(R^{n+1})$.

После применения преобразования Фурье по пространственным переменным задача (1), (2) принимает вид

$$|\sigma|^{2k} d_t^2 \hat{u}(t) + \hat{u}(t) = 0, \quad t \geq 0, \quad (4)$$

$$\hat{u}(t)|_{t=0} = \hat{g}_0, \quad \frac{d\hat{u}(t)}{dt} \Big|_{t=0} = g_1, \quad (5)$$

а уравнение (3) превращается в уравнение

$$|\sigma|^{2k} d_t^2 \hat{u} + \hat{u} = |\sigma|^{2k} \hat{g}_0 \otimes \delta'_t + |\sigma|^{2k} \hat{g}_1 \otimes \delta_t, \quad (6)$$

где \hat{f} – преобразование Фурье обобщенной функции f .

В силу изоморфности рассматриваемых пространств относительно преобразования Фурье задачи (1), (2) и (4), (5) эквивалентны. В дальнейшем будет изучаться задача (4), (5).

Решением уравнения (6) на множестве $(R^n \setminus \{0\}) \times R$ является обобщенная функция

$$\hat{u}(\sigma, t) = \eta(t) \left(\cos \frac{t}{|\sigma|^k} \hat{g}_0 + |\sigma|^k \sin \frac{t}{|\sigma|^k} \hat{g}_1 \right), \quad (7)$$

где $\eta(t) = 1$, если $t \geq 0$ и $\eta(t) = 0$, если $t < 0$.

Функция $\hat{u}(\sigma, t)$, является сильным обобщенным решением задачи (4), (5) для любых $\hat{g}_0 \in S'(R^n)$, $\hat{g}_1 \in S'(R^n)$, носители которых не содержат точку $\sigma = 0$. Задача состоит в описании начальных данных \hat{g}_0, \hat{g}_1 , для которых задача (4), (5) имеет слабое обобщенное решение и при каких условиях оно является сильным обобщенным решением.

В настоящей работе приведены необходимые и достаточные условия на начальные данные из пространства медленно растущих обобщенных функций, обеспечивающие существование слабого обобщенного решения (1), (2), имеющего следы на гиперплоскостях $t = \text{const}$, гладко зависящие от t и

сильного обобщенного решения этой задачи в классе обобщенных функций медленного роста.

В силу единственности слабого обобщенного решения задачи (4), (5) в пространстве $D'_+(R, S'(R^n))$ [3] всякое слабое обобщенное решение этой задачи, соответствующее начальным данным \hat{g}_0 и \hat{g}_1 совпадает с функцией (7) на множестве $(R^n \setminus \{0\}) \times R$, то есть является продолжением этой функции на $R^n \times R$.

Разрешимость задачи (4), (5) сводится к возможности построения регуляризации обобщенной функций (7) и выбора такой "добавки", чтобы ее сумма с регуляризацией была слабым или сильным обобщенным решением этой задачи.

В построении указанной регуляризации существенно используются свойства функций

$$q_1(\sigma, t) = \cos \frac{t}{|\sigma|^k} \quad \text{и} \quad q_2(\sigma, t) = |\sigma|^k \sin \frac{t}{|\sigma|^k},$$

которые являются решениями уравнения (4) при $(\sigma, t) \in (R^n \setminus \{0\}) \times \bar{R}_+$ и при $t = 0, \sigma \in R^n \setminus \{0\}$ удовлетворяют условиям (5) соответственно при $\hat{g}_0 = 1, \hat{g}_1 = 0$ и $\hat{g}_0 = 0, \hat{g}_1 = 1$.

Рассмотрим регуляризации функций $q_1(\sigma, t)\hat{g}_0$ и $q_2(\sigma, t)\hat{g}_1$, построенные методом вычитания:

$$([q_2(\sigma, t)\hat{g}_1]_j, \varphi) = \left(\hat{g}_1, |\sigma|^k \sin \frac{t}{|\sigma|^k} \varphi_j(\sigma) \right) \left([q_1(\sigma, t)\hat{g}_0]_j, \varphi \right) = \left(\hat{g}_0, \cos \frac{t}{|\sigma|^k} \varphi_j(\sigma) \right)$$

, где $\varphi \in S(R^n), \varphi_j(\sigma) = \varphi(\sigma) - \mu(\sigma) \left(\sum_{|\alpha| \leq j} \varphi^\alpha(0) \frac{\sigma^\alpha}{\alpha!} \right), \mu(\sigma) \in C_0^\infty(R^n), \text{supp } \mu$

принадлежит некоторой ограниченной окрестности точки $\sigma = 0$ и $\mu(\sigma) = 1$ в некоторой меньшей окрестности этой точки.

Теорема 1. Для любой обобщенной функции $h \in H_s^l$ существуют такие числа j, l', s' и p , что определены обобщенные функции $[q_i(\sigma, t)h]_j, i = 1, 2$, принадлежащие пространству $C_p^2(\bar{R}_+, H_{s'}^{l'})$ и справедливо неравенство

$$\| [q_i(\sigma, t)h]_j \|_{C_p^2(\bar{R}_+, H_{s'}^{l'})} \leq C \| h \|_{s'}^{l'},$$

где $C > 0$ – некоторое число не зависящее от h .

Рассмотрим функции

$$w_{\alpha}(\hat{g}_0, \hat{g}_1, t) = (-1)^{|\alpha|} \left((\hat{g}_0, \mu(\sigma) \cos \frac{t}{|\sigma|^k} \frac{\sigma^{\alpha}}{\alpha!} + \right. \\ \left. + \left(\hat{g}_1, \mu(\sigma) |\sigma|^k \sin \frac{t}{|\sigma|^k} \frac{\sigma^{\alpha}}{\alpha!} \right) \right).$$

Они определены для любых начальных данных \bar{g}_0, \bar{g}_1 из $S'(R^n)$ при достаточно больших значениях $|\alpha|$.

Построение слабого обобщенного решения задачи (4), (5) с помощью регуляризации функции (7) приводит к следующим условиям:

$$w_{\alpha} \left(|\sigma|^{2kq} \hat{g}_0, |\sigma|^{2kq} \hat{g}_1, t \right) \in C^{2(q+1)}(\bar{R}_+), \text{ где } q = \left\lfloor \frac{j-|\alpha|}{2k} \right\rfloor; \quad (8)$$

$$(-1)^q w_{\beta}^{(2q)} \left(|\sigma|^{2k(q+1)} \hat{g}_0, |\sigma|^{2k(q+1)} \hat{g}_1, 0 \right) = \\ = (-1)^{|\beta|} \left(|\sigma|^{2k} \hat{g}_0, \mu(\sigma) \frac{\sigma^{\beta}}{\beta!} \right), |\beta| \leq j - 2k; \quad (9)$$

$$(-1)^q w_{\beta}^{(2q+1)} \left(|\sigma|^{2k(q+1)} \hat{g}_0, |\sigma|^{2k(q+1)} \hat{g}_1, 0 \right) = \\ = (-1)^{|\beta|} \left(|\sigma|^{2k} \hat{g}_1, \mu(\sigma) \frac{\sigma^{\beta}}{\beta!} \right), |\beta| \leq j - 2k. \quad (10)$$

Их существенность содержится в следующем утверждении.

Теорема 2. Слабое обобщенное решение задачи Коши (4), (5) для начальных данных $\hat{g}_0 \in H_s^j, \hat{g}_1 \in H_s^j$, принадлежащие пространству $C^2(\bar{R}_+, S'(R^n))$ существует тогда и только тогда, когда выполняются условия (8), (9), (10), где j – натуральное число, обеспечивающее существование регуляризации функции (7) и принадлежность ее пространству $C^2(\bar{R}_+, S'(R^n))$.

Полученное в теореме 2 слабое обобщенное решение задачи (4), (5) является сильным обобщенным решением, если при $|\alpha| \leq j$, выполнены условия:

$$(-1)^q w_{\alpha}^{(2q)} \left(|\sigma|^{2kq} \hat{g}_0, |\sigma|^{2kq} \hat{g}_1, 0 \right) = (-1)^{|\alpha|} \left(\hat{g}_0, \mu(\sigma) \frac{\sigma^{\alpha}}{\alpha!} \right) \quad (11)$$

$$(-1)^q w_\alpha^{(2q+1)} \left(|\sigma|^{2kq} \hat{g}_0, |\sigma|^{2kq} \hat{g}_1, 0 \right) = (-1)^{|\alpha|} \left(\hat{g}_1, \mu(\sigma) \frac{\sigma^\alpha}{\alpha!} \right). \quad (12)$$

Теорема 3. Если $\hat{g}_0 \in H_s^l$, $\hat{g}_1 \in H_s^{l'}$ и выполнены условия (8), (11), (12), то существует сильное обобщенное решение задачи Коши (4), (5), принадлежащее пространству $C^2(\bar{R}_+, H_{s'}^{l'})$, где l', s' – некоторые числа зависящие от l и s .

Литература

- [1] Демиденко Г.В., Успенский С.В. Уравнения и системы, не разрешенные относительно старшей производной. Новосибирск: Науч.кн., 1998.
- [2] Свешников А.Г., Альшин А.Б., Корпусов М.О., Плетнер Ю.А. Линейные и нелинейные уравнения соболевского типа. М.: Физматлит, 2007.
- [3] Павлов А.Л. Задача Коши для уравнения типа Соболева-Гальперна в пространствах функций степенного роста // Мат. сб. 1993. Т. 184, № 11. С. 3–20.

УДК 517.984.52

К СПЕКТРАЛЬНОЙ ТЕОРИИ ВЕКТОРНОЗНАЧНЫХ ОПЕРАТОРОВ ШТУРМА-ЛИУВИЛЛЯ С СУММИРУЕМЫМИ ПОТЕНЦИАЛАМИ И ТОЧЕЧНЫМИ ВЗАИМОДЕЙСТВИЯМИ

Я.И. Грановский

Отдел уравнений в частных производных ГУ ИПММ

yarvodoley@mail.ru

Аннотация.

Грановский Я.И. К спектральной теории векторнозначных операторов штурма-лиувилля с суммируемыми потенциалами и точечными взаимодействиями. Рассматривается матричный оператор Штурма-Лиувилля с суммируемым потенциалом и точечными взаимодействиями. Требования к потенциалу обусловлены физическими моделями, в которых данный оператор имеет широкое применение. Основной результат – выявление природы спектра данного оператора.

Ключевые слова: потенциал, оператор Штурма-Лиувилля, дифференциальные операторы

1. Введение. Дифференциальные операторы с точечными взаимодействиями имеют широкое физическое приложение в качестве точно

разрешимых моделей, описывающих сложные физические явления (см., например, [2] и [3], а также литературу в них). Важными представителями данного класса операторов являются дифференциальные операторы, коэффициенты которых имеют сингулярный носитель на дискретном множестве изолированных точек. Наиболее известным примером является оператор $H_{X,\alpha,q}$, ассоциированный с формальным дифференциальным выражением

$$l_{X,\alpha,q} := -\frac{d^2}{dx^2} + q(x) + \sum_{x_n \in X} \alpha_n \delta(x - x_n).$$

Данный оператор описывает дельта-взаимодействие на дискретном множестве $X = \{x_n\}_{n \in \mathbb{N}} \subset \mathbb{R}$, а коэффициенты α_n называются силами взаимодействия в точке $x = x_n$. Впервые данная модель была исследована Кронигом и Пенни (см. [4]).

2. Матричный оператор Штурма-Лиувилля с суммируемым потенциалом.

Основным объектом данной работы является векторнозначное дифференциальное выражение Штурма-Лиувилля с суммируемым матричным потенциалом $Q(\cdot) = Q^*(\cdot) \in L^1(\mathbb{R}_+, \mathbb{C}^{m \times m})$ и конечным числом точечных взаимодействий:

$$L_{X,\alpha,Q} := -\frac{d^2}{dx^2} + Q(x) + \sum_{x_n \in X} \alpha_n \delta(x - x_n), \quad x \in \mathbb{R}_+ = (0, \infty).$$

Здесь $X = \{x_n\}_{n=1}^p \subset \mathbb{R}_+$ – конечная строго возрастающая последовательность, $x_{n+1} > x_n, n \in \{1, \dots, p\}, p < \infty$, и $\alpha = \{\alpha_n\}_1^p \subset \mathbb{C}^{m \times m}$, $\alpha_n = \alpha_n^*$.

Заметим, что гамильтониан $H_{X,0,q}$ с $\alpha_n = 0, n \in \mathbb{N}$ идентифицируется с реализацией Дирихле H_Q следующего выражения:

$$L_Q := -\frac{d^2}{dx^2} + Q(x), \quad x \in \mathbb{R}_+,$$

рассматриваемого в $L^2(\mathbb{R}_+, \mathbb{C}^m)$.

Теперь рассмотрим следующее уравнение:

$$L_Q(Y(x, z)) = zY(x, z), \quad x \in \mathbb{R}_+, z \in \mathbb{C}.$$

Пусть $C(x, z)$ и $S(x, z)$ – его матричнозначные решения, удовлетворяющие начальным условиям:

$$\begin{aligned} C(0, z) &= S'(0, z) = I_m, \\ S(0, z) &= C'(0, z) = \mathbb{O}_m, \quad z \in \mathbb{C}. \end{aligned}$$

Здесь I_m и \mathbb{O}_m – единичная и нулевая матрицы порядка m соответственно.

Введём следующие обозначения:

$$\begin{aligned} N_1(z) &= \frac{I_m}{2i\sqrt{z}} + \frac{1}{2i\sqrt{z}} \int_0^\infty e^{it\sqrt{z}} Q(t) S(t, z) dt, \\ N_2(z) &= \frac{I_m}{2} - \frac{1}{2i\sqrt{z}} \int_0^\infty e^{it\sqrt{z}} Q(t) C(t, z) dt, \quad z \in \mathbb{C} \setminus \{0\}. \end{aligned}$$

Далее, положим:

$$Af := \mathcal{L}_Q(f), \quad x \in \mathbb{R}_+, f \in \text{dom}(A),$$

$$\text{dom}(A) := \left\{ f \in L^2(\mathbb{R}_+, \mathbb{C}^m): \begin{array}{l} f, f' \in AC_{\text{loc}}(\mathbb{R}_+, \mathbb{C}^m), \\ \mathcal{L}_Q(f) \in L^2(\mathbb{R}_+, \mathbb{C}^m), f(0) = f'(0) = 0 \end{array} \right\}$$

и отметим, что оператор A совпадает с минимальным оператором H_Q^{\min} , ассоциированным с выражением \mathcal{L}_Q . Сопряжённый оператор имеет вид:

$$A^*f := \mathcal{L}_Q(f), \quad x \in \mathbb{R}_+, f \in \text{dom}(A^*),$$

$$\text{dom}(A^*) := \left\{ f \in L^2(\mathbb{R}_+, \mathbb{C}^m): \begin{array}{l} f, f' \in AC_{\text{loc}}(\mathbb{R}_+, \mathbb{C}^m), \\ \mathcal{L}_Q(f) \in L^2(\mathbb{R}_+, \mathbb{C}^m) \end{array} \right\}$$

и совпадает с максимальным оператором H_Q^{\max} , ассоциированным с выражением \mathcal{L}_Q .

Лемма 1. (i) Тройка $\Pi = \{\mathcal{H}, \Gamma_0, \Gamma_1\}$, где

$$\mathcal{H} = \mathbb{C}^m, \Gamma_0 f = f(0), \Gamma_1 f = f'(0), f = (f_1, \dots, f_m)^T \in \text{dom}(H_{\max}),$$

является граничной тройкой для оператора $A^* = H_{\max}$.

(ii) Для функции Вейля $M(\cdot)$, соответствующей граничной тройке Π , выполняется следующее равенство:

$$N_1(z)M(z) = N_2(z), \quad z \in \mathbb{C}_+ = \{z \in \mathbb{C}: \text{Im}z > 0\}.$$

Теорема 1. Оператор H_Q ограничен снизу, и его отрицательный спектр или конечен, или дискретен с единственной точкой накопления $x = 0$. Его неотрицательный спектр является чисто абсолютно непрерывным.

Замечание. Теорема 1 обобщает классический результат Титчмарша (см. главу 5 в [1]) на случай оператора Штурма-Лиувилля с матричнозначным суммируемым потенциалом и совпадает с указанным результатом в скалярном случае ($m = 1$).

3. Матричный оператор Штурма-Лиувилля с суммируемым потенциалом и точечными взаимодействиями.

Теперь рассмотрим матричнозначный оператор Шрёдингера $H_{X,\alpha,Q}$ в $L^2(\mathbb{R}_+, \mathbb{C}^{m \times m})$, ассоциированный с дифференциальным выражением (1). Определим в $L^2(\mathbb{R}_+)$ замкнутый симметрический оператор \tilde{A} , ассоциированный с выражением $\mathcal{L}_{X,\alpha,Q}$:

$$(\tilde{A}f)(x) := \mathcal{L}_Q(f)(x), \quad x \in \mathbb{R}_+ \setminus X, \quad f \in \text{dom}(\tilde{A}),$$

$$\text{dom}(\tilde{A}) = \left\{ f \in L^2(\mathbb{R}_+ \setminus X, \mathbb{C}^m): \begin{array}{l} f, f' \in AC_{\text{loc}}(\mathbb{R}_+ \setminus X, \mathbb{C}^m) \\ \mathcal{L}_Q(f) \in L^2(\mathbb{R}_+, \mathbb{C}^m) \\ f(0) = 0, f(x_n \pm) = 0, n \in \{1, \dots, p\} \\ f'(0) = 0, f'(x_n \pm) = 0, n \in \{1, \dots, p\} \end{array} \right\}.$$

Отметим, что \tilde{A} является минимальным оператором, ассоциированным с дифференциальным выражением $\mathcal{L}_{X,\alpha,Q}$. В дальнейшем будем обозначать минимальный оператор $H_{X,\alpha,Q}^{\min}$. Имеем:

$$\text{dom}(H_{X,\alpha,Q}^{\min}) = \{f \in \text{dom}(H_Q^{\max}): f(x_n) = f'(x_n) = 0, n \in \{1, \dots, p\}\}.$$

Сопряжённый оператор имеет вид:

$$(\tilde{A}^* f)(x) := \mathcal{L}_Q(f)(x), \quad x \in \mathbb{R}_+ \setminus X, \quad f \in \text{dom}(\tilde{A}^*),$$

$$\text{dom}(\tilde{A}^*) = \left\{ f \in L^2(\mathbb{R}_+ \setminus X, \mathbb{C}^m): \begin{array}{l} f, f' \in AC_{\text{loc}}(\mathbb{R}_+ \setminus X, \mathbb{C}^m) \\ \mathcal{L}_Q(f) \in L^2(\mathbb{R}_+, \mathbb{C}^m) \end{array} \right\}.$$

Оператор \tilde{A}^* называется максимальным оператором, ассоциированным с $\mathcal{L}_{X,\alpha,Q}$, и обозначается $H_{X,\alpha,Q}^{\max}$. Рассмотрим самосопряжённое расширение $H_{X,\alpha,Q}$, определённое соотношением

$$H_{X,\alpha,Q} = \tilde{A}^* \upharpoonright \text{dom}(H_{X,\alpha,Q}).$$

Имеем:

$$\text{dom}(H_{X,\alpha,Q}) = \left\{ f \in \text{dom}(H_Q^{\max}): \begin{array}{l} f(0) = 0, f(x_n +) = f(x_n -), \\ f'(x_n +) - f'(x_n -) = \alpha_n f(x_n), \\ n \in \{1, \dots, p\} \end{array} \right\},$$

где α_n – самосопряжённые матрицы.

Основной результат данной работы представлен следующей теоремой.

Теорема 2. Пусть $Q = Q^* \in L^1(\mathbb{R}_+, \mathbb{C}^{m \times m})$, и пусть $H_{X,\alpha,Q}$ – гамильтониан, ассоциированный с $\text{dom}(H_{X,\alpha,Q})$. Тогда положительная часть разложения единицы $E_{H_{X,\alpha,Q}}(\mathbb{R}_+)H_{X,\alpha,Q}$ оператора $H_{X,\alpha,Q}$ унитарно эквивалентна положительной части разложения единицы $E_{H_Q}(\mathbb{R}_+)H_Q$ реализации Дирихле $H_Q = H_{X,0,Q}$. В частности, спектр оператора $E_{H_{X,\alpha,Q}}(\mathbb{R}_+)H_{X,\alpha,Q}$ является чисто абсолютно непрерывным постоянной спектральной кратности m .

Кроме того, гамильтониан $H_{X,\alpha,Q}$ является полуограниченным снизу, и его отрицательный спектр или конечен, или является сходящейся к нулю последовательностью.

Литература

1. Титчмарш Э. Ч. Разложения по собственным функциям, связанные с дифференциальными уравнениями второго порядка. Том 1. Пер. с англ. В.Б. Лидского. Под ред. Б.М. Левитана. – М.: Издательство иностранной литературы, 1960. – 278 с.
2. S. Albeverio, F. Gesztesy, R. Hoegh-Krohn, and H. Holden. Solvable models in quantum mechanics. Second edition. AMS Chelsea Publishing, Providence, R.I., 2005. With an Appendix by P. Exner.
3. S. Albeverio and P. Kurasov. Singular perturbations of differential operators. Solvable Schrödinger type operators. London Mathematical Society Lecture Note Series, 271. Cambridge University Press, Cambridge, 2000.
4. R. de L. Kronig and W. G. Penney. Quantum mechanics of electrons in crystal lattices. Proc. Roy. Soc. London Ser. A 130 (1931), 499-513.

УДК 517.518

ОБ ОТРИЦАТЕЛЬНЫХ КВАДРАТАХ ФУНКЦИЙ, НЕ ЯВЛЯЮЩИХСЯ
РАДИАЛЬНО ПОЛОЖИТЕЛЬНО ОПРЕДЕЛЁННЫМИ

Оридорога Л.Л.

ГОУ ВПО «Донецкий национальный университет», г. Донецк.

vremenny-orid@mail.ru

Аннотация

Оридорога Л.Л. *Об отрицательных квадратах функций, не являющихся радиально положительно определёнными. Рассматриваются функции не являющиеся радиально положительно определёнными в данной размерности. Для функции Ω_n явно указан $n + 1$ -мерный набор точек, на котором она имеет отрицательный квадрат. Также приведены результаты о бесконечности числа отрицательных квадратов функций, не являющиеся радиально положительно определёнными, в размерности $n \geq 2$.*

Ключевые слова: функции

Определение 1. Функция $f \in C[0, \infty)$ называется радиально положительно определённой в размерности n функцией (или функцией класса Φ_n), если она непрерывна в нуле и для каждого множества точек $\{x_j\}_{j=1}^m \subset \mathbb{R}^n$ матрица $(f(\|x_j - x_k\|))_{j,k=1}^m$ неотрицательно определённая.

Известно, что $f \in \Phi_n$ в том и только том случае, если функция f имеет представление

$$f(t) = \int_0^\infty \Omega_n(st) d\sigma(s),$$

где $\sigma(s)$ ограниченная неубывающая функция, а Ω_n задаётся равенством

$$\Omega_n(z) = \Gamma\left(\frac{n}{2}\right) \cdot \left(\frac{2}{z}\right)^{\frac{n-2}{2}} \cdot J_{\frac{n-2}{2}} = 1 - \frac{z^2}{2 \cdot n} + \frac{z^4}{2 \cdot 4 \cdot n \cdot (n+2)} - \frac{z^6}{2 \cdot 4 \cdot 6 \cdot n \cdot (n+2) \cdot (n+4)} \pm \dots$$

Очевидно, что $\Phi_{n+1} \subset \Phi_n$.

Следующая конструкция показывает, что вложение строгое. А именно, что $\Omega_n \notin \Phi_{n+1}$.

Теорема 1. Пусть $x_1, x_2, x_3, \dots, x_{n+2}$ — вершины правильного симплекса в \mathbb{R}^{n+1} с ребром l . x_{n+3} — центр этого симплекса. Тогда матрица $(\Omega_n(\|x_j - x_k\|))_{j,k=1}^{n+3}$ при достаточно малых l имеет одно отрицательное собственное значение.

Доказательство.

Обозначим радиус сферы, описанной около данного симплекса, r .

Известно, что в этом случае $l = r \sqrt{\frac{2n+4}{n+1}}$.

Заметим, что матрица $(\Omega_n(\|x_j - x_k\|))_{j,k=1}^{n+3}$ имеет вид

$$\begin{pmatrix} 1 & a & a & \dots & a & b \\ a & 1 & a & \dots & a & b \\ a & a & 1 & \dots & a & b \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a & a & a & \dots & 1 & b \\ b & b & b & \dots & b & 1 \end{pmatrix}, \text{ где } a = \Omega_n(l), b = \Omega_n(r). \text{ Данная матрица имеет}$$

собственное число $1 - a$ кратности $n + 1$, а также собственные числа $\frac{2+(n+1)a+\sqrt{(n+1)^2a^2+4(n+2)b^2}}{2}$ и $\frac{2+(n+1)a-\sqrt{(n+1)^2a^2+4(n+2)b^2}}{2}$. Очевидно, что все собственные числа кроме последнего положительны. Учитывая, что $\min \Omega_n(x) \geq \frac{2}{n+1}$ последнее собственное значение отрицательно при $(2 + (n + 1)a)^2 < (n + 1)^2a^2 + 4(n + 2)b^2$, т.е. при $1 + (n + 1)a - (n + 2)b^2 < 0$.

Заметим, что при $l \rightarrow 0, a = \Omega_n(l) = 1 - \frac{l^2}{2n} + \frac{l^4}{8n(n+2)} + o(l^4) = 1 - \frac{n+2}{n(n+1)}r^2 + \frac{n+2}{2n(n+1)^2}r^4 + o(r^4)$ и $b^2 = \Omega_n^2(r) = \left(1 - \frac{1}{2n}r^2 + \frac{1}{8n(n+2)}r^4 + o(r^4)\right)^2 = 1 - \frac{1}{n}r^2 + \frac{n+1}{2n^2(n+2)}r^4 + o(r^4)$. Таким образом $1 + (n + 1)a - (n + 2)b^2 = -\frac{r^4}{2n^2(n+1)} + o(r^4)$, т.е. отрицательно, при достаточно малых r .

Замечание. При $n > 1$ матрица $(\Omega_n(\|x_j - x_k\|))_{j,k=1}^{n+2}$, где $x_1, x_2, x_3, \dots, x_{n+2}$ — вершины правильного симплекса в \mathbb{R}^{n+1} с ребром l всегда положительно определённая.

Определение 2. Функция $F: \mathbb{R}^n \rightarrow \mathbb{R}$ называется имеющей k отрицательных квадратов, если существует множество $\{x_j\}_{j=1}^m \subset \mathbb{R}^n$, для которого матрица $(F(x_j - x_k))_{j,k=1}^m$ имеет k отрицательных собственных

чисел, но ни для какого набора точек эта матрица не имеет более k отрицательных собственных чисел.

Если для любого k существует множество $\{x_j\}_{j=1}^m \subset \mathbb{R}^n$, для которого матрица $\left(F(x_j - x_k)\right)_{j,k=1}^m$ имеет k отрицательных собственных чисел, то функция F называется имеющей бесконечное число отрицательных квадратов.

Известно, что при $n = 1$ функция может иметь любое количество отрицательных квадратов.

Теорема 2. Если функция f не является радиально положительно определённой в размерности $n \geq 3$, то функция $F(x) = f(\|x\|)$ имеет бесконечное число отрицательных квадратов.

В случае размерности 2 на текущий момент неизвестно существуют ли радиальные функции с конечным (но не равным 0) числом отрицательных квадратов. Но следующая теорема заставляет предположить, что таких функций не существует.

Теорема 3. Функция $F(x) = \Omega_1(\|x\|)$ имеет в размерности 2 бесконечное число отрицательных квадратов.

Более подробно с приведенными результатами можно ознакомиться в [1], [2].

Литература

1. L. Golinskii, M. Malamud, L. Oridoroga, Schoenberg matrices of radial positive definite functions and Riesz sequences of translates in $L^2(\mathbb{R}^n)$, J. Fourier Anal. Appl. 21 (2015), no. 5, 915–960.
2. L. Golinskii, M. Malamud, L. Oridoroga, Radial positive definite functions and Schoenberg matrices with negative eigenvalues, arxiv.org/pdf/1502.07179.pdf

УДК 517.9

УСЛОВИЯ ПОДЧИНЕННОСТИ ДЛЯ ТЕНЗОРНОГО ПРОИЗВЕДЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ ПОЛИНОМОВ

Лиманский Д.В.

Донецкий национальный университет
кафедра математического анализа и дифференциальных уравнений

E-mail: 4125aa@gmail.com

Аннотация

Лиманский Д.В. Условия подчиненности для тензорного произведения дифференциальных полиномов. Рассматривается задача об описании линейного пространства минимальных дифференциальных операторов, подчиненных тензорному произведению двух обыкновенных дифференциальных полиномов в пространстве с равномерной нормой. Получено описание указанного пространства для ряда случаев, когда символы сомножителей содержат вещественные и комплексные нули различной кратности.

Ключевые слова: полином

Общая постановка проблемы

Пусть Ω — область в \square^n , $p \in [1, \infty]$. Рассматривается задача об описании линейного пространства $L_{p,\Omega}(P)$ минимальных дифференциальных операторов $Q(D)$, подчиненных фиксированному дифференциальному полиному $P(D)$, т. е. пространства операторов $Q(D)$, для которых справедлива априорная оценка

$$\|Q(D)f\|_{L^p(\Omega)} \leq C_1 \|P(D)f\|_{L^p(\Omega)} + C_2 \|f\|_{L^p(\Omega)}, \quad f \in C_0^\infty(\Omega),$$

с некоторыми постоянными $C_1, C_2 > 0$, не зависящими от выбора f .

Здесь $D = (D_1, \dots, D_n)$, $D_k := -i \frac{\partial}{\partial x^k}$, $k \in \{1, \dots, n\}$; символы $Q(\xi)$ и $P(\xi)$ операторов $Q(D)$ и $P(D)$ соответственно — полиномы от вещественных переменных $\xi = (\xi_1, \dots, \xi_n)$ над полем \square .

При $p=2$ и ограниченной области Ω эта задача была исчерпывающе решена Л. Хермандером [4]. Используя этот критерий, в той же работе [4] было показано, что для тензорного произведения

$$P(D) = P_1(D_1, \dots, D_{p_1}, 0, \dots, 0) P_2(0, \dots, 0, D_{p_1+1}, \dots, D_n) \quad (1)$$

двух дифференциальных операторов $P_1(D)$ и $P_2(D)$, действующих по различным переменным, пространство $L_{2,\Omega}(P)$ равно тензорному произведению $L_{\infty,\square^{p_1}}(P_1) \otimes L_{\infty,\square^{p_2}}(P_2)$.

В работах [1-2] указанная задача рассматривалась для $p = \infty$, $\Omega = \square^n$ и эллиптических операторов $P_k(D)$ в (1). Так, в [1] доказано, что в случае невырожденных символов $P_k(\xi)$, т. е. $P_k(\xi) \neq 0 \quad \forall \xi \in \square^n$, пространство $L_{\infty,\square^n}(P)$ такое же, как и в описанной выше теореме Л. Хермандера. Но если символы $P_k(\xi)$ однородны, то $L_{\infty,\square^n}(P)$ минимально возможное: оно совпадает с линейной оболочкой тождественного оператора I и оператора $P(D)$ (см. [2]).

Исследования

В работе рассматривается задача описания пространства $L_{\infty,\square^2}(P)$ для тензорного произведения $P(D) = p_1(D_1)p_2(D_2)$, где $p_k(D_k)$ — обыкновенные дифференциальные полиномы. Эта задача была решена в [3] для случая, когда все нули первого сомножителя $p_1(\xi_1)$ вещественные и простые. Оказалось, что в этом случае пространство $L_{\infty,\square^2}(P)$ тем “беднее”, чем больше вещественных нулей (с учетом кратности) имеет второй сомножитель $p_2(\xi_2)$.

Мы распространяем результаты работы [3] на ряд случаев, когда не обязательно все нули одного из сомножителей $p_k(\xi_k)$ в произведении $P(\xi) = p_1(\xi_1)p_2(\xi_2)$ вещественные и простые.

Предложение 1. Пусть $P(\xi) = p_1(\xi_1)p_2(\xi_2)$, где $p_1(\xi_1) = \xi_1^l$, а $p_2(\xi_2)$ не имеет вещественных нулей, $\deg p_2 = m$. Тогда пространство $L_{\infty,\square^2}(P)$ — линейная оболочка дифференциальных мономов $D_1^\alpha D_2^\beta$, где $\alpha \leq l$, $\beta l < \alpha m$, и операторов I и $P(D)$.

Предложение 2. Пусть $P(\xi) = p_1(\xi_1)p_2(\xi_2)$, где $p_1(\xi_1) = \xi_1^2(\xi_1^2 + 1)$ и $p_2(\xi_2) = \xi_2^2(\xi_2 - 1)(\xi_2^2 + 1)$. Тогда пространство $L_{\infty,\square^2}(P)$ — линейная

оболочка дифференциальных полиномов $D_1^{2+\alpha} D_2^{2+\beta} (D_2 - I)$, где $\alpha \leq 2$, $\beta \leq 2$, и операторов I и $D_1 D_2 (D_2 - I)$.

Выводы

В настоящей работе описана структура пространства $L_{\infty, \square^2} (P)$ операторов $Q(D)$, подчиненных в $C(\square^2)$ -норме тензорному произведению $P(D) = p_1(D_1) \otimes p_2(D_2)$ двух обыкновенных дифференциальных полиномов. Приведенные выше предложения 1 и 2 говорят о том, что при наличии среди нулей символа $P(\xi)$ одновременно кратных вещественных и не вещественных нулей структура пространства $L_{\infty, \square^2} (P)$ может быть нетривиальной. В этих случаях имеют место новые эффекты по сравнению с работами [1-3].

Литература

1. Лиманский Д. В. Эллиптические и слабо коэрцитивные системы операторов в пространствах Соболева / Д. В. Лиманский, М. М. Маламуд. – Матем. сборник. – 2008. – Т. 199. – № 11. – С. 75-112.
2. Лиманский Д. В. Об оценках для тензорного произведения двух однородных эллиптических операторов / Д. В. Лиманский. – Укр. мат. вісник. – 2011. – Т. 8. – № 1. – С. 101-111.
3. Лиманський Д. В. Умови підпорядкованості для тензорного добутку двох звичайних диференціальних операторів / Д. В. Лиманський. – Доповіді НАН України. – 2012. – № 4. – С. 25-29.
4. Хермандер Л. К теории общих дифференциальных операторов в частных производных / Л. Хермандер. – М.: ИЛ, 1959. – 131 с.

УДК 004.514

СРАВНИТЕЛЬНЫЙ АНАЛИЗ CMS СИСТЕМ ДЛЯ СОЗДАНИЯ САЙТА-ВИЗИТКИ

А.В.Саксон¹, А.Д. Шевцова², И.Ю. Анохина³, Е.В. Перинская⁴

1-2 - студентки гр. ОмДск.-16; 3- доцент кафедры прикладной математики; 4 - ассистент кафедры прикладной

Донецкий национальный технический университет, г. Донецк

кафедра прикладной математики

IngaTula@mail..ru

Аннотация

Саксон А.В., Шевцова А.Д., Анохина И.Ю., Перинская Е.В. Сравнительный анализ CMS систем для создания сайта-визитки. В статье проанализированы популярные бесплатные системы управления контентом сайта CMS, которые можно использовать для создания сайта-визитки. Проведен сравнительный анализ CMS по выбранным критериям. Подробно рассмотрены особенности разработки сайтов на платформах: WordPress, Ucoz, Wix и Moozle.

***Ключевые слова:** системы управления контентом сайта, рейтинг бесплатных CMS, Content Management System, WordPress, Wix, Ucoz, Mozello*

Введение. Сегодня утверждение о том, что если вашего бизнеса нет в Интернете, значит, вас нет в бизнесе, стало аксиомой. Для каждого из нас сеть – это, как минимум, информационный ресурс, причем самый оперативный. Для компаний это так называемое «рекламное поле». Грамотно разработанный сайт является отличной визитной карточкой не только предприятия, но и одного человека.

Каждый выпускник вуза мечтает о хорошей и интересной работе. Ему предстоит поиск работы и вакансий, заполнение резюме, собеседование. Не всегда на одной странице резюме или в получасовой беседе мы сможем правильно рассказать о своих навыках, умениях возможностях. Сделанный сайт с уже набранным количеством посетителей с описанием наших профессиональных возможностей, со списком опубликованных статей, интересных тем курсовых работ, хобби даст достаточно полное представление о наших знаниях и выделит из числа конкурентов.

Постановка проблемы. Еще несколько лет назад разработка собственного сайта была процессом трудоемким, требующим от создателя профессиональных знаний и получившая определение как «сборка сайта по кирпичикам». Созданный сайт нужно было еще и разместить в сети на одном из хостингов.

С появлением систем управления содержимым сайта CMS (Content Management System) пользователь получил готовую функциональную

платформу для разработки сайта. Сейчас широко используется несколько десятков CMS. Мы поставили задачу выявить критерии, предъявляемые к CMS и, используя их, сравнить функциональные возможности различных систем, которые можно рекомендовать для создания сайта – визитки.

Выбор оптимальных бесплатных CMS. Критерии отбора.

На рис.1 показан рейтинг первых десяти бесплатных CMS, полученный при анализе данных за 2016г. компанией ITrack, черным цветом выделены системы с положительным трендом и штриховкой – с отрицательным. В столбце 1 – названия CMS, в столбце 2 – процент сайтов, существующих на данной платформе относительно общего количества сайтов, созданных в бесплатных системах; в столбце 3 – динамика изменения за год; столбец 4 – общее количество существующих на платформе сайтов [1].

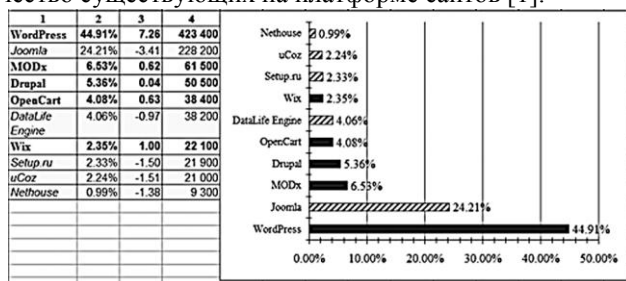


Рисунок 1 – Рейтинг бесплатных CMS

При анализе систем рассматривается множество критериев. В большинстве случаев системы рейтинуются по нескольким позициям:

- степени обеспечиваемой безопасности (защита от автоматического заполнения форм, разделение прав доступа, уведомления администратору, наличие защиты от получения рекламного или нежелательного контента);
- поддержка сервисов Интернета (публичный форум и рассылка);
- удобство работы и возможностям интерфейса (юзабилити), включающее в себя встроенный редактор изображений, пакетную загрузку файлов, визуальный редактор, простоту в создании и администрировании сайта, удобство панели администратора – реализация внутренней навигации по сайту, с чем приходится постоянно сталкиваться разработчику;
- гибкость системы (многоязычный контент и многосайтовость, обновление версий с учетом новых тенденций развития);
- включенный функционал (статистика посещений; управление стилями и шаблонами; формы обратной связи; опросы и голосования; наличие форума и блога; интернет-магазины и платежные системы, встроенные виджеты)[2-4].

Создание сайтов. Для проверки возможностей CMS были разработаны четыре сайта на платформах: WordPress, Ucoz, Wix и Mozello. На рис.2. показаны сайты, созданные в Wordpress[5](левая часть рисунка) и Mozello (правая часть) [6].



Рисунок 2 – Сайты, созданные на платформах WordPress и Mozello
Платформа Mozello пока не слишком популярна, поэтому на рис.3 показаны стадии разработки сайта на ее основе.



Рисунок 3 – Панель администратора Mozello

1- визуальный редактор (элементы: страницы, дизайн, адрес, настройки, просмотр); 2 - текстовый редактор (возможности форматирования, добавление ссылок и графических изображений); 3 – определение типа новой страницы.

На рис. 4 показаны сайты, созданные в Ucoz (левая часть [7]) и Wix (справа [8]). Платформа Wix ценится в первую очередь как платформа с большим количеством модных, современных шаблонов, поэтому чаще всего дизайнеры рекомендуют платформу Wix для создания визиток с креативным дизайном.

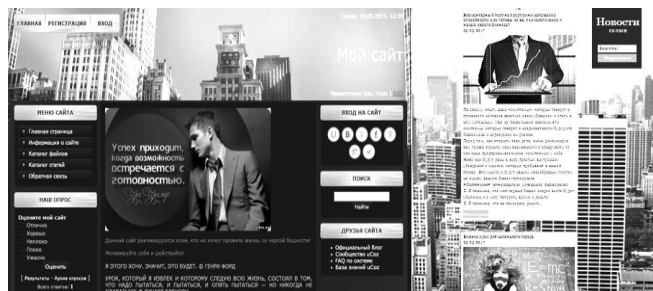


Рисунок 4 – Сайты, созданные на платформах Ucoz и Wix

Проанализировав платформы, нами были выделены преимущества и недостатков систем. К преимуществам следует отнести: **Mozello** (простота, доступность в управлении и создании сайтов, сервис на русском языке, 48 бесплатных шаблонов); **WordPress** (простота, доступность в управлении и

создании сайтов, сервис на русском языке, 165 бесплатных шаблонов, стабильность плагинов, дополнений и модулей, возможность редактирования HTML кода); **Wix** (хорошо переведенный на русский язык интерфейс, более 1000 бесплатных шаблонов, безупречный внешний вид, современный дизайн, возможность входа через социальные сети, мобильный вариант сайта); **Ucoz** (многоструктурное меню, наличие статистики, полностью бесплатные шаблоны, большое количество настроек).

К недостаткам следует отнести: **Mozello** (минимальные возможности форматирования текста, сложности с добавлением графики, CSS стили доступны только в платном тарифе); **WordPress** (весомая нагрузка на сервер, возможна низкая скорость загрузки сайта); **Wix** (медленно грузится и работает, некачественная помощь по освоению сайта, реклама); **Ucoz** (сложен для новичка, усложненная структура панели управления сайтом, навязчивая реклама некачественные, «серые» шаблоны).

Выводы. Рассмотрены различные системы управления контентом сайта CMS. Проведенный анализ существующих систем, их рейтинга, предъявляемых к системам требований, позволил выделить две системы, которые можно рекомендовать обычному пользователю для разработки собственного сайта. Для создания сайта-визитки можно использовать WordPress, т.к. система проста, имеет широкий функционал, удобство администрирования, большое количество бесплатных шаблонов, виджетов и гаджетов. WordPress можно определить как классику. Если необходим модный, стильный, современный сайт с безупречным внешним видом, то этим требованиям отвечает Wix.

Список литературы

1. Рейтинг CMS [электронный ресурс] // Компания iTrack[сайт]. URL: <https://itrack.ru/research/cmsrate/>
2. Давлетбаева Р.Р. Сравнительный анализ CMS: выбор оптимальной системы для создания общеобразовательных сайтов // Научное сообщество студентов XXI столетия. Технические науки: сб. ст. по мат. XXXIX междунар. студ. науч.-практ. конф. № 2(39). URL: [http://sibac.info/archive/technic/2\(38\).pdf](http://sibac.info/archive/technic/2(38).pdf)
3. Рейтинг Лучший конструктор сайтов Рунета[электронный ресурс]. URL: <https://itrack.ru/research/cmsrate/>
4. Создание сайта на CMS: что это такое и какой выбрать? // Seostop.ru - обучающие материалы по созданию и продвижению сайтов и заработку в интернете URL: <http://www.seostop.ru/sozдание-saita/html-css/cms.html>
5. Сайт, созданный Саксон А.В. в Wordpress//URL: <https://handmade58.wordpress.com>
6. Сайт, созданный Саксон А.В. в Mozello//URL: <http://handmade-4.mozello.ru>
7. Сайт, созданный Шевцовой А.Д. в Ucoz //URL: <http://hasher.ucoz.net/publ/>
8. Сайт, созданный Шевцовой А.Д. в Wix//URL: <https://schevtsovaanastasi.wixsite.com/mysite>

УДК 517.984.46

САМОСОПРЯЖЕННОСТЬ МАТРИЧНОГО ОПЕРАТОРА ДИРАКА С ТОЧЕЧНЫМИ ВЗАИМОДЕЙСТВИЯМИ

В.С. Будька

ГП «Институт прикладной математики и механики», г. Донецк

budyka.vik@gmail.com

Аннотация

Будька В.С. Самосопряженность матричного оператора Дирака с точечными взаимодействиями. Исследованы некоторые спектральные свойства матричного оператора Дирака с точечными взаимодействиями.

Ключевые слова: индексы дефекта, якобиева матрица, самосопряженность оператора.

Постановка проблемы. В работах [1] и [2] были исследованы спектральные свойства одномерного оператора Дирака с точечными взаимодействиями. Данная работа расширена на матричный случай $2p \times 2p$.

Рассмотрим одномерное дифференциальное выражение Дирака

$$D = -ic \frac{d}{dx} \otimes \sigma_1 + \frac{c^2}{2} \otimes \sigma_3 = \begin{pmatrix} c^2/2 & -ic \frac{d}{dx} \\ -ic \frac{d}{dx} & -c^2/2 \end{pmatrix} \otimes I_p, \quad (1)$$

в котором $c > 0$ - скорость света, σ_1 и σ_3 - $2p \times 2p$ - матрицы Паули

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes I_p, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes I_p \in C^{2p \times 2p},$$

где I_p - единичная матрица порядка $p \times p$.

Пусть $K = (a, b)$, $-\infty < a < b \leq +\infty$. Пусть также $\alpha_n = \alpha_n^* \in C^{p \times p}$. Также предположим, что $X = \{x_n\}_{n=0}^\infty$ - дискретное подмножество интервала K , $x_{n-1} < x_n \in N$. Положим $d_n := x_n - x_{n-1}$.

Всюду в дальнейшем $\hat{f} = \{f_1, f_2, \dots, f_{2p}\}^T$ - вектор-столбец $\hat{f} = \begin{pmatrix} \hat{f}_1 \\ \hat{f}_2 \end{pmatrix}$,

где

$$\hat{f}_1 = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_p \end{pmatrix} = \{f_1, f_2, \dots, f_p\}^T, \quad \hat{f}_2 = \begin{pmatrix} f_{p+1} \\ f_{p+2} \\ \dots \\ f_{2p} \end{pmatrix} = \{f_{p+1}, f_{p+2}, \dots, f_{2p}\}^T.$$

Здесь T - операция транспонирования.

В пространстве $L^2(K, C^{2p}) = L^2(K) \otimes C^{2p}$ с выражением (1) ассоциировано семейство (незамкнутых) симметрических операторов $D_{X,\alpha}^0$:

$$D_{X,\alpha}^0 = D, \quad \text{dom}(D_{X,\alpha}^0) = \{f \in W_{comp}^{1,2}(K \setminus X) \otimes C^{2p} : \hat{f}_1 \in AC_{loc}(K), \\ \hat{f}_2 \in AC_{loc}(K \setminus X); \hat{f}_2(a+) = 0, \hat{f}_2(x_n+) - \hat{f}_1(x_n-) = -\frac{i\alpha_n}{c} \hat{f}_1(x_n), n \in N\}.$$

Обозначим $D_{X,\alpha} = \overline{D_{X,\alpha}^0}$ - замыкание оператора $D_{X,\alpha}^0$.

Рассмотрим якобиеву матрицу

$$B_{X,\alpha} = \begin{pmatrix} O_p & -\frac{\nu(d_1)}{d_1^2} I_p & O_p & \dots \\ -\frac{\nu(d_1)}{d_1^2} I_p & -\frac{\nu(d_1)}{d_1^2} I_p & \frac{\nu(d_1)}{d_1^{3/2} d_2^{1/2}} I_p & \dots \\ O_p & \frac{\nu(d_1)}{d_1^{3/2} d_2^{1/2}} I_p & \frac{\alpha_1}{d_2} I_p & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix},$$

где O_p - нулевая матрица порядка $p \times p$.

Теорема 1. Пусть K - бесконечный интервал, т.е. или $K = \mathbb{R}_\pm$, или $K = \mathbb{R}$, а также $\alpha_n = \alpha_n^* \in C^{p \times p}$. Тогда для индексов дефекта оператора $D_{X,\alpha}$ справедливо соотношение

$$n_\pm(D_{X,\alpha}) = n_\pm(B_{X,\alpha}) \leq p.$$

Также оператор $D_{X,\alpha}$ - самосопряженный.

Список литературы

1. R. Carlone, M/ Malamud, A. Posilicano, On the spectral theory of Gestezy-Seba realizations of 1-D Dirac operators with point interactions on a discrete set, J. Diff. Eq. 254 (2013) 3835-3902.
2. F. Gestezy, P. Seba, New analytically solvable models of relativistic point interactions, Lett. Math. Phys. 13 (1987) 345-358.

УДК 622.14

ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ДЛЯ ФОРМИРОВАНИЯ ТЕХНОГЕННОГО МЕСТОРОЖДЕНИЯ

Е.В.Прокопенко¹, С.В.Масло², Е.В.Пахомов³

Донецкий национальный технический университет, г. Донецк
кафедра прикладной математики
доцент кафедры прикладной математики¹, ассистент кафедры прикладной
математики, студент группы Шск-16
prokopenko1515@rambler.ru

Аннотация

Прокопенко Е.В., Масло С.В., Пахомов Е.В. Применение линейного программирования для формирования техногенного месторождения. В статье рассмотрены вопросы применения линейного программирования для нахождения максимального компонента среди работающих пластов в результате чего имеется возможность построить хронолитологическую модель данного месторождения. Данная модель позволит выявить экологически опасные зоны в пределах месторождения и дать рекомендации для дальнейшего использования месторождения.

Ключевые слова: техногенное месторождение, компонент, хронолитологическая модель.

В отвалах угольных шахт много запасов некоторых металлов, соизмеримых по объему с природными месторождениями полезных ископаемых, получение которых для Украины будет экономически выгодным, тем более что в настоящее время многие из минеральных ресурсов уже исчерпаны, что является одной из глобальных проблем. Порода попадает на породный отвал не хаотически, а в определенных объемах и последовательности. Основой для этого могут служить маркшейдерская горно-графическая документация, которая, по сути, является единым достоверным источником информации за весь период эксплуатации шахты, и, следовательно, за весь период отсыпки породы на отвал.[1]

Технология формирования породного отвала заключается в насыпке породы на отвал отдельными ярусами. Так как отдельный ярус породного отвала насыпается определенное количество лет, то можно составить карту насыпки отвала за определенный период времени. Так как каждый пласт имеет свое геологическое строение, то можно составить прогноз тех химических реакций, которые произойдут при соприкосновении тех или иных

элементов, содержащихся в различных пластах, то есть заранее выявить неблагоприятные зоны на отдельном ярусе и в целом на отвале. [2]

Используя выше сказанное, можно сформулировать задачу о нахождении максимального компонента среди состава пород, входящих в пласт, а также сформировать вектор, позволяющий оценить объем выдачи на отвал породы, содержащий максимальный компонент по пластам. [3]

С точки зрения линейного программирования, данная задача формулируется следующим образом.

Существует функция

$$f(v_1, v_2, \dots, v_m), \quad (1)$$

где

$$V_1 = (x_1^{(1)}, x_2^{(1)}, \dots, x_j^{(1)}, \dots, x_k^{(1)}) \quad g_1(x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(i)}, \dots, x_1^{(m)}) \leq b_1$$

$$V_2 = (x_1^{(2)}, x_2^{(2)}, \dots, x_j^{(2)}, \dots, x_k^{(2)}) \quad g_2(x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(i)}, \dots, x_2^{(m)}) \leq b_2$$

... при ограничениях ...

$$V_i = (x_1^{(i)}, x_2^{(i)}, \dots, x_j^{(i)}, \dots, x_k^{(i)}) \quad g_j(x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(i)}, \dots, x_j^{(m)}) \leq b_j$$

... где ...

$$V_m = (x_1^{(m)}, x_2^{(m)}, \dots, x_j^{(m)}, \dots, x_k^{(m)}) \quad g_k(x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i)}, \dots, x_k^{(m)}) \leq b_k$$

$f(v_1, v_2, \dots, v_m)$ - целевая функция

или критерий эффективности. В данном случае, вектор f характеризует значение максимального объема по пластам в зависимости от максимального компонента в составе пород пласта, $X = (x_1, x_2, \dots, x_j, \dots, x_k)$ - варьируемые параметры, которые обозначают соответствующую компоненту в составе пород пласта,

$$\begin{aligned} g_1(x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(i)}, \dots, x_1^{(m)}) &\leq b_1 \\ g_k(x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i)}, \dots, x_k^{(m)}) &\leq b_k \end{aligned} \quad (2)$$

- функции, которые задают ограничения на компоненты, входящие в состав пород пласта.

Таким образом, используя данные выше приведенных функций, можно составить матрицу размерностью $[1..k, 1..m]$, где k -количество компонент, m -количество пластов. Данная матрица будет иметь вид, представленный ниже.

$$\begin{array}{cccccc}
 V_1 & V_2 & \dots & V_i & \dots & V_m \\
 x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(i)} & \dots & x_1^{(m)} \\
 x_2^{(1)} & x_2^{(2)} & \dots & x_2^i & \dots & x_2^m \quad (3) \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 x_j^{(1)} & x_j^{(2)} & \dots & x_j^{(i)} & \dots & x_j^{(m)} \\
 \text{Для} & & & & & \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 x_k^{(1)} & x_k^{(2)} & \dots & x_k^{(i)} & \dots & x_k^{(m)}
 \end{array}$$

нахождения вектора $V = (V^{(1)}, V^{(2)}, \dots, V^{(m)})$, характеризующего максимальное значение объемов по пластам в зависимости от химического компонента, необходимо вычислить параметры, входящие в вектор, по формуле

$$V^{(l)} = \sum_{i=1}^m x_l^{(i)} \cdot V_i \quad (4)$$

Используя найденные объемы, можно составить сделать вывод о составе компонент, входящих в пласт, то есть, какие компоненты больше содержатся в объеме породы, отсыпанной на отвал, в зависимости от графика проведения подготовительных выработок. На рисунке 1 показан процесс нахождения такого компонента и дальнейшее его применение для формирования техногенного месторождения (рис.2).

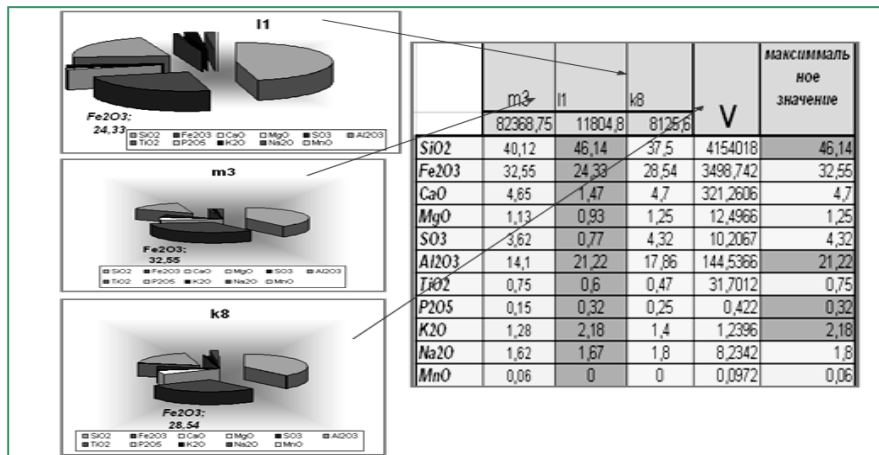
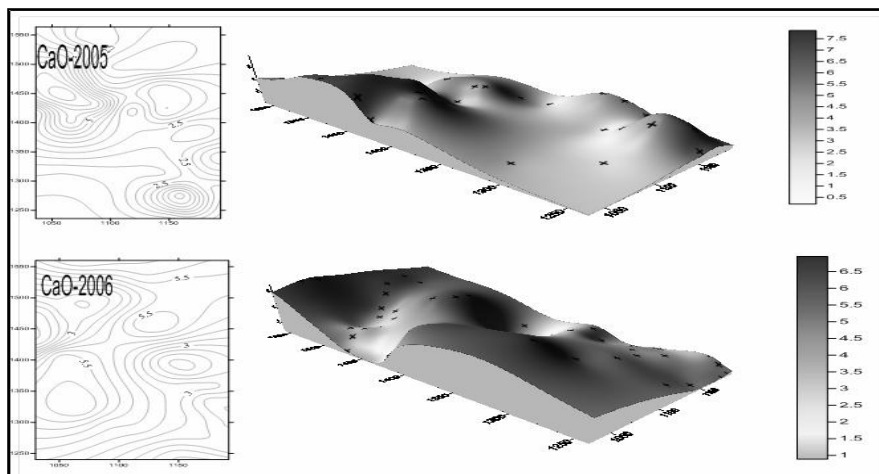


Рис.1. Нахождение максимального компонента среди работающих пластов



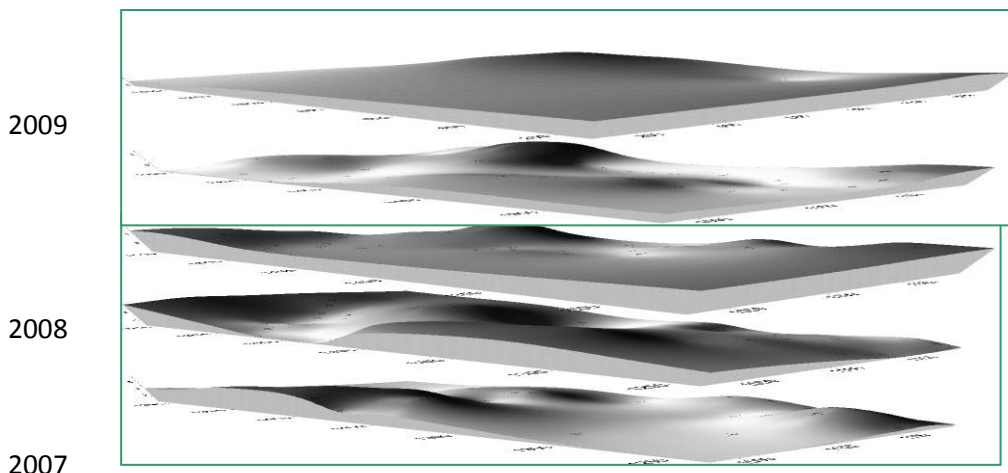


Рис.2. Построение поверхности техногенного месторождения в зависимости от найденного максимального

Таким образом, зная, что происходит с насыпкой породы в пределах одного пласта, и, зная какие работают пласты, то ли одновременно, то ли каждый по отдельности, можно составить хронологическую модель насыпки породы, в результате которой могут быть выявлены места с накоплением вредных веществ. По данной модели можно составить рекомендации контроля за данными веществами, а также дать рекомендации по дальнейшему использованию этих веществ.

ЛИТЕРАТУРА

1. Разработка динамической модели породных отвалов угольных шахт/ Прокопенко Е.В., Борщевский С.В.//Наукові праці УкрНДМІ НАН України. Випуск 6 /Під заг.ред. А.В.Анциферова.-Донецьк, УкрНДМІ НАН України, 2010.-14с.
2. Вахмянин И.С. Разработка модели алгоритма управления информационными потоками в ситуационных центрах органов государственной власти / Вахмянин И.С., Ильин Н.И., Новикова Е.В.—Бизнес-Информатика №1(15)- 2011.—С. 3—9.
3. Прокопенко Е.В. Построение пространственной модели участков ярусов породного отвала с использованием программного пакета Surfer /С.В.Борщевский, Е.В.Прокопенко// Збірник наукових праць НГУ. — Національний гірничий університет, 2010. —№34, т.1.— С. 82—87.

УДК 004.0-517.8- 621.3

РЕФЛЕКСИВНАЯ РЕФЕРЕНЦИЯ И АНАЛИЗ КВАЙНОВ

А.И. Андрухин, В.А. Полетаев

Донецкий национальный технический университет, г. Донецк,
alexandruckin@ramber.ru

Аннотация:

Андрухин А.И., Полетаев В.А. Рефлексивная референция и анализ квайнов. В работе рассматривается известная задача построения программы, которая сама себя распечатывает. Построены расширения этой проблемы. Реализованы программы, которые печатают заданное число копий собственного кода, копию себя с задаваемыми суффиксами и префиксами. Программы построены на языке Python. Выполнен историко-философский анализ семиотического соотношения имени объекта и самого объекта. Рассмотрена связь задачи программы, которая сама себя распечатывает с известными семиотическими проблемами и концепциями. В частности, показана тесная связь структуры таких программ с диадой Пирса.

Ключевые слова: рефлексия, имя, референция, семиотика, самосознание

Введение. Естественным и истинным представляется утверждение, что мышление не существует без слов и слово, в частности слово (имя), является необходимым и высшим результатом мысли. Но является справедливым и утверждение, что человек может мыслить и без употребления слов и так называемое бессловесное мышление есть необходимый этап в его развитии. Более того, бессловесное мышление зачастую понимается как переход на высший уровень мысли и преодоления словесной ступени.

Это прекрасно выражено в известной притче, которая отражает соотношения смысла и слова: «Силками пользуются для ловли зайцев. Поймав зайца, забывают про силки. Словами пользуются, чтобы внушить смысл. Внушив смысл, забывают про слова. Где мне найти человека, забывшего слова, чтобы перекинуться с ним словом?»

Основным актом интеллектуальной системы есть операция «дать имя» и естественно является базовой в деятельности программиста.

Без рефлексивных механизмов понимания невозможен полный анализ объектов. Инструменты рефлексии мы должны применять для повышения уровня понимания. Простейший рефлексивный механизм работает, как диада двух уровней знания: предметного знания и знания о предметном знании («я знаю, что я знаю...»).

Согласно [1], «Когито» («cogito»), означающее, как известно, «мыслю», включает в себе нечто большее, чем указание на познавательный акт, символизируемый записью $S \rightarrow O$; в акте «когито» присутствует момент самонаправленности ($S \rightarrow S$): «Я мыслю себя мыслящим», или «Я мыслю: «Я мыслю»».

Рефлексивное понимание в своем одном действии использует одно множество актов для репрезентации самого объекта, другое множество для репрезентации знаний об объекте. Учет рефлексивности принципиально усложняет познавательные процедуры, однако без соответствующих метаописаний, связанных, с рефлексивными конструкциями, очевидно не выйти из поверхности Мебиуса наблюдаемых объектов науки.

ЕЯ как организм и гипотеза лингвистической относительности Сепира-Уорфа. В настоящее время естественный язык рассматривается как некоторый живой организм, который возникнув под действием определенных, нам до конца неясных причин начинает самостоятельно развиваться, и оказывает решающее влияние на поведение и интеллект человека[2].

Простейшая формулировка гипотезы Сепира-Уорфа выглядит так: «Мышление человека обусловлено языком, на котором он говорит».

Так утверждают, что физические и математические конструкции произведений Ньютон определяются латинским и старофранцузским языками, на которых они описаны. Существует племя североамериканских индейцев «хоппи», конструкции языка которых наиболее приспособлены для выражения идей теории относительности Эйнштейна[3].

В настоящее время существуют различные группы специалистов, которые разделяют или не поддерживают истинность гипотезы Сепира-Уорфа по отношению к процессам восприятия или мышления человеком.

Историко-философский анализ понятия имени. В античности считали, что слово – имя вещи и сами вещи связаны единственным способом и поэтому развивалось учение о таинственных и магических свойствах имен.

Так Платон считал, что звуки языка могут представить объект или класс объектов и применяя соответствующие комбинации звуков можно добиться полного сходства с объектом и полного управления им.

Приведем абзац из диалога «Кратил» «Сократ. Да ведь смешные вещи, Кратил, творились бы с именами и вещами, которым принадлежат эти имена, если бы они были во всем друг другу тождественны. Тогда все бы словно раздвоилось, и никто не мог бы сказать, где он сам, а где его имя»[4].

Кант значительно позже подчеркнул идею условной связи имени(знака) с тем, что оно означает и указал, что отождествление знаков и объектов приводит к приписыванию знакам существования в области собственной

реальности. Тем самым резко меняется модель мира и поэтому мистическое знание базируется на неправомерном смешивании объектов и знаков, знаковых операций и реальных действий.

Кант показал, что высказывание лишь имя (символ) для рефлексии, так как в нем нет непосредственного отображения объекта и фокус рефлексии направлен на имя (понятие), которое не является объектом сенсорного восприятия (созерцания), а не на сам объект: «...все наше познание о боге только символическое; и тот, кто берет его схематически с такими свойствами, как рассудок, воля и т.д., которые доказывают свою объективную реальность лишь на существах в мире, впадает в антропоморфизм...»[5].

До сих пор интересен вопрос о произвольности имен и приведем мнения мыслителей прошлого.

Платон: «Давать имена нужно в соответствии с природой, а не так как заблагорассудится.»[6]

Гоббс: «Имя есть слово, произвольно выбранное в качестве метки с целью возбуждения в уме мыслей, сходных с прежними мыслями.

Имя служит признаком того, какие мысли были в уме говорящего и каких не было. Я считаю возникновение имен результатом произвола.»[7]

Локк: «Люди способны делать звуки знаками идей или мыслей. У людей есть способность пользоваться этими звуками как знаками внутреннего представления и замещать ими идеи своей души с тем, чтобы люди могли делать известными другим людям свои идеи, свои мысли.»[8]

Кант также объединял проблему знака с познанием и в частности, он подчеркивал факт связи настоящего с прошедшим через обозначение и следовательно познание настоящего через знаковую систему.

Высказывание Канта: «Мыслить-значит говорить с самим собой...значит внутренне...слышать себя самого » есть сущность сегодняшней модели существования обратной связи(рефлексии) в говорящем. Сама рефлексия контролирует мысли говорящего и его отношение к именам(знакам) согласно правилам среды существования его же.

Исследование логики имени привело Лосева А.Ф. к описанию 67 его атрибутов[9].

Он дает сначала самое простое и общее определение слова, что «слово, или имя, есть смысл, или понимаемая, понимаемая сущность»[9]. Развивая и уточняя это определение, получаем следуя Лосеву определение «имя есть смысловая энергия сущности предмета».

Более сложное определение имени согласно Лосева обнаруживает в себе рефлексивную связь: «имя есть...смысловая встреча субъекта с его предметом;короче: имя есть энергично выраженная умно-символическая стихия мифа, осмыслившая собой то или другое инобытие и тем приведшая его к встрече с самим собою:...»

Поиски универсальных языков. Идея универсального языка часто связывается с религиозным представлением о золотом веке человечества. Существуют современные исследования, которые пытаются реконструировать единый праязык.

Люди часто занимались проектированием универсального языка. Здесь мы укажем на римского врача Галена (130 г. н. э), Уильямс, дель Карно в конце 17в и др. К примеру, Уильямс: в «Эссе о реальной земле и философской земле» выделил 6 родов ЕЯ:

- 1)Трансцендентальные понятия(бог, природа)
- 2)Субстанции 3)Количества 4)Качества 5)Действия 6)Отношения

Таким образом этими категориями, он пытался представить полную энциклопедию человеческого знания.

Этот подход до сих пор представляет интерес для КИС, так как по ключу объекта можно определить часть информации об объекте.

Широкому распространению эсперанто в настоящее препятствуют по различным причинам (политическим, экономическим и т. д.)

Семиотический треугольник. Соотношения между знаком, субъектом и объектом представлен на рис. 1.[10]

Следуя этой парадигме, предложение ЕЯ можно понимать, как сложное имя ситуации.



Предложение для европейского человека (Индоевропейская семья) обычно состоит из 3-х различных элементов: главными являются 2 (рис 2).

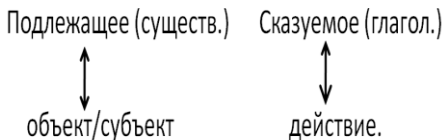


Рисунок 2 – Стандартной сопоставление частей предложения и выделяемых феноменов.

Следовательно, наша сетка восприятия мира или категоризация действительности триочна.

Система MARGIE для исследования ЕЯ. Появление этой системы оказала и оказывает большое влияние на решение проблемы понимания текста [11,12].

Обработка ЕЯ происходит в 3-х аспектах: отражение предложений естественного языка в их смысловое представление, хранение смыслового представления в памяти компьютера и осуществление умозаключений относительно полученного смысла, перевод смыслового представления на естественный язык.

Создатели этой системы (группа психологов, филологов, программистов) считали справедливым идею Хомского: «существует некоторый язык, язык посредник, который является врожденным для представителя человеческого рода». ЯП (язык-посредник) – это представление смыслов в предложениях ЕЯ1 и ЕЯ2.

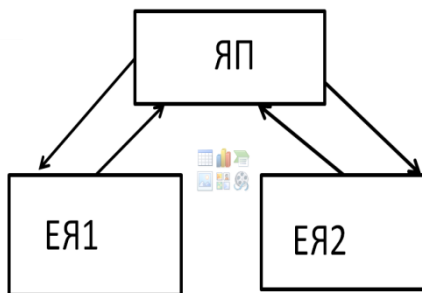


Рисунок 3 – Положение Н.Хомского.

К сожалению, в этой системе мало уделено внимание рефлексивной референции, хотя косвенно к ней можно отнести обратные умозаключения (рис.4). Это очевидно связано с основной целью системы: отображению и обработке разговорной речи.

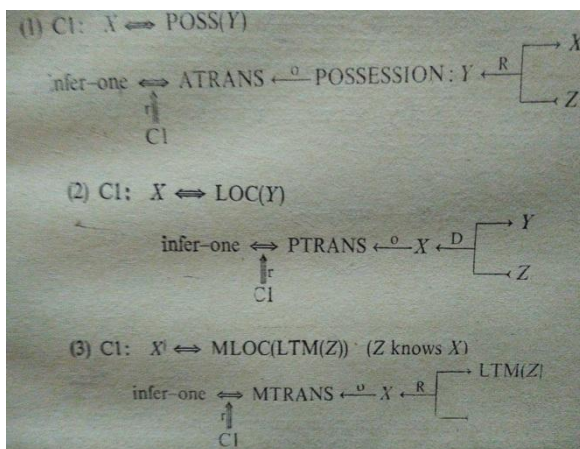


Рисунок 4 – Варианты обратного умозаключения в системе концептуальной обработки Margie.

Анализ квайна. Требования известной уже много десятилетий задачи построения программы [13], которая сама себя распечатывает можно изобразить согласно рис.5 для языка программирования Python.

PP



Рисунок 5 – Р имя программы-воздействия P.

Построены расширения этой проблемы. Реализованы программы, которые печатают заданное число копий собственного кода ,копию себя с задаваемыми суффиксами и префиксами.

Решение этой задачи представлено своей структурно-функциональной основой на языке Pythonна рис .6. Программный код этих решений на языке Python представлен на рис.7-9. Можно выделить несколько основных положений при ее решении. Длина всей программы Pравна сумме длин строк $q, S, PrintP_s$. Здесь q определяет символ (имя) начала и конца имен (символьных констант) в языке Python. Имя Sпредставляет собой указатель для символьной строки, которая является основой для построения имени всей программы Pна

основе указателей s_1, s_2, \dots , и q , т.е. $P = P_s(q, s_1, s_2, \dots)$. Подчеркнем, что S включает в себя $\text{Print} P_s$.

Мы обнаруживаем, здесь образцовый случай упорядоченной пары, т.е. пример, когда говорим о двух объектах одновременно, как будто это два объекта одного вида, рассматриваемые в один и тот же момент времени как один.

Под упорядоченной парой мы понимаем по Пирсу диадку, т.е. ментальную диаграмму, состоящая из двух образов, двух объектов, один из которых экзистенциально связан с одним членом пары, а другой с другим; Один присоединяет к объекту в качестве его обозначения символ, значение которого первый, а другой символ, значение которого второй[14].

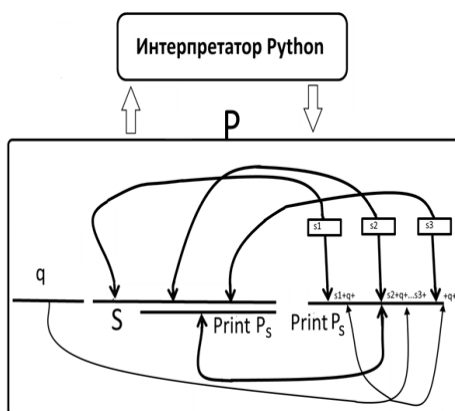


Рисунок 6 – Структура самораспечатывающей программы.

На рис. 6 мы видим несколько обратных связей.

Иные варианты решения этой задачи на разных языках программирования представлены в[2], в том числе и переход на собственное представление на плоскости, что демонстрирует формула Таппера.

Формула Таппера. Формула была опубликована в 2001 году в работе Джеффа Таппера, посвященной разработанной им программе рисования графиков GrafEqxi создаёт собственное изображение на плоскости[15]. Формула представлена на рис.11.

$$\frac{1}{2} < \left[\text{mod} \left(\left\lfloor \frac{y}{17} \right\rfloor 2^{-17[x]} - \text{mod}(\lfloor y \rfloor, 17), 2 \right) \right]$$

Рисунок 11 – Формула Таппера[15]

Сама формула имеет назначение для декодирования растровых изображений, которые представлены двоичным числом k . Формулу можно использовать для отображения различных изображений.

$$\frac{1}{2} \left\lfloor \text{mod} \left(\left(\frac{y}{17} \right) z^{-17} [x]^{-\text{mod}(y, 17)}, z \right) \right\rfloor$$

Рисунок 12 – Результат работы формулы Таппера (область 1696x272 пикселей).

Отметим, что k представляет собой простой монохромный растр, который представляется, как двоичное число, умноженное на 17.

Младший бит от деления k на 17 будет соответствовать верхнему правому углу; 17 младших битов будут соответствовать правой колонке пикселей; следующие 17 младших битов будут определять вторую правую колонку и т. д.

```

k = 960 939 379 918 958 884 971
672 962 127 852 754 715 004 339 660 129 306 651 505 519 271 702 802 395 266
424 689 642 842 174 350 718 121 267 153 782 770 623 355 993 237 280 874 144
307 891 325 963 941 337 723 487 857 735 749 823 926 629 715 517 173 716 995
165 232 890 538 221 612 403 238 855 866 184 013 235 585 136 048 828 693 337
902 491 454 229 288 667 081 096 184 496 091 705 183 454 067 827 731 551 705
405 381 627 380 967 602 565 625 016 981 482 083 418 783 163 849 115 590 225
610 003 652 351 370 343 874 461 848 378 737 238 198 224 849 863 465 033 159
410 054 974 700 593 138 339 226 497 249 461 751 545 728 366 702 369 745 461
014 655 997 933 798 537 483 143 786 841 806 593 422 227 898 388 722 980 000
748 404 719 .

```

Рисунок 13 – Значение k [15].

Заключение. В работе рассматривается известная задача построения программы, которая сама себя распечатывает. Выполнен историко-философский анализ соотношения имени объекта и самого объекта.

Показана связь задачи программ, которая сама себя распечатывают с известными семиотическими проблемами и концепциями.

В частности, имеем тесную связь таких программ с диадой Пирса.

Построены расширения этой проблемы.

Реализованы программы, которые печатают заданное число копий собственного кода, копию себя с задаваемыми суффиксами и префиксами.

Программы построены на языке Python.

```
q=""';s=q=""';s=;print(s[:3]+q+s[3:7]+q+s+q+s[7:]);print(s[:3]+q+s[3:7]+q+s+q+s[7:])
```

а)

```
>>>
```

```
====RESTART: C:\MY_LYBRARY_ALL\SCIENCE\ARTICLES\2017\quine_N7\quain0.py====
```

```
q=""';s=q=""';s=;print(s[:3]+q+s[3:7]+q+s+q+s[7:]);print(s[:3]+q+s[3:7]+q+s+q+s[7:])
```

```
>>>
```

б)

Рисунок 7 – Python-программа самораспечатки а) код; б) результат.

```
q=""';s=q=""';s=;print(s[:3]+q+s[3:7]+q+s+q+s[7:]);Some string';print(s[:3]+q+s[3:7]+q+s+q+s[7:])
```

а)

```
>>>
```

```
RESTART: C:\MY_LYBRARY_ALL\SCIENCE\ARTICLES\2017\quine_N7\quine_with_string.py
```

```
q=""';s=q=""';s=;print(s[:3]+q+s[3:7]+q+s+q+s[7:]);Some string';print(s[:3]+q+s[3:7]+q+s+q+s[7:]);Some string
```

б)

Рисунок 8 – Python-программа самораспечатки с суффиксом: а) код; б) результат.

```
q=""',m="Some string ";s=q=""',m=""';s=;print(m+s[:3]+q+s[3:8]+m+s[8:12]+q+s+q+s[12:]);print(m+s[:3]+q+s[3:8]+m+s[8:12]+q+s+q+s[12:])
```

а)

```
>>>
```

```
RESTART: C:\MY_LYBRARY_ALL\SCIENCE\ARTICLES\2017\quine_N7\quine_string_begining.py
```

```
Some sting q=""',m="Some sting ";s=q=""',m=""';s=;print(m+s[:3]+q+s[3:8]+m+s[8:12]+q+s+q+s[12:]);print(m+s[:3]+q+s[3:8]+m+s[8:12]+q+s+q+s[12:])
```

б)

Рисунок 9 – Python-программа самораспечатки с префиксом: а) код; б) результат

```
q=""';s=q=""';s=;print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3);print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3)
```

а)

```
>>>
```

```
RESTART: C:\MY_LYBRARY_ALL\SCIENCE\ARTICLES\2017\quine_N7\quine_multiple_times.py
```

```
q=""';s=q=""';s=;print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3);print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3)
```

```
q=""';s=q=""';s=;print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3);print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3)
```

```
q=""';s=q=""';s=;print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3);print((s[:3]+q+s[3:7]+q+s[:6]+s[-7:]+q+s[7:]+'\n')*3)
```

б)

Рисунок 10 – Python-программа самораспечатки с копиями а) код; б) результат.

Литература

1. В.А.Петровский.К построению алгебры когито:опыт игры в бисер//Модели мира.-М.:Российская ассоциация искусственного интеллекта, 1997-240 с. 2.Налимов В.В.Вероятностная модель языка. — М.: Наука, 1979. — 303 с.
- 3.Б.Л.Уорф.Отношение норм поведения и мышления к языку//Языки как образ мира.-М.:ООО «Издательство АСТ», 2003.-568 с.
- 4.Платон.Кратил. Собр. соч. в 4-х томах. Том 1. М.: "Мысль", 1990
- 5.Кант И.Антропология с прагматической точки зрения. М.:Наука,2002
6. www.nyx.net/~gthomps/quine.htm
7. Гоббс Т.О человеке// Сочинения в 2 т. Т.1. М. Мысль, 1989.- 622с.
8. Локк Дж.Опыт о человеческом разуме. Сочинения: В 3-х т. Т. 1 / Под ред. И. С. Нарского. — М.: Мысль, 1985. — 623 с
9. Лосев А.Ф. Философия имени.-М.: Издательство Московского университета,1990.-с.269.
10. Г.Фреге.Мысль: логическое исследование //Философия, логика, язык.- М.:Прогресс,1987.-с.18-47.
11. Шенк Р.Обработка концептуальной информации.Пер. с англ.- М.Энергия,1980.-360 с.
- 12.Минский М. Фреймы для представления знаний. М.: Мир, 1979.
- 13.Уэзерелл Ч. Этюды для программистов: Пер. с англ. - М.: Мир, 1982. - 288 с
- 14.Пирс Ч. С. Принципы философии. Том I. - СПб.: Санкт-Петербургское философское общество, 2001. - 224 С.
- 15.Tupper, Jeff. «Reliable Two-Dimensional Graphing Methods for Mathematical Formulae with Two Free Variables» http://www.dgp.toronto.edu/people/mooncake/papers/SIGGRAPH2001_Tupper.pdf

УДК 004.92

ЖИЗНЕННЫЙ ЦИКЛ ГРУПП В СОЦИАЛЬНЫХ СЕТЯХ

Е.В. Рощина¹, И.Ю. Анохина²,

1-студентка гр. ПМск.-15; 2- доцент кафедры прикладной математики,
Донецкий национальный технический университет, г. Донецк
кафедра прикладной математики
IngaTula@mail.ru

Аннотация.

Рощина Е.В., Анохина И.Ю. Жизненный цикл групп в социальных сетях. Выполнен сравнительный анализ жизненных циклов социально-политических и бытовых групп в Контакте.

Ключевые слова: социальные сети, группы в Контакте, жизненный цикл, активность подписчиков.

Введение. Социальные сети являются одними из самых посещаемых ресурсов Интернета. По данным Mediascope [1], аудитория в 2016 - марте 2017г. достигла 87 млн человек, что составило 71% от всего населения России, причем за год российская интернет-аудитория увеличилась на 2%. По данным GfK (Gesellschaft für Konsumforschung) Group 97% молодых россиян (в возрасте от 18 до 29лет) образуют интернет – аудиторию [2], при этом раз в сутки его используют 48.7% россиян.

В истории социальных сетей можно выделить три этапа. Первый – социальные сети 90-х годов. В 1995г. был разработан портал Classmates.com с простейшим функционалом; в 1999г. открыт сервис Livejournal.com, ставший первым социальным сервисом, популярным в России.

Второй этап- создание социальных сетей с широким функционалом для базового взаимодействия. В 2003—2004г.г. запущены MySpace и Facebook, что послужило стартом для бума социальных сетей; появились социальные сети, ориентированные на создание и поддержание деловых контактов, так называемые сети бизнес-направления, кА пример, LinkedIn.

Третий этап – В 2004 году Марком Цукербергом был запущен портал Facebook. Американский программист Джек Дорси запустил в 2006 г. проект Twitter. В том же 2006 г. появились «Одноклассники» (автор Альберт Попков) и проект «В контакте» (автор Павел Дуров). Появились социальные сети, решающие конкретные проблемы: поиск сотрудников (бизнес-сети), игры (игровые сети), поиск информации (контент-сети), корпоративные, коммуникативные и т.д.

Сегодня сеть «В контакте» является самым крупным в русскоязычном сегменте Интернета сетевым ресурсом. С технической точки зрения,

социальная сеть – интерактивный многопользовательский web-сайт, контент которого наполняется самими участниками сети. Социальные сети являются не только средством для общения, но и мощным инструментом воздействия как с точки зрения маркетинга, так и с точки зрения воздействия на личность в социальном, политическом смысле.

Постановка проблемы. Цель статьи – провести анализ возможных методов аналитики социальных сетей, выявить существующие закономерности на примере развития групп в Контакте, оценить отличия коммуникативных групп с социальным уклоном от обычных, бытовых или бизнес-групп. Такие исследования позволят осуществлять несколько функций: мониторинг общественного мнения о деятельности организации и правительства; выявление и прогноз реакции на информационные волны, связанные с нестандартными ситуациями или значимыми общественными событиями; оперативное получение информации о чрезвычайных происшествиях с места событий; мониторинг и анализ реакции общества в режиме реального времени на те или иные инициативы, события, публичные выступления или высказывания, пресс-конференции

Это позволит сделать выводы о наиболее эффективных каналах коммуникации, времени подачи «горячих», требующих обсуждения новостей.

Исследования. Одним из показателей «жизненной активности» социальной сети является число пользователей, вступающих в обсуждение. В социальных сетях их называют авторами. Авторы, создавая контент, делают социальные сети индикатором общественных настроений, позволяя уловить общественные проблемы в самом начале их формирования. Последние годы по количеству авторов в русскоязычном сегменте лидирует ВКонтакте (18.8 млн. авторов на 2016г.), у Instagram – 10.6млн., третье место занимает Twitter, причем в отличие от предыдущих сетей, число активных авторов снижается и в декабре 2016г. зафиксировано немногим более 1 млн. человек [3].

Одним из популярных инструментов активности социальных сетей является BRAND ANALYTICS, который собирает и индексирует публичные сообщения в сети. При анализе используется несколько показателей:

автор – пользователь, участник, написавший не менее одного сообщения за месяц;

сообщение – открытый пост, комментарии;

аудитория – количество участников, заходивших в группу не реже одного раза в месяц;

ER(Engagement Rate, Индекс Вовлеченности) - показатель реакции аудитории на публикации. Определяется как сумма комментариев, лайков, репостов ко всем публикациям за месяц.

По результатам исследований BRAND ANALYTICS ежемесячная аудитория групп ВКОНТАКТЕ составляет более 46 млн. человек, активных авторов около 19млн., из них 43.7% составляют мужчины, 56.3% - женщины [4].

Большинство исследователей социальных сетей рассматривают историю сети или отдельной группы в виде кривой жизненного цикла, рис.1[5]. Жизнь группы или всей сети начинается с начальной зоны, за которой следует «всплеск молвы» (резкое увеличение участников), затем спад «лощина сомневающихся» (или долина сомнений), когда часть участников покидает группу, затем рост числа участников сети до зоны стабильности и неизбежное умирание проекта с заменой его новым, более актуальным.

Нами проанализированы нескольких групп в Контакте. Были выбраны группы общего направления (кино, досуг, здоровье...) и группы, созданные для освещения и обсуждения событий в ДНР/ЛНР. Как показали исследования, группы политической направленности имеют свою динамику и достаточно резко отличаются от групп неполитического направления.

В среднем в обычных группах посещаемость за сутки колеблется от 1 до 2% в день. Например, группа Новинки музыки (<https://vk.com/public27895931>) имеет более 12 млн. подписчиков, однако в день страницу группы посещает не более 2% участников. В тоже время у группы Шторм Мариуполь (<https://vk.com/public71830794>), имеющей 25тыс. подписчиков, уровень посещаемости превышает 11%. У группы Киномания (9 млн. подписчиков) – процент посещаемости равен 2.6%; у группы АнтиМайдан (500тыс. участников) – 8.8%.

Две политических группы (Антимайдан и Сводки от ополчения Новороссии) были созданы в 2014г., российская новостная группа РенТВ в 2011г. Для первых двух групп характерен резкий всплеск числа подписчиков и зона стабильности (количество членов группы варьируется незначительно), рис.2., левая часть. Группа РенТВ не имеет зон всплеска, темпы роста медленнее.

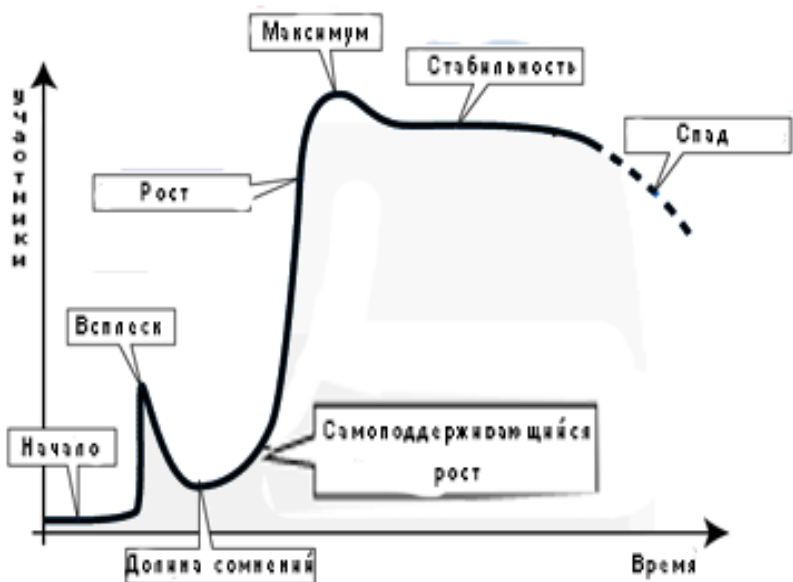


Рисунок 1 – Жизненный цикл социальных сетей

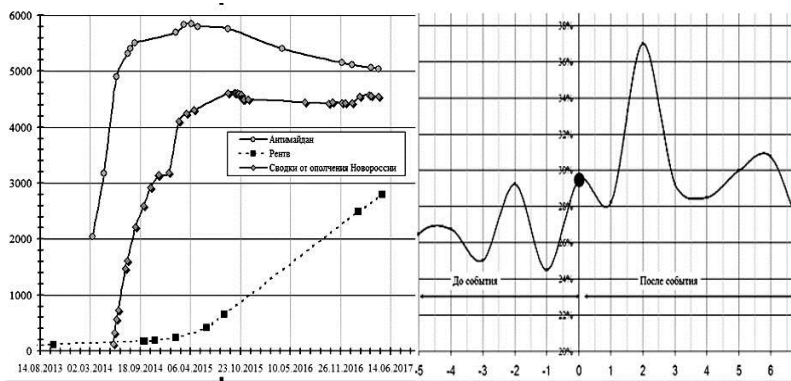


Рисунок 2- Жизненный цикл политических групп

В правой части рисунка показан всплеск активности участников в случае трагических событий. За начало отсчета (точка 0 соответствует дате смерти

одного из командиров ВСН). Анализ праздничных, памятных дат показал отсутствие аналогичных всплесков.

Рассматривая историю групп как временные ряды, мы выделили сезонную составляющую. Как видно из рис.3., наибольшей активностью участники группы отличаются в феврале и в осенние месяцы, наименьшей – в августе.

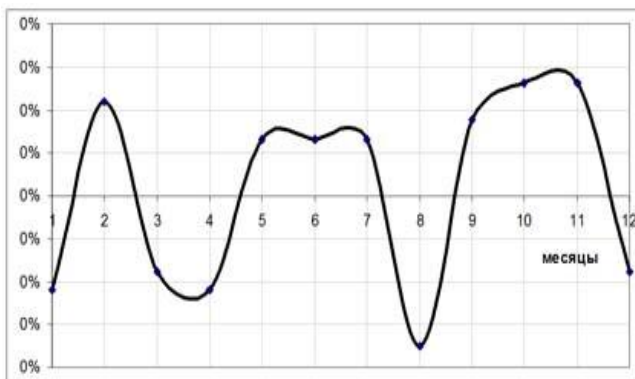


Рисунок 3 – Сезонная составляющая временных рядов

Анализ активности по дням недели показал, что наименьшее количество участников участвуют в обсуждениях в субботу.

Выводы. Проведен анализ активности социально-политических группа в Контакте, определен вид кривых жизненного цикла групп и его отличие от кривых, описывающих жизненный цикл бытовых групп. Определены пики и спады социальной активности по месяцам года.

Список литературы

1. Аудитория пользователей интернета в России в 2017 году составила 87 млн. человек [электронный ресурс] // URL: <http://mediascope.net/press/news/744498/>
2. Интернет в России и в мире. Количество пользователей интернета в России. // URL: http://www.bizhit.ru/index/users_count/0-151
3. ZONK сервис рейтинга. Рейтинг социальных сетей 2017. // URL: <http://zonkservice.ru/interest/rejting-sotsialnyih-setey-2017.html>
4. Социальные сети в России, осень 2016. Цифры, тренды, прогнозы// URL: <https://adindex.ru/publication/analytics/100380/2016/12/8/156545.phtml>
5. Wisdum P. The Ebb and Flow of Social Networking. URL: <http://www.wisdump.com/web/the-ebb-and-flow-of-social-networking/>

УДК 681.53

РАЗРАБОТКА АЛГОРИТМА ДЛЯ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ОХЛАЖДЕНИЕМ ШАХТНОГО ГАЗА-МЕТАНА

¹Кузнецов П.А., ²Масло С.В., ³Лисицин А.Н.

¹Рыбинский государственный авиационный технический университет им. П.А. Соловьева;

кафедра электротехники и промышленной электроники

²Донецкий национальный технический университет

кафедра прикладной математики

³ПАО «НПО «Сатурн»

E-mail: kuznetsovpavel@inbox.ru

Аннотация

Кузнецов П.А., Масло С.В., Лисицин А.Н. Разработка алгоритма для системы автоматического управления охлаждением шахтного газа-метана. В работе описывается предлагаемая система автоматического управления для охлаждения, сжигаемого в когенерационной станции ПАО «Шахта им. А.Ф. Засядько» газа-метана, на линии его предварительной подготовки.

Ключевые слова: метан, газ, шахта

Газ-метан, получаемый из недр земли, может обладать различными коррозионными характеристиками в зависимости от региона его добычи. Данный факт необходимо учитывать при проектировании и эксплуатации газопроводов КГЭС.

В работе рассмотрены особенности решения данного вопроса на КГЭС одного из крупнейших угольных предприятий Донбасса «Шахта им. А.Ф. Засядько».

Коррозия не только быстро выводит из строя рабочие элементы КГЭС, но может привести к появлению взрывов или пожаров [1, 2], которые являются опасными не только для рабочего персонала, но и для близлежащих объектов, связанных с жизнедеятельностью человека, т.к. КГЭС находится в черте г. Донецка. Для сокращения расходов на ремонт и обслуживание оборудования, создания более безопасных условий труда для персонала КГЭС была создана линия предварительной подготовки газа.

При эксплуатации одного из элементов линии предварительной подготовки газа, перед сотрудниками встала проблема по автоматизации

теплообменного процесса. Перед подачей газа в камеру сгорания он проходит несколько стадий осушки, охлаждения и смешивания. Для охлаждения газа использовались две холодильные установки, производства фирмы «GE Jeanbacher», однако их использование было целесообразным только в летнее время из-за большого потребления энергии компрессорами (около 400 кВт). Поэтому, для охлаждения газа в зимний, весенний и осенний периоды, используется система из двух теплообменников, которые охлаждаются вентиляторами, за счет температуры воздуха окружающей среды [3].

В данный момент система предварительной подготовки газа абсолютно не имеет узлов автоматического управления. Это не только подвергает жизнь рабочих дополнительному риску, но и не гарантирует нужной стабильности поддержания необходимой температуры охлаждаемого вещества. Это, в свою очередь, приводит к снижению коэффициента полезного действия КГЭС и уменьшению количества вырабатываемой электроэнергии. Схема теплообменника приведена на рис. 1.

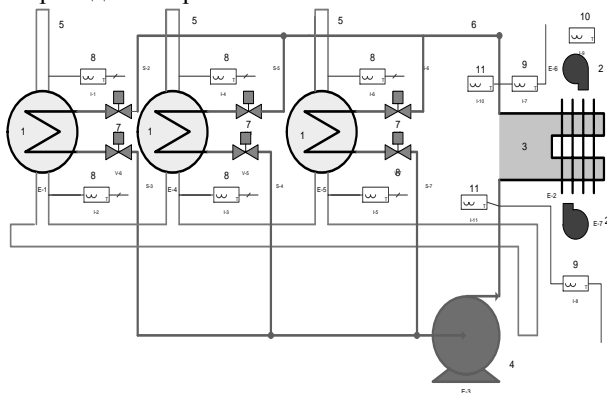


Рис. 1. Схема механизма теплообмена КГЭС

Система работает следующим образом. Вентиляторы (2), имеющие в качестве привода асинхронные двигатели мощностью 10 кВт каждый, обдувают теплообменник с водой (3). Далее охлажденная вода по трубопроводу (6) перегоняется с помощью центробежного насоса (4) в теплообменники (1). В последний поступает газ, требующий охлаждения по газопроводу (5). Давление воды регулируется с помощью вентиля (7). Замеры температуры газа на входе и выходе газопровода происходит с помощью датчиков температуры газа (8), температуры воды в трубопроводе замеряется датчиками (9). Температура окружающего воздуха замеряется датчиком (10). Весь процесс охлаждения управляется микроконтроллером в автономном режиме. Для написания кода автоматизации была разработана блок-схема автоматического управления процессом охлаждения.

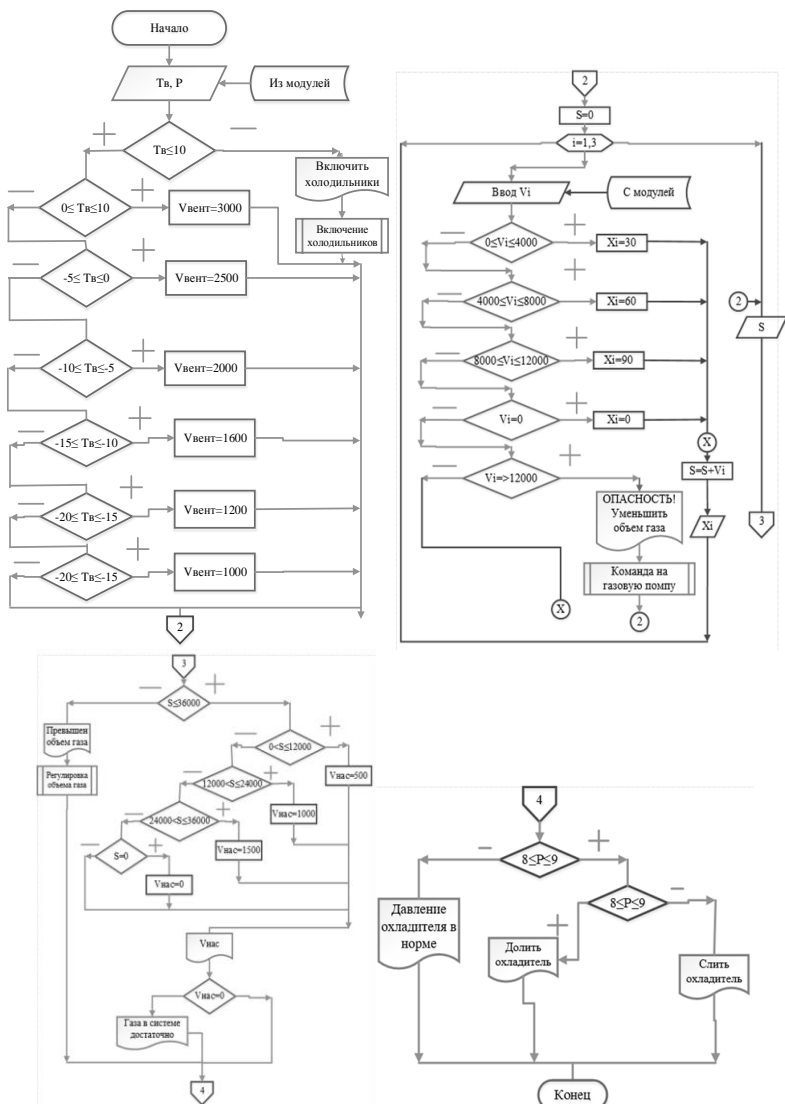


Рис. 2. Блок-схема алгоритма управления вентиляторами

Принцип работы программы автоматизации состоит в следующем: вначале происходит инициализация контроллера и сбор информации от всех датчиков. Исходными параметрами являются температуры газа в ветвях,

температуры охладителя, объем прокачиваемого воздуха, температура воздуха, скорости вращения приводов вентиляторов, давление охладителя в системе.

После инициализации происходит регулирование скорости вращения вентиляторов в зависимости от температуры воздуха. Опытным путем был установлен минимум, необходимый для обеспечения нужных скоростей вращения. После этого системе необходимо отрегулировать положение дроссельных заслонок, в зависимости от объема газа, находящегося в каждой ветви. Расходомером измеряется текущая величина и, в зависимости от объема, положение заслонки выставляется в необходимую позицию. В последних двух частях схемы происходит проверка давления газа и количества охладителя в линии.

Визуальная оболочка была разработана на языке ObjectPascal в среде Delphi (см. рис. 3).

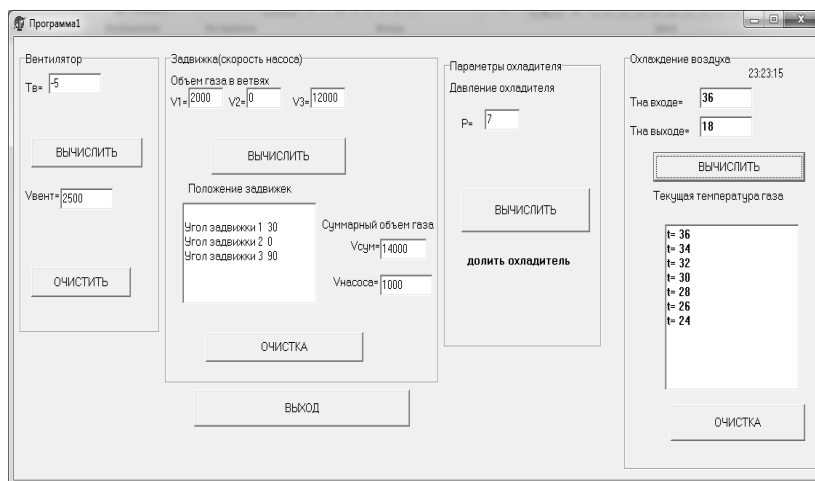


Рис. 3. Главная форма программы автоматизации

В ней критические области алгоритма визуально разделены на 4 части, каждая из которых выводит показания по соответствующим параметрам. В дальнейшем планируется реализовать в программе возможность предсказания аварийных ситуаций путем аппроксимации рабочих величин.

Использование вентиляторов вместо холодильных установок позволит сократить расход электроэнергии в 20 раз, а введение линии предварительной подготовки газа уменьшает износ дорогостоящего оборудования КГЭС на 60 % [4].

Таким образом, была показана необходимость создания системы автоматического управления узлом охлаждения шахтного газа-метана. Система автоматизации, разработанная для узла охлаждения газа, позволяет снизить влияние человеческого фактора в процессы, протекающие на КГЭС. Ее внедрение позволит сэкономить расход электрической энергии в узле охлаждения в холодное время года в 20 раз, и снизить риск травматизма рабочего персонала. Кроме того, использование модернизированного комплекса предварительной подготовки газа уменьшит коррозионные процессы в трубопроводах и увеличит износ рабочих узлов КГЭС.

Программа автоматизации и предлагаемые улучшения позволят значительно сократить вмешательство человека в рабочий процесс и облегчить использование альтернативного теплообменника в холодное время года.

Список литературы

1. ГОСТ 12.1.004-91. Пожарная безопасность. Общие требования. – М.: Изд-во стандартов, 1992 г. – 77 с.
2. ГОСТ 12.1.044-89. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения. – М.: Изд-во стандартов, 1989 г. – 99 с.
3. Кузнецов П.А. Использование когенерационных станций на загазованных шахтах Донбасса / П.А. Кузнецов, С.В. Борщевский // Социально-экономические и экологические проблемы горной промышленности, строительства и энергетики. Сборник научных трудов 9-ой международной конференции по проблемам горной промышленности, строительства и энергетики: 29-31 октября 2013 г., Минск-Тула-Донецк, 2013. – С. 16-21.
4. DIA.NE.XT. Руководство пользователя v.11.02. GE Jeanbacher. 2004, Austria. – 73 с.
5. Simatic. Программируемый контроллер S7-300. Данные модулей, Справочное руководство. Siemens Aktiengesellschaft. 02/2004. – 558 с.

МОДЕЛИРОВАНИЕ РАБОТЫ ГИДРОПРИВОДА ГОРНЫХ МАШИН

В.Г. Черепов¹, В.И. Зензеров²

Донецкий национальный технический университет, г. Донецк
студент гр. ПМК-15¹, доцент кафедры прикладной математики²
E-mail: mpvik@3g.ua

Аннотация

Черепов В.Г., Зензеров В.И. Моделирование работы гидропривода горных машин. *Рассматривается комплексная математическая модель, описывающая динамические процессы в гидросистеме механизированных крепей. При построении модели использован принцип декомпозиции объекта моделирования.*

Ключевые слова: *механизированная крепь, процесс, гидросистема, математическая модель, параметры.*

Целью работы является разработка математической модели, алгоритма и программы моделирования на ПЭВМ, позволяющие решать задачу исследования и расчета параметров гидропривода горных машин.

Научно-техническое обоснование и развитие методов и средств исследований, проектирования и расчета параметров гидропривода горных машин с характеристиками, обеспечивающими их эффективную эксплуатацию в условиях повышения нагрузок на очистной забой, является важной научной и практической задачей, имеющей отраслевое значение.

Решение задачи исследования и расчета параметров гидропривода горных машин посвящены работы ряда авторов [1-3]. Разработаны аналитические методы, проведены экспериментальные исследования. Вместе с тем, на современном этапе полученные результаты не обеспечивают достаточно полное решение задачи математического моделирования гидропривода, для выбора его рациональных параметров и обоснованного внесения изменений в конструкцию гидропривода.

В работе предлагается усовершенствованная математическая модель гидропривода, позволяющие решать задачу исследования и расчета параметров гидропривода горных машин. Решается задача разработки математической модели гидропривода горных машин в виде системы дифференциальных уравнений, описывающих процесс движения рабочей жидкости и перемещения гидроцилиндров, и исследования параметров динамических процессов в гидроприводе при выполнении операций технологического цикла. Для численного решения полученной системы на ПЭВМ предлагается использовать метод Рунге-Кутты, с внесением изменений для адаптации разработанного алгоритма к особенностям работы моделируемого гидропривода.

В качестве примера, выполним исследование работы гидросистемы механизированной крепи, как одной из базовых машин очистного комплекса.

При составлении схемы использованы исследования и графические обозначения, приведенные в работах [1- 3].

Работа гидросистемы описывается двумя типами уравнений:

- дифференциальные уравнения движения элементов системы, составленные по принципу Даламбера;
- уравнения баланса расходов, учитывающие, что жидкость в гидросистеме неразрывна и однородна.

На основании правила узлов и ветвей [3] составим математическую модель гидросистемы механизированной крепи.

Гидросистема крепи, схема которой приведена на рис. 1, состоит из N отдельных ветвей, моделирующих работу гидроцилиндров, и общих для всех гидроцилиндров участков напорной и сливной магистралей.

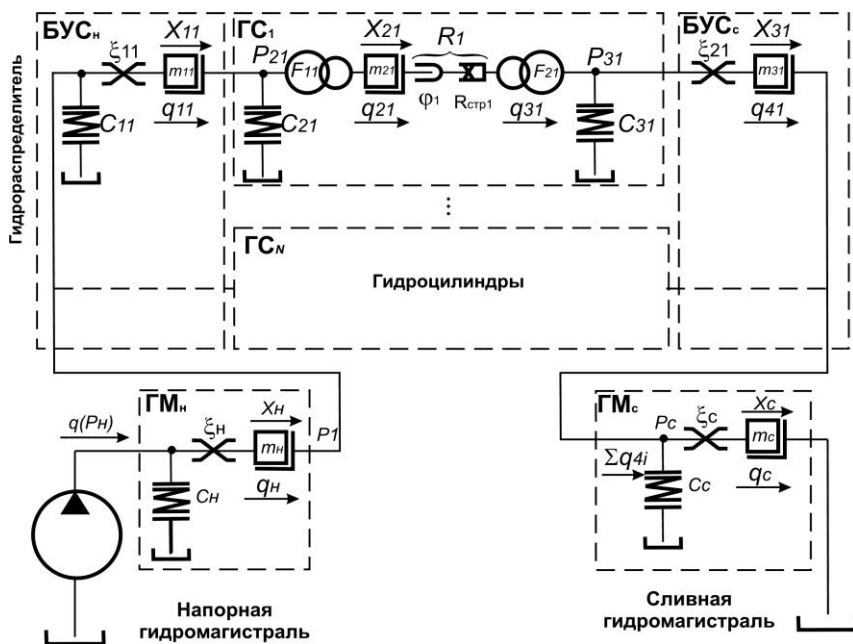


Рисунок 1 – Гидромеханическая схема гидросистемы механизированной крепи

Движение рабочей жидкости от насоса к гидрораспределителю секции крепи описывается системой трех дифференциальных уравнений:

$$\left. \begin{aligned} C_H \frac{dP_H}{dt} &= q(P_H) - q_H \\ \frac{m_H}{(f_H)^2} \cdot \frac{dq_H}{dt} &= P_H - \xi_H (q_H)^2 - P_1 \\ C_\Sigma \frac{dP_1}{dt} &= q_H - \sum_{i=1}^N q_{1i} \end{aligned} \right\} \quad (1)$$

Математическая модель работы каждого гидроцилиндра состоит из системы шести дифференциальных уравнений:

$$\left. \begin{aligned} \frac{m_{1i}}{(f_{\text{ПВД}i}^H)^2} \cdot \frac{dq_{1i}}{dt} &= P_1 - \xi_{1i} (q_{1i})^2 - P_{2i} \\ C_{2i} \frac{dP_{2i}}{dt} &= q_{1i} - q_{2i} \\ \frac{m_{2i}}{(F_{1i})^2} \cdot \frac{dq_{2i}}{dt} &= P_{2i} - \beta_i R_i (X_{2i}, \dot{X}_{2i}) - \alpha_i P_{3i} \\ \frac{dX_{2i}}{dt} &= \beta_i q_{2i} \\ C_{3i} \frac{dP_{3i}}{dt} &= \alpha_i q_{2i} - q_{4i} \\ \frac{m_{3i}}{(f_{\text{ПВД}i}^C)^2} \cdot \frac{dq_{4i}}{dt} &= P_{3i} - \xi_{2i} (q_{4i})^2 - P_C \end{aligned} \right\} , \quad (2)$$

где $R_i(X_{2i}, \dot{X}_{2i})$ - сопротивление перемещению i -ого гидроцилиндра.

Течение жидкости по сливной гидромагистральной от секции крепи в бак гидронасоса описывается системой двух дифференциальных уравнений:

$$\left. \begin{aligned} C_C \frac{dP_C}{dt} &= \sum_{i=1}^N q_{4i} - q_C \\ \frac{m_C}{(f_C)^2} \cdot \frac{dq_C}{dt} &= P_C - \xi_C (q_C)^2 \end{aligned} \right\} . \quad (3)$$

Системы дифференциальных уравнений (1), (2) и (3) образуют математическую модель гидросистемы механизированной крепи. Количество уравнений математической модели зависит от количества одновременно работающих гидроцилиндров N при моделировании различных технологических операций и определяется из выражения:

$$n = 6N + 5. \quad (4)$$

Исследование статических и динамических характеристик отдельных гидроэлементов и гидропривода в целом позволит выявить факторы, оказывающие решающее влияние на режим работы машины, и обосновать допущения, которые могут быть приняты в инженерных расчетах.

Специфичность гидропривода определяет особенности и сложности математического описания процессов, протекающих в его гидроэлементах. Для примера, при расчете гидросистемы четырехстоечной секции крепи типа во время выполнения операции передвижения с одновременной разгрузкой необходимо моделирование одновременной работы шести гидроцилиндров, что соответствует решению системы сорока одного дифференциального нелинейного уравнения. Число уравнений может изменяться в процессе моделирования в результате разного времени начала и окончания выполнения технологической операции различными гидроцилиндрами.

Перечисленные выше особенности расчетов показывают необходимость разработки алгоритмов и программ на ПЭВМ, позволяющих получить необходимый объем данных о динамике работы проектируемого гидропривода, что дает возможность автоматизировать процесс проектирования и сократить объем и трудоемкость экспериментальных и доводочных работ в целом.

В качестве контрольного примера был выполнен процесс моделирования операции перемещения секции крепи 1М88Н с одновременной разгрузкой гидростоек (параллельная работа трех гидроцилиндров), что соответствовало решению системы 23 дифференциальных уравнений.

Для визуального отображения процесса моделирования и ввода исходных данных разработана форма, представленная на рисунке 2.

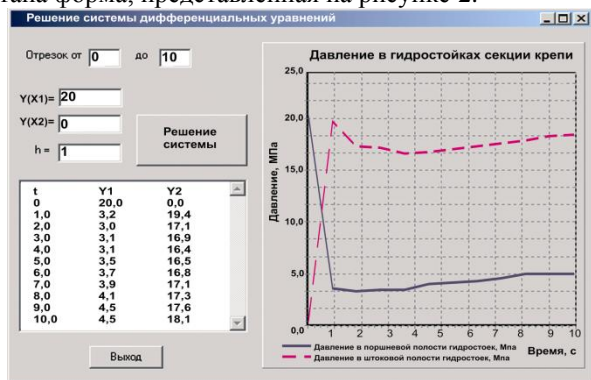


Рисунок 2 – Форма для решения системы дифференциальных уравнений

В результате проведенных исследований разработана математическая модель гидропривода горных машин, в виде системы дифференциальных

уравнений, описывающих процесс движения рабочей жидкости и перемещения гидроцилиндров, и исследования параметров динамических процессов в гидроприводе при выполнении операций технологического цикла [4,5].

Выводы. Предложена математическая модель процесса функционирования гидросистемы секции механизированной крепи при отработке наклонных пластов.

Для численного решения полученной системы на ПЭВМ предлагается использовать метод Рунге-Кутты, с внесением изменений для адаптации разработанного алгоритма к особенностям работы моделируемого гидропривода.

Разработанная математическая модель, алгоритм и программа приняты для включения в состав математического обеспечения системы автоматизированного проектирования гидропривода механизированных крепей (САПГМК) и используются в учебном процессе при изучении основ программирования.

Литература

1. Пономаренко Ю.Ф., Баландин А.А. Инженерная методика проектного расчета параметров гидросистемы механизированных крепей. – М.: ИГД им. А.А. Скочинского, 1981. – 22 с.
2. Расчет и конструирование гидроприводов механизированных крепей / Ю.Ф. Пономаренко, А.А. Баландин, И.Т. Богатырев и др. // Под общ. ред. Ю.Ф. Пономаренко. – М.: Машиностроение, 1961. – 327 с.
3. Тесленко А.И. Основы гидравлических расчетов механизированных крепей. – М.: Недра, 1974. – 216 с.
4. Математическое моделирование и расчет параметров систем угольных шахт и забойного оборудования: Монография / С.С. Гребенкин, В.Н. Павлыш, А.В. Агафонов, В.В. Косарев, В.В. Радченко, В.Д. Рябичев, В.П. Глебов, В.И. Зензеров, А.И. Егурнов. – Донецк: ВИК, 2007. – 263 с.
5. Основы создания и эффективной эксплуатации систем жизнеобеспечения очистного оборудования для угольных шахт: [моногр.] / [С.С. Гребёнкин, В.В. Косарев, С.Е. Топчий, Н.И. Стадник, В.И. Зензеров, В.В. Стеблин, Б.А. Перепелица, В.Н. Поповский]; под общей редакцией Гребенкина С.С. и Косарева В.В. – Донецк: «ВИК», 2009. – 375с.

УДК 531.38; 531.39

ПРИМЕНЕНИЕ ТЕОРИИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ПОСТРОЕНИИ ПОЛИНОМИАЛЬНЫХ РЕШЕНИЙ УРАВНЕНИЙ ЭЙЛЕРА-ПУАССОНА

Ткаченко Д.Н.

Государственное учреждение «Институт прикладной математики и механики» Министерства образования и науки Донецкой народной республики

E-mail: dntkachenko@mail.ru

Аннотация

Ткаченко Д.Н. Применение теории дифференциальных уравнений в построении полиномиальных решений уравнений Эйлера-Пуассона. Исследованы условия существования полиномиальных решений класса Стеклова-Ковалевского-Горячева-Чаплыгина уравнений движения динамически симметричного тяжелого твердого тела. На основе уравнений Н.Ковалевского показано, что такими решениями являются только частные случаи интегрируемости уравнений Эйлера-Пуассона, полученные С.В.Ковалевской и Д.Н.Горячевым.

Ключевые слова: дифференциальные уравнения

Введение

В статье исследованы условия существования полиномиальных решений класса Стеклова-Ковалевского-Горячева-Чаплыгина уравнений движения динамически симметричного тяжелого твердого тела. На основе уравнений Н.Ковалевского показано, что такими решениями являются только частные случаи интегрируемости уравнений Эйлера-Пуассона, полученные С.В.Ковалевской и Д.Н.Горячевым.

Уравнения Эйлера-Пуассона, моделирующие движение тяжелого твердого тела с неподвижной точкой, легли в основу формирования новых задач аналитической механики. Обзор результатов, полученных в исследовании классической задачи, отражен в многочисленных монографиях [1-3]. Поскольку уравнения Эйлера-Пуассона в общем случае неинтегрируемы в квадратурах, то для изучения свойств движения тела применялись различные методы интегрирования этих уравнений. Среди этих методов особую роль в построении новых решений играют редуцированные уравнения динамики твердого тела [3]. Наибольшее количество частных решений уравнений Эйлера-Пуассона получено в случае, когда центр масс тела лежит на главной

оси эллипсоида инерции. В статье рассмотрен класс частных решений данных уравнений, который характеризуется полиномиальной структурой инвариантных соотношений на компоненты вектора угловой скорости. Известно [4], что редуцированные уравнения для случая, когда центр масс тела лежит на главной оси, можно привести к виду, в котором основными переменными являются квадраты проекций вектора угловой скорости на небарицентрические оси, а вспомогательной переменной служит проекция вектора угловой скорости на барицентрическую ось. То есть, уравнения Н.Ковалевского представляют собой систему двух дифференциальных уравнений второго порядка. Если основные переменные являются многочленами по вспомогательной переменной, то указанные уравнения допускают решения В.А.Стеклова [5], Н. Ковалевского[4], Д.Н.Горячева [6]. Обобщение этих решений на задачу о движении тяжелого гиригастата рассмотрено П.В.Харламовым [7]. Г.В.Горпоказал, что для динамически несимметричных тел уравнения Н.Ковалевского допускают единственное решение С.А.Чаплыгина, которое характеризуется полиномиальной структурой основных переменных от кубического корня вспомогательной переменной. Анализ частных решений уравнений Эйлера-Пуассона показывает, что для динамически симметричного тела частный случай решения С.В.Ковалевской относится к полиномиальному решению класса Стеклова-Ковалевского-Горячева, а частный случай решения Д.Н.Горячева — к полиномиальному решению класса С.А.Чаплыгина.

Целью данной статьи является исследование полиномиальных решений классов В.А.Стеклова, Н.Ковалевского, Д.Н.Горячева и С.А.Чаплыгина в случае динамического симметричного твердого тела. На основе уравнений Н. Ковалевского показано, что для первого класса имеет место только частный случай решения С.В.Ковалевской, а для второго случае — частный вариант решения Д.Н.Горячева, который характеризуется условием $A_1 = A_2 = 4A_3$.

Постановка задачи. Уравнения Н.Ковалевского

Рассмотрим уравнения движения тяжелого гиригастата, центр масс которого лежит на главной оси эллипсоида инерции

$$A_1 \dot{p} = (A_2 - A_3)qr \quad (1)$$

$$A_2 \dot{q} = (A_3 - A_1)pr - sv_3, \quad A_3 \dot{r} = (A_1 - A_2)qp + sv_2 \quad (2)$$

$$\dot{v}_1 = rv_2 - qv_3, \quad \dot{v}_2 = pv_3 - rv_1, \quad \dot{v}_3 = qv_1 - pv_2 \quad (3)$$

В уравнениях (1)-(3) введены следующие обозначения: p, q, r — компоненты вектора угловой скорости; v_1, v_2, v_3 — компоненты единичного вектора, указывающего направление силы тяжести; A_i — главные моменты инерции тела; s — произведение массы тела, ускорения свободного падения и расстояния от неподвижной точки до центра масс тела; точка над переменными означает производную по времени t . Уравнения (1)-(3) имеют первые интегралы

$$v_1^2 + v_2^2 + v_3^2 = 1, \quad A_1pv_1 + A_2qv_2 + A_3rv_3 = k, \quad (4)$$

$$A_1p^2 + A_2q^2 + A_3r^2 - 2sv_1 = 2E. \quad (5)$$

где E – произвольные постоянные

Следующая статья [19], система уравнений и интегралов (1)-(5) может быть редуцирована, с учетом замены $\sigma = p^{1/3}$ к следующим уравнениям

$$\begin{aligned} & [2R(\sigma) + (A_1 - A_3)\sigma^6] \left(\sigma \frac{d^2Q(\sigma)}{d\sigma^2} - 2 \frac{dQ(\sigma)}{d\sigma} \right) + \\ & + \sigma \frac{dQ(\sigma)}{d\sigma} \left[\frac{dR(\sigma)}{d\sigma} + 3(A_1 - A_3)\sigma^5 \right] + 3A_2\sigma^6 \frac{dR(\sigma)}{d\sigma} - \\ & - \frac{9A_1A_2}{A_2 - A_3} \sigma^5 (Q(\sigma) - R(\sigma)) + 9A_2E\sigma^5 = 0, \end{aligned} \quad (6)$$

$$\begin{aligned} & [2Q(\sigma) + (A_1 - A_2)\sigma^6] \left(\sigma \frac{d^2R(\sigma)}{d\sigma^2} - 2 \frac{dR(\sigma)}{d\sigma} \right) + \\ & + \sigma \frac{dR(\sigma)}{d\sigma} \left[\frac{dQ(\sigma)}{d\sigma} + 3(A_1 - A_2)\sigma^5 \right] + 3A_2\sigma^6 \frac{dR(\sigma)}{d\sigma} - \\ & - \frac{9A_1A_3}{A_2 - A_3} \sigma^5 (Q(\sigma) - R(\sigma)) + 9A_3E\sigma^5 = 0, \end{aligned} \quad (7)$$

$$\begin{aligned} & A_2(A_2 - A_3)[2Q(\sigma) + (A_1 - A_2)\sigma^6] \left(\frac{dR(\sigma)}{d\sigma} \right)^2 + \\ & + A_3(A_3 - A_2)[2R(\sigma) + (A_1 - A_3)\sigma^6] \left(\frac{dQ(\sigma)}{d\sigma} \right)^2 + \\ & + 9A_1A_2A_3\sigma^4 \left(Q(\sigma) - R(\sigma) + \frac{(A_3 - A_2)}{A_1} E \right)^2 = \frac{9A_2A_3(A_2 - A_3)^2 s^2}{A_1} \sigma^4. \end{aligned} \quad (8)$$

Исследуем условия существования уравнений (6)-(8) решений полиномиального вида

$$Q(\sigma) = a_0 + a_1\sigma + \dots + a_6\sigma^6, \quad R(\sigma) = b_0 + b_1\sigma + \dots + b_{12}\sigma^{12} \quad (9)$$

Поскольку в [20] анализ выполнен при $A_1 \neq A_2$, то в данной статье полагаем в (6)-(8), что $A_1 = A_2$.

Тестовые примеры решений

С учетом предположения $A_1 = A_2$ проверим систему (6), (7) на совместность в случае С.В.Ковалевской, где при условии $A_1 = A_2 = 2A_3$ решение принимает вид:

$$q^2(p) = -\alpha p + \beta p^2, \quad r^2(p) = 4R^2 - 4p^2. \quad (10)$$

При переходе к вспомогательной переменной σ решение примет вид:

$$Q(\sigma) = A_3(R^2 - \sigma^6), \quad R(\sigma) = -\frac{A_3}{2}(-\alpha\sigma^3 + (\beta + 1)\sigma^6) \quad (11)$$

Подставив (11) в (6)-(7) увидим, что система разрешима при $\beta = -1$.

Исследуем (6), (7) на разрешимость в случае Д.Н.Горячева, решение которого имеет вид:

$$q^2(p) = p^{2/3}(b - p^{4/3}), \quad r^2(p) = p^{2/3}(kp^{2/3} - 4b^2), \quad (12)$$

При переходе к из (12) имеем

$$Q(p) = \frac{3A_3}{8}(\sigma^2(k\sigma^2 - 4b^2)), \quad R(p) = -\frac{3A_3}{2}b\sigma^2. \quad (13)$$

В этом случае система так же разрешима.

Условия существования общего класса полиномиальных решений

Рассмотрим случай, когда функции $Q(\sigma)$ и $R(\sigma)$ имеют вид (9). Подставим (9) в уравнения (6)-(8) и потребуем, чтобы полученное равенство было тождеством по σ . Применяя компьютерное моделирование, получим систему алгебраических уравнений на параметры задачи и параметры решения (9).

Анализ данной системы показал, что она разрешима только в случаях С.В. Ковалевской и Д.Н.Горячева.

Литература

1. Гашененко И.Н., Горр Г.В., Ковалев А.М. Классические задачи динамики твердого тела. – Киев: Наук. думка, – 2012. – 401 с.
2. Архангельский Ю.А. Аналитическая механика твердого тела. – М.: Наука. – 1977. – 328 с.
3. Харламов П.В. Лекции по динамике твердого тела. – Новосибирск: Изд-во Новосиб. университета. – 1965. – 221 с.
4. Kowalewski N. Eine neue partikuläre Lösung der Differentialgleichungen der Bewegung eines schweren starren Körpers um einen festen Punkt // Math. Ann – 1908. – **65**. – S/ 528-537/
5. Стеклов В.А. Новое частное решение дифференциальных уравнений движения тяжелого твердого тела, имеющего неподвижную точку // Тр. отд-ния физ. наук о-ва любителей естествознания. – 1899. – **10**, вып. 1. – С. 1–3.
6. Горячев Д.Н. Новое частное решение задачи о движении тяжелого твердого тела вокруг неподвижной точки // Тр. отд-ния физ. наук о-ва любителей естествознания. – 1899. – **10**, вып. 1. – С. 23–24.
7. Харламов П.В. Полиномиальные решения уравнений движения тела, имеющего неподвижную точку // Прикл. математика и механика. – 1965. – **29**, вып. 1. – С. 26-34.

УДК 622.14

ПРИМЕНЕНИЕ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ДЛЯ НАХОЖДЕНИЯ МИНИМАЛЬНОГО ВРЕМЕНИ ВЗАИМОДЕЙСТВИЯ ЗВЕНЬЕВ ТРАНСПОРТНОЙ ЛОГИСТИЧЕСКОЙ ЦЕПИ

А.Р.Клищевский¹, Л.А.Лазебная²

Донецкий национальный технический университет, г. Донецк
студент гр. КИТ-16¹, ст.преподаватель кафедры прикладной математики²

lazebnav@mail.ru

Аннотация

Клищевский А.Р., Лазебная Л.А. Применение линейного программирования для нахождения минимального времени взаимодействия звеньев транспортной логистической цепи. В статье предлагается концепция распределения суммарных ресурсов транспортной логистической цепи, которая позволит минимизировать общее время на погрузочно-разгрузочные операции в данной цепи.

Ключевые слова: грузовой фронт, логистическая цепь, целевая функция, матрица.

Общая постановка проблемы. Транспортно-логистическая цепь (ТЛЦ) представляет собой совокупность перевозок грузов и перегрузочных складов, расположенных в пунктах взаимодействия разных видов транспорта [1] (рис. 1).

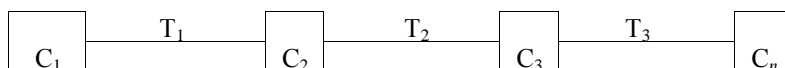


Рис. 1. Пример транспортно-логистической цепи

В данной работе в качестве примера рассмотрены железнодорожный и автомобильный транспорт, участвующие в процессе распределения материальных потоков. Последовательная логистическая транспортная цепь представлена в виде взаимодействующих звеньев.

Главным критерием оценки работы ТЛЦ является время доставки грузов. Задача состоит в том, чтобы таким образом распределить между звеньями ТЛЦ общие ресурсы, выделенные на оснащение данного объекта, чтобы минимизировать суммарное время доставки грузов [2,3].

В качестве расчетного примера рассмотрим достаточно простой случай, когда необходимо распределить средства между двумя звеньями ТЛЦ. В качестве первого выступает грузовой фронт на железнодорожной станции ГФ1, в качестве второго – грузовой фронт у грузополучателя ГФ2 (рис. 2)

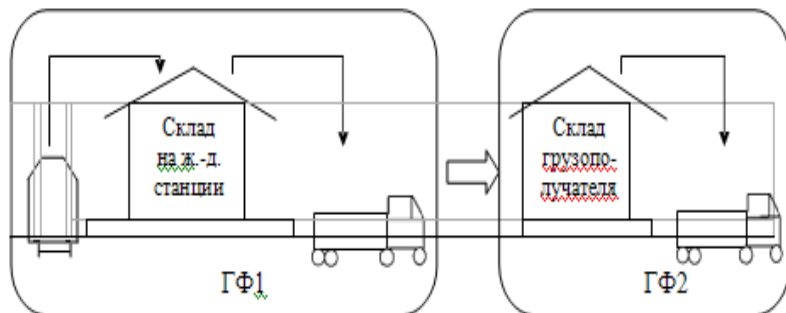


Рис.2. Звенья транспортно-логистической цепи:

ГФ1 – железнодорожная станция; ГФ2 – склад грузополучателя

С точки зрения линейного программирования, данная задача формулируется следующим образом.

Существует функция

$$f(T_1, T_2) \quad (1)$$

где

$$T_1 = (Q_c, Z_1, Q_{t1}, T_3) \quad (2)$$

$$T_2 = (Q_c, Z_2, Q_{t2}) \quad (3)$$

при ограничениях $f(T_1, T_2) \rightarrow \min$ (3)

где $f(T_1, T_2)$ - целевая функция, или критерий эффективности.

В данном случае, вектор f характеризует значение минимального времени в зависимости от вида погрузочно-разгрузочных операций.

Вектора T_1 и T_2 - варьируемые параметры, которые обозначают соответствующие компоненты для нахождения времени на соответствующих узловых фронтах.

Критерием оптимизации в данной задаче является общее время T на выполнение погрузочно-разгрузочных работ по всем грузовым фронтам.

Таким образом, используя данные выше приведенных функций, можно составить матрицу размерностью $[1..k, 1..m]$, где k - количество компонент, m - доля инвестиций для каждого звена (грузового фронта).

По результатам вычисления данной матрицы можно сделать следующий **вывод**: зная минимальное время на погрузочно-разгрузочные операции в данной цепи (это грузовой фронт на железнодорожной станции и грузовой фронт у грузополучателя), можно распределить суммарные ресурсы, выделенные на

развитие транспортно-логистической цепи таким образом, чтобы данная стоимость погрузочно-разгрузочных машин по всем звеньям цепи не превышала объема выделенных инвестиций.

Список литературы

1. Управление транспортными системами. Транспортное обеспечение логистики : учебник и практикум для академического бакалавриата / [В.Д.Герامي](#), [А. В. Колик](#). - М.: Издательство Юрайт, 2014. — 510 с. — Серия : Бакалавр. Академический курс. - ISBN 978-5-9916-4081-7
2. [Федоренко А. И.](#), [Домнина С. В.](#), [Токарева Е. В.](#), [Чурилова М. И.](#) [Бизнес-планирование в логистике](#) / Под общ. ред.: [В. И. Сергеев](#). - М.: Эс-Си-Эм Консалтинг, 2013.
3. Транспортная логистика: учебное пособие / Л. Э. Еремеева; Сыкт. лесн. ин-т. — Сыктывкар: СЛИ, 2013. — 260 с.

УДК 531.38

О ПЕРВЫХ ИНТЕГРАЛАХ УРАВНЕНИЙ ПУАССОНА ПРИ НАЛИЧИИ ТРЕХ ИНВАРИАНТНЫХ СООТНОШЕНИЙ В ЗАДАЧЕ О ДВИЖЕНИИ ГИРОСТАТА

А.В.Мазнев, Т.В.Белоконь

ГОУ ПО Донецкий национальный университет, г.Донецк
ГОУ ПО Донецкий национальный университет экономики
и торговли имени Михаила Туган-Барановского, г.Донецк

E-mail: aleksandr_maznev@rambler.ru

В.Tatyana13@mail.ru

Аннотация.

Мазнев А.В., Белоконь Т.В. О первых интегралах уравнений Пуассона при наличии трех инвариантных соотношений в задаче о движении гиростата. В работе рассмотрены уравнения движения твердого тела с неподвижной точкой под действием потенциальных и гироскопических сил. Исследованы вопросы интегрирования этих уравнений, когда они допускают три инвариантных соотношения.. Построено интегральное соотношение, позволяющее получать первые интегралы некоторых классов уравнений Пуассона.

Ключевые слова: уравнение Пуассона, уравнения Кирхгофа-Пуассона
Общая постановка проблемы

Как правило, многие задачи динамики твердого тела с неподвижной точкой описываются дифференциальными уравнениями, которые имеют шестой порядок и допускают три первых интеграла. Согласно теории К.Якоби [1] для их интегрирования в квадратурах достаточно найти дополнительный первый интеграл. В классической задаче о движении тяжелого твердого вокруг неподвижной точки такие интегралы получены Л.Эйлером, Ж. Лагранжем и С.В. Ковалевской [2].

Обобщением уравнений Эйлера-Пуассона являются уравнения Кирхгофа-Пуассона, которые описывают задачу о движении тела под действием потенциальных и гироскопических сил и задачу о движении тела в жидкости. Однако, как показали В.В.Козлов и Д.А. Онищенко[3], эти уравнения не интегрируются в квадратурах. В связи с этим актуальным направлением исследования свойств движения тела вокруг неподвижной точки является направление, которое основано на построении частных решений уравнений динамики. Универсальным методом построения новых частных решений уравнений динамики служит общий метод инвариантных соотношений (ИС), предложенный П.В.Харламовым для автономных дифференциальных уравнений [4]. В монографии [5] дан обзор результатов построения первых интегралов при наличии у уравнений динамики твердого тела определенного числа ИС. В последние годы в динамике твердого тела интенсивно изучается задача о движении гиростата с переменным гиростатическим моментом. Моделирование движений гиростата в этом случае приводит к системе неавтономных дифференциальных уравнений. А.М.Ковалев, Г.В.Горр и В.Н.Неспирный[6] получили необходимые и достаточные условия существования ИС неавтономных дифференциальных уравнений и обобщили метод ИС П.В.Харламова. Уравнения Д. Гриоли являются наиболее общими уравнениями движения твердого тела под действием потенциальных и гироскопических сил, которые допускают три первых интеграла. При решении обратной задачи о нахождении условий существования у этих уравнений трёх ИС необходимо проинтегрировать уравнения Пуассона произвольной структуры. Различные случаи нахождения решений уравнений Пуассона приведены в статье Г.В. Горра, А.В. Мазнева [7].

Исследования

Рассмотрим уравнение Д. Гриоли[8], которое описывает движение твёрдого тела с неподвижной точкой под действием потенциальных и гироскопических сил:

$$A\ddot{\bar{\omega}} = A\bar{\omega} \times \bar{\omega} + \left[\mu(v_1, v_2, v_3)\bar{v} + \frac{\partial L(v_1, v_2, v_3)}{\partial \bar{v}} \right] \times \bar{\omega} + \frac{\partial U(v_1, v_2, v_3)}{\partial \bar{v}} \times \bar{v}, \dot{\bar{v}} = \bar{v} \times \bar{\omega}. (1)$$

где $\bar{\omega} = (\omega_1, \omega_2, \omega_3)$ - вектор угловой скорости тела; $\bar{v} = (v_1, v_2, v_3)$ - единичный вектор оси симметрии силовых полей; $A = (A_{ij})$ - тензор инерции;

$\mu(v_1, v_2, v_3), L(v_1, v_2, v_3), U(v_1, v_2, v_3)$ - дифференцируемые функции переменных v_1, v_2, v_3 ; $\frac{\partial L}{\partial \bar{v}}$ и $\frac{\partial U}{\partial \bar{v}}$ - градиенты функций $L(v_1, v_2, v_3)$ и $U(v_1, v_2, v_3)$.

Уравнения (1) допускают интегралы

$$\bar{v} \cdot \bar{v} = 1, A\bar{\omega} \cdot \bar{v} + L(v_1, v_2, v_3) = k, A\bar{\omega} \cdot \bar{\omega} - 2U(v_1, v_2, v_3) - 2E, \quad (2)$$

где k и E - произвольные постоянные.

Поставим задачу об исследовании у уравнений (1) трёх инвариантных соотношений

$$\omega_1 = \omega_1(v_1, v_2, v_3), \omega_2 = \omega_2(v_1, v_2, v_3), \omega_3 = \omega_3(v_1, v_2, v_3), \quad (3)$$

где функции заданы в пространстве $R_3(v_1, v_2, v_3)$. Будем предполагать, что ИС (3) являются ИС по Леви-Чивите [5], т.е. производные от соотношений (3) обращаются в нуль на всем множестве (3) при $\bar{v} \in R_3$. Для нахождения условий существования ИС (3) у уравнений (1), (2) необходимо вначале подставить $\bar{\omega}$ из (3) во второе уравнение системы (1):

$$\dot{\bar{v}} = \bar{v} \times \omega(\bar{v}), \quad (4)$$

где $\omega(\bar{v}) = (\omega_1(v_1, v_2, v_3), \omega_2(v_1, v_2, v_3), \omega_3(v_1, v_2, v_3))$, а затем изучить решения первого уравнения из (1) на ИС (3) с учетом (4).

Из интегралов (2) найдём:

$$L(v_1, v_2, v_3) = k - A\omega(\bar{v}) \cdot \bar{v}, \quad U(v_1, v_2, v_3) = \frac{1}{2}(A\omega(\bar{v}) \cdot \omega(\bar{v})) - E. \quad (5)$$

Из первого уравнения системы (1) с учётом (4), (5) получим

$$\mu(v_1, v_2, v_3) = A_1 \frac{\partial \omega_1(v_1, v_2, v_3)}{\partial v_1} + A_2 \frac{\partial \omega_2(v_1, v_2, v_3)}{\partial v_2} + A_3 \frac{\partial \omega_3(v_1, v_2, v_3)}{\partial v_3} \quad (6)$$

где $A_i (i = \overline{1,3})$ - главные моменты инерции тела. Вводя обозначения $\bar{g} = (A_1\omega_1, A_2\omega_2, A_3\omega_3)$, условие (6) запишем в виде

$$\mu(v_1, v_2, v_3) = \frac{\partial g_1(v_1, v_2, v_3)}{\partial v_1} + \frac{\partial g_2(v_1, v_2, v_3)}{\partial v_2} + \frac{\partial g_3(v_1, v_2, v_3)}{\partial v_3} = \text{div} \bar{g}, \quad (7)$$

где $g_i = A_i \omega_i(v_1, v_2, v_3)$. Таким образом, задача интегрирования уравнений (1) на ИС (3) сведена к интегрированию уравнения (4).

Положим в (1)

$$U(v_1, v_2, v_3) = \bar{s} \cdot \bar{v} - \frac{1}{2}(C\bar{v} \cdot \bar{v}), \quad L(v_1, v_2, v_3) = \bar{\lambda} \cdot \bar{v} - \frac{1}{2}(B\bar{v} \cdot \bar{v}) \quad (8)$$

где $\bar{s} = (s_1, s_2, s_3)$, $\bar{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ постоянные векторы; B и C - постоянные симметричные матрицы третьего порядка. Тогда из уравнения (1), в силу (8), получим уравнение класса Кирхгофа-Пуассона

$$A\dot{\bar{\omega}} = (A\bar{\omega} + \bar{\lambda}) \times \bar{\omega} + \bar{\omega} \times B\bar{v} + \bar{v} \times (C\bar{v} - \bar{s}), \quad \dot{\bar{v}} = \bar{v} \times \bar{\omega} \quad (9)$$

Рассмотрим ИС (3) в виде

$$\omega_1 = \beta_1 v_3, \quad \omega_2 = \beta_2 v_3, \quad \omega_3 = \frac{\varphi(v_3)}{v_3}. \quad (10)$$

Запишем для ИС (10) уравнения Пуассона (4):

$$\dot{v}_1 = \frac{1}{v_3}(v_2\varphi(v_3) - \beta_2 v_3^3), \quad \dot{v}_2 = \frac{1}{v_3}(-v_1\varphi(v_3) + \beta_1 v_3^3), \quad \dot{v}_3 = v_3(\beta_2 v_1 - \beta_1 v_2). \quad (11)$$

Уравнения (11) допускают интегральное соотношение

$$\beta_1 v_1 + \beta_2 v_2 + \int \frac{\varphi(v_3)}{v_3} dv_3 = c. \quad (12)$$

где c - произвольная постоянная. Формула (12) позволяет при заданной функции $\varphi(v_3)$, например, при

$$\varphi(v_3) = \alpha_0 + \alpha_1 v_3 + \alpha_3 v_3^2 + \dots + \alpha_n v_3^n \quad (13)$$

получить первый интеграл уравнений (11):

$$\beta_1 v_1 + \beta_2 v_2 - \frac{\alpha_0}{v_3} + \alpha_1 \ln |v_3| + \alpha_2 v_3 + \dots + \alpha_n v_3^{n-1} = c. \quad (14)$$

С помощью соотношений $\bar{v} \cdot \bar{v} = 1$ и (14) интегрирование уравнений (11) сводится к нахождению квадратур. Реализацией этого подхода являются результаты статьи [7].

Выводы. С помощью метода инвариантных соотношений найдено интегральное соотношение, позволяющее получать первые интегралы одного класса уравнений Пуассона. Данный подход позволяет задавать инвариантные соотношения определенного вида и находить первые интегралы для новых классов уравнений Пуассона.

Литература

1. Jacobi C.G.J./ Second memoire sur la rotation d'un corps non soumis a des forces acceleratrices //Gesammelte Werke. – Berlin: G. Reimer, 1882. – 2. – 427-467.
2. Гашененко Н.Н., Горр Г.В., Ковалев А.М. Классические задачи динамики твердого тела. // Серия "Задачи и методы: Математика, Механика, Кибернетика", ИПММ, Киев, Наукова Думка, – 2012.– 402 с.
3. Козлов В.В., Онищенко Д.А. Неинтегрируемость уравнений Кирхгофа // Докл. АН СССР. – 1982. – 266, № 6. – С. 1298-1300.
4. Харламов П.В. Об инвариантных соотношениях системы дифференциальных уравнений // Механика твердого тела. – 1974. – Вып. 6. – С. 15-24.
5. Горр Г.В. Инвариантные соотношения уравнений динамики твердого тела. Теория, результаты, комментарии. – М.-Ижевск: Институт компьютерных исследований; 2017. – 424 с.
6. Ковалев А.М, Горр Г.В., Неспирный В.Н. Инвариантные соотношения неавтономных дифференциальных уравнений с приложением в механике // Механика твердого тела. – 2013. – Вып. 43. – С. 3-18.
7. Горр Г.В., Мазнев А.В. Об интегрировании уравнений динамики твердого тела на инвариантных многообразиях // Механика твердого тела. – 2016. – Вып. 46. – С. 26-36.
8. Grioli G. Questioni di dinamica solido pesante asimmetrico //Attl. Accad. Naz. Lincei, Rend. Cl. sci. fis., mat. e natur. – 1963. – 35, № 1-2. – P. 35-30.

УДК 517.9

ЭКСТРЕМАЛЬНЫЕ РАСШИРЕНИЯ НЕОТРИЦАТЕЛЬНОГО ОПЕРАТОРА ЯКОБИ

Ананьева А.Ю.

ГУ Институт прикладной математики и механики

Отдел уравнений в частных производных

E-mail: ananeva89@gmail.com

Аннотация

Ананьева А.Ю. Экстремальные расширения неотрицательного оператора Якоби. Рассмотрена минимальный неотрицательный оператор Якоби с матричными элементами. Используя технику граничных троек и соответствующим им функций Вейля, описаны расширения Фридрихса и Крейна минимального оператора Якоби. Кроме того, параметризуемо множество всех неотрицательных расширений в терминах граничных условий.

Ключевые слова: расширения Фридрихса и Крейна, оператор Якоби

Пусть A - плотно определенный неотрицательный симметричный оператор в гильбертовом пространстве H . Так как A неотрицательно, то по теореме Фридрихса-Крейна оно допускает неотрицательные самосопряженные расширения. Квалифицированное описание всех неотрицательных самосопряженных расширений A , а также критерий единственности неотрицательного самосопряженного расширения A сначала были даны Крейном в [16]. Его результаты были обобщены в многочисленных работах (см. [3, 9, 11] и ссылки там).

Среди всех неотрицательных самосопряженных расширений A два (экстремальных) расширения являются особенно интересными и достаточно важными, чтобы иметь имя. Расширение Фридрихса (так называемое «жесткое» расширение) A_F является «наибольшим» в смысле квадратичных форм, а крейновское расширение является «наименьшим».

В настоящей работе рассматривается задача описания экстремальных расширений неотрицательного оператора Якоби, которая будет определена ниже. Пусть $A_j, B_j \in C^{p \times p}$. Более того, мы считаем, что матрицы A_j самосопряжены и матрицы B_j обратимы для каждого $j \geq 0$ (см. [5, глава VII, §2]). Рассмотрим полубесконечную матрицу Якоби с матричными элементами

$$J = \begin{pmatrix} A_0 & B_0 & O_p & \dots \\ B_0^* & A_1 & B_1 & \dots \\ O_p & B_1^* & A_2 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}.$$

Где O_p является нулевой $p \times p$ матрицей. Если положить $B^{-1} = O_p$, то $(Ju)_j = B_j u_{j+1} + A_j u_j + B_{j-1}^* u_{j-1}$, $j \geq 0$.

Максимальный оператор T_{\max} определяется как $(T_{\max} u)_j = (Ju)_j$, $j \geq 0$ на области $\text{dom}(T_{\max}) = \{u \in l_p^2 : Ju \in l_p^2\}$.

Обозначим через $(P_j(z))$ решение матричного уравнения $(Ju)_j = zu_j$, $j \geq 0$, с $P_0(z) = I_p$, $P_1(z) = B_0^{-1}(zI_p - A_0)$.

Здесь $I_p \in C^{p \times p}$ есть единичная матрица. Кроме того, через $(Q_j(z))$ обозначим решение уравнения $(Ju)_j = zu_j$, $j \geq 1$, с $Q_0(z) = O_p$ и $Q_1(z) = B_0^{-1}$. Матричные функции $P_j(z)$ и $Q_j(z)$ являются полиномами по комплексной переменной z степени j и $j-1$, соответственно с матричными коэффициентами.

Отметим, что $(P_j(z))$ и $(Q_j(z))$ называются матричными многочленами первого и второго рода, соответственно.

Следуя [10], определим граничную тройку $\Pi = \{H, \Gamma_0, \Gamma_1\}$ для T_{\max} , задав

$H = C_p$, $\Gamma_1 u = (Q(0)) * T_{\max} u - P_0 u$, $\Gamma_0 u = (P(0)) * T_{\max} u$,

где $u \in \text{dom}(T_{\max})$, а P_0 – ортопроектор.

Теорема 1. Пусть $\Pi = \{H, \Gamma_0, \Gamma_1\}$ – граничная тройка для T_{\max} , и пусть $M(\cdot)$ – соответствующая функция Вейля. Тогда крайновское расширение T_K совпадает с T_0 , то есть,

$$\text{dom}(T_K) = \text{dom}(T_{\max}) \setminus \ker(\Gamma_0) = \{u \in T_{\max} : (P(0)) * T_{\max} u = 0\}.$$

Теорема 2. Пусть $p = 1$. Пусть также $\Pi = \{H, \Gamma_0, \Gamma_1\}$ – граничная тройка для T_{\max} , заданного формулой (11), и пусть $M(\cdot)$ – соответствующая функция Вейля. Область определения Фридрихсова расширения задается формулой

$$\text{dom}(T_F) = \{u \in T_{\max} : (\Gamma_1 - \alpha \Gamma_0) u = 0\},$$

где α определяется формулой $\alpha := \lim_{j \rightarrow \infty} \frac{Q_j(0)}{P_j(0)}$.

Замечание 4. Следует отметить, что описание расширений Крейна и Фридрихса, приведенных в Теоремах 1 и 2 в скалярном случае, совпадает с описанием, полученным ранее Brown и Christiansen в [7].

Следствие 3. Пусть $p = 1$. Пусть $p = 1$. Пусть также $\Pi = \{H, \Gamma_0, \Gamma_1\}$ – граничная тройка для T_{\max} , и пусть $M(\cdot)$ – соответствующая функция Вейля. Множество

всех неотрицательных самосопряженных расширений T_h оператора T_{\min} параметризуется следующим образом

$$\text{dom}(T_h) = \{u \in \text{dom}(T_{\max}) : (\Gamma_1 - h\Gamma_0)u = 0\},$$

$$h \in [-\infty; \alpha].$$

$$\text{В частности, } \text{dom}(T_{-\infty}) = \{u \in \text{dom}(T_{\max}) : \Gamma_0 u = 0\}.$$

Литература

1. Yu. Arlinskii, E. Tsekanovskii, The von Neumann Problem for Nonnegative Symmetric Operators, *Integr. Equ. Oper. Theory* 51 (2005), 319–356.
2. Yu.M. Berezansky, *Expansions in eigenfunctions of self-adjoint operators*, Amer. Math. Soc., Providence, RI, 1968.
3. B. Malkolm Brown, J.S. Christiansen, On the Krein and Friedrichs extension of a positive Jacobi operator, *Expo. Math.* 23 (2005), 176–186.
4. V.A. Derkach, M.M. Malamud, Generalized resolvent and the boundary value problems for Hermitian operators with gaps, *J. Funct. Anal.* 95 (1991), no. 1, 1–95.
5. V.A. Derkach, M.M. Malamud, Characteristic Functions of Almost Solvable Extensions of Hermitian Operators, *Ukr. Math. J.* 44 (1992), no. 4, 435–459 (Russian).
6. V.A. Derkach, M.M. Malamud, The extension theory of Hermitian operators and the moment problem, *J. Math. Sci. (New York)* 73 (1995), 141–242.
7. M.G. Krein, The theory of self-adjoint extensions of semibounded Hermitian transformations and its applications, I, *Sb. Math.* 20 (1947), no. 3, 431–495 (Russian).

Секция № 2
Программная
инженерия



УДК 330.59:332.1

АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УСЛОВИЯХ ОГРАНИЧЕННОГО ВРЕМЕНИ И РОСТА НЕОПРЕДЕЛЁННОСТИ.

ЛЕБЕЗОВА Э. М.,

старший преподаватель кафедры информационных технологий ГОУ ВПО «Донецкая академия управления и государственной службы при Главе Донецкой Народной Республики». E-mail: ellis54@rambler.ru

Аннотация

Лебезова Э. М., Анализ эффективности методологий разработки программного обеспечения в условиях ограниченного времени и роста неопределённости. Данная статья содержит анализ методологий разработки и внедрения программного обеспечения – водопадной и семейства Agile, сравнение которых даст возможность выявить их слабые и сильные стороны и будет способствовать обеспечению более устойчивого управления ИТ-проектами в условиях ограниченного времени разработки и роста неопределенности.

Ключевые слова: ИТ-проект, управление проектами, методология разработки ПО, Waterfall-модель, Agile-модель.

Общая постановка проблемы. В отрасли управления ИТ-проектами на сегодняшний день вызрели три актуальные проблемы:

- Заказчик зачастую не в состоянии сформулировать четкие требования к ИТ-системе.

- Главное требование конкурентоспособности современного бизнеса - оперативное применение новых ИТ-технологий, что требует от исполнителя быстрой поставки наиболее приоритетной функциональности ПО путём минимизации времени появления очередного релиза.

- Процесс взаимодействия заказчика и разработчика ИТ-систем характеризуется как неудовлетворительный и малоэффективный.

В сложившейся ситуации следует отметить значимость дальнейшего исследования и усовершенствования моделей разработки программного обеспечения для отрасли управления ИТ-проектами, особенно в современных условиях быстро изменяющегося внешнего окружения и роста неопределенности.

Анализ последних исследований и публикаций. Исследования проблематики разработки качественного программного обеспечения в условиях ограниченного времени представляют большой академический и практический интерес не только для учёных, но и для участников разработки ПО, в особенности руководителей проектов. Многочисленные наработки

методик представлены в работах [1-3]. Однако гибкие методы на основе Agile-модели рассмотрены недостаточно и нуждаются в дальнейшем исследовании.

Исследования. С каждым годом растёт уровень зависимости компаний от ИТ-индустрии. С одной стороны, эффективность бизнеса напрямую зависит от автоматизации основных бизнес-процессов и быстрого внедрения новых технологий. С другой стороны, вероятность успеха ИТ-проектов крайне низка.

В отчеты компании Panorama Consulting Solutions за 2015 и 2016 годы [4-5] о внедрении ERP-систем попали данные 777 респондентов о сроках выполнения и освоении бюджета проектов. На рисунке 1 представлена диаграмма доли ERP-проектов, уложившихся в плановый бюджет, на второй диаграмме (рисунок 2) – доля проектов, закончившихся в плановый срок.

Средняя стоимость внедрения уменьшилась с 4,5 млн.\$ в 2015 году до 3,8 млн. в 2016 году, однако организации продолжают испытывать значительные перерасходы бюджета. В 2015 году перерасход был у 55% проектов, а в 2016 году превысили бюджет 77% проектов (рис. 1). В целом диапазон перерасхода средств остаётся в пределах рекомендуемого для таких проектов 20-процентного резерва.



Рисунок 1. Освоение бюджета внедрения ERP-систем по данным Panorama Consulting Solutions



Рисунок 2. Выполнение сроков внедрения ERP-систем по данным Panorama Consulting Solutions

25% в 2015 году и 43% в 2016 году ERP-проектов закончились в плановый срок (рис. 2). На первый взгляд виден очевидный прогресс. Однако если посмотреть на среднюю продолжительность проектов, то в 2015 году она была 14.3 месяца, а в 2016 году 21 месяц. Что указывает на то, что ИТ-компании просто увеличивали плановые сроки внедрения.

В отчете, который в 2015 году выпустила компания The Standish Group [7], предоставлены данные 50000 проектов по разработке и внедрению программных продуктов. Успеха добились - 29% проектов, в категорию «спорные проекты» (проект завершен и работает, но плановый бюджет или срок был превышен, либо реализован не весь запланированный функционал) попало 52%, а в категории провальных проектов (реализация которых не завершилась) оказалось 19%. Из этих данных видно, что спорные и провальные проекты составляют большинство (71%).

В таблицах 1-3 представлены показатели по внедрению ИТ-проектов (завершение в плановый срок, освоение планового бюджета и реализация всех запланированных функций).

Таблица 1. Превышение бюджета ИТ-проектов в % от планового значения (данные отчёта The Standish Group [6])

Превышение бюджета	% проектов
1	2
До 20%	15,5%
21 - 50%	31,5%
51 - 100%	29,6%
101 - 200%	10,2%
201 - 400%	8,8%
Свыше 400%	4,4%

Таблица 2. Отклонения ИТ-проектов от планового срока сдачи (данные отчёта The Standish Group [6])

Превышение срока	% проектов
1	2
До 20%	13,9%
21 - 50%	18,3%
51 - 100%	20,0%
101 - 200%	35,5%
201 - 400%	11,2%
Свыше 400%	1,1%

Таблица 3. Доля реализованного функционала ПО от запланированного (данные отчёта The Standish Group [6])

% реализованных функций	% проектов
1	2
До 25%	4,6%
25 - 49%	27,2%
50 - 74%	21,8%
75 - 99%	39,1%
100%	7,3%

Из таблицы 1 видно, что у 23,4% проектов бюджет был превышен на 100% и выше от планового. Данные таблицы 2 показывают, что сроки завершения почти у половины рассматриваемых проектов (47,8%), были увеличены более чем на 100%. Что касается реализованного функционала (таблица 3), то у более четверти проектов он составил лишь 25-49% от первоначально запланированного.

Становится очевидным тот факт, что эффективность проектов по разработке ПО ещё ниже, чем проектов по внедрению. Опрос руководителей о причинах провалов ИТ-проектов (рис. 3) показал, что основными причинами являются отсутствие вовлечённости заказчика в процесс разработки, а также неполнота и изменчивость требований.



Рисунок 3. Причины провалов ИТ-проектов по данным отчёта The Standish Group [7]

В новых программных проектах требования не всегда могут быть определены в начале разработки. Потребности и функционал выявляются в процессе работы над проектом. В ходе испытаний промежуточных версий как у заказчиков, так и у самих разработчиков возникает понимание необходимости новых свойств или изменения существующих свойств системы. Понятно, что такие изменения должны рассматриваться как

нормальная часть процесса разработки ПО. Однако классические модели разработки программного обеспечения настаивают на неизменяемости спецификаций в процессе работы над ИТ-проектом. Так каскадная (водопадная или waterfall) модель предполагает только пошаговое продвижение вперед по заранее установленным этапам (рис. 4).



Рисунок 4. Последовательность разработки согласно "классической" каскадной модели.

Следует отметить, что такая методология, при обнаружении ошибок, сделанных на ранних этапах работы, вызывает серьезные проблемы. Но еще большие проблемы возникают при изменениях внешнего окружения, в котором разрабатывается программное обеспечение (например, изменения политик, ролей, артефактов в разрабатываемой системе, внесение новых требований и изменений в отраслевые стандарты, появление на рынке конкурирующих продуктов и др.).

Разрабатывать ИТ-системы в соответствии с таким подходом очень трудно, так как требуется особенно тщательно собирать, анализировать и структурировать данные на начальных этапах, с тем чтобы впоследствии можно было пользоваться их результатами вне зависимости от вероятных изменений.

В “Четвертом Глобальном исследовании в области управления портфелями программ и проектов”, проведенном PwC [8] были выявлены и ранжированы факторы, которые делают проекты более успешными. Ключевой частью этого анализа была методология проектного управления Agile. В Agile-манифесте, который переведён на 68 языков мира, так характеризуются более совершенные методы разработки:

- Люди и взаимодействие важнее процессов и инструментов
- Работающий продукт важнее исчерпывающей документации
- Сотрудничество с заказчиком важнее согласования условий

контракта

- Готовность к изменениям важнее следования первоначальному плану” [9].

Agile – семейство гибких методологий разработки, включающее в себя несколько методик, каждая из которых заслуживает отдельного исследования. Это:

- Scrum - принципы организации взаимодействия между Бизнесом и ИТ;
- Kanban - принципы упорядочивания мультизадачности в работе сотрудника;
- XP - принципы экстремального программирования;
- Lean - принципы, относящихся к качеству, скорости и клиентоориентированности.

Более детальная природа Agile Project Management показана на рис. 5.

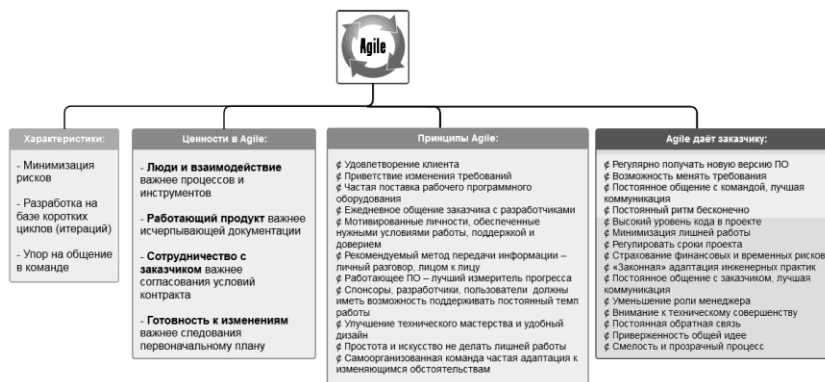


Рисунок 5. Характеристики, ценности и принципы методологий семейства Agile.

В “Четвертом Глобальном исследовании в области управления портфелями программ и проектов”, отмечается, что популярность применения методологии Agile для управления проектами повысилась с 2012 года на 11%, количество штатных специалистов аттестованных по методикам Agile, пока невелико, но также растёт. Более половины компаний полагают, что применение на практике Agile-методов в существенной мере способствует достижению установленных ими показателей результативности.

На вопрос: “На основании каких критериев Ваша компания делает выбор в пользу методологии Agile в сравнении с каскадными моделями (Waterfall)?”, респонденты (3025 компаний из 110 стран) отвечают - скорость выхода на рынок (50%) и возможность итеративной доработки в части объема работ и подхода к применению (43%) (рис. 6).

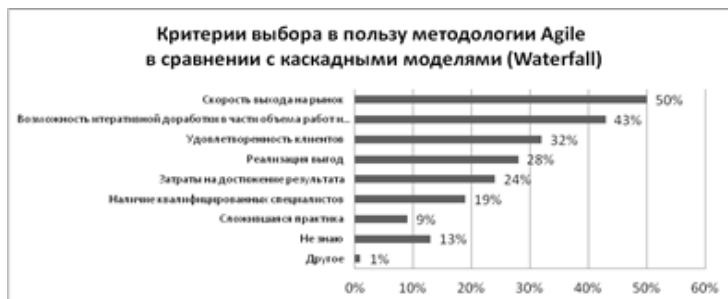


Рисунок 6. Критерии выбора респондентов “Четвертого Глобального исследования в области управления портфелями программ и проектов” в пользу методологии Agile в сравнении с каскадными моделями (Waterfall)

Независимо от Глобального исследования статистические исследования успешности проектов от «The Standish Group» в 2015 году показали, что процент полностью завершённых проектов выполненных согласно каскадной модели, катастрофически мал - 3% для больших проектов и 7% для средних. Только для небольших проектов этот показатель достаточно велик – 44% (таб. 4). У Agile показатель успешности на 28% выше, а показатель провальности проектов на 20% ниже. Что касается спорных проектов, здесь разница между Agile и каскадной моделью незначительна (52% и 60% соответственно).

Таблица 4. Успешность моделей проектирования по данным отчёта The Standish Group [7]

Масштаб проекта	Модель	Успешное завершение	Спорные проекты	Проект не завершился успешно
1	2	3	4	5
Все проекты	Agile	39%	52%	9%
	Водопадная модель	11%	60%	29%
Большие проекты	Agile	18%	59%	23%
	Водопадная модель	3%	55%	42%
Средние проекты	Agile	27%	62%	11%
	Водопадная модель	7%	68%	25%
Небольшие проекты	Agile	58%	38%	4%
	Водопадная модель	44%	45%	11%

Выводы и перспективы дальнейшего развития в этом направлении. На основании вышесказанного следует сделать следующие выводы:

1. В условиях быстро изменяющейся, неопределенной, сложной и неоднозначной внешней среды разработки ПО изменяются и сами методы разработки. Все большее значение приобретают гибкие, динамичные (Agile) подходы.

2. Разработка ПО должна допускать изменения, так как непрерывные изменения приводят к минимизации ошибок на ранних стадиях проекта и позволяют клиентам и заказчикам эффективно взаимодействовать.

3. Популярность agile-подходов в проектах по разработке и внедрению программного обеспечения продолжают расти. Появление Agile как глобального движения выходит за рамки разработки ПО, так как её преимущества вполне можно применить и для проектов не связанных с ИТ. Тем более, что очевидной тенденцией последних лет является рост неопределенности как в деловых так и в государственных процессах, а государственных учреждений и госкомпаний, заинтересованные в выстраивании эффективных гибких моделей управления, становится всё больше.

Список литературы

1. Литвак Е.Г. Оценка сложности разработки программного обеспечения в ИТ-проектах с ограниченными ресурсами // Материалы II Всероссийской научно-практической конференции «Информационные технологии в экономике и управлении». 2016. Махачкала. С. 87-91

2. Брагина Т. И., Табунщик Г. В. Сравнительный анализ итеративных моделей разработки программного обеспечения // Радиоэлектроника, информатика, управления. 2010. №2 (23). URL: <http://cyberleninka.ru/article/n/sravnitelnyu-analiz-iterativnyh-modeley-razrabotki-programmnogo-obespecheniya> (дата обращения: 31.03.2017).

3. Николаенко В.С. Разработка принципов управления ИТ-проектом // Вестн. Том. гос. ун-та. 2015. №390. URL: <http://cyberleninka.ru/article/n/razrabotka-printsipov-upravleniya-it-proektom-1> (дата обращения: 31.03.2017).

4. 2016 REPORT ON ERP SYSTEMS AND ENTERPRISE SOFTWARE - Panorama Consulting Solutions. [Электронный ресурс]. – Режим доступа: <http://go.panorama-consulting.com/rs/panoramaconsulting/images/2016-ERP-Report.pdf>

5. 2015 REPORT ON ERP SYSTEMS AND ENTERPRISE SOFTWARE - Panorama Consulting Solutions. [Электронный ресурс]. – Режим доступа: <http://go.panorama-consulting.com/rs/panoramaconsulting/images/2015%20ERP%20Report.pdf>

6. CHAOS Report 2014 - The Standish Group. [Электронный ресурс]. – Режим доступа: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
7. CHAOS Report 2015 - The Standish Group. [Электронный ресурс]. – Режим доступа: <https://www.infoq.com/articles/standish-chaos-2015>
8. Четвертое Глобальное исследование в области управления портфелями программ и проектов - PwC. [Электронный ресурс]. – Режим доступа: https://www.pwc.ru/ru/riskassurance/publications/assets/global_ppm_survey_russia_n.pdf
9. Agile-манифест разработки программного обеспечения [Электронный ресурс]. – Режим доступа: <http://agilemanifesto.org/iso/ru/manifesto.html>

УДК 004.056.5

ПОДСИСТЕМА ЗАЩИТЫ АВТОРСКОГО ПРАВА В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

А.В. Чернышова, Д.В. Кубашевский

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
alla@donntu.org, dehax12@gmail.com

Аннотация

Чернышова А.В., Кубашевский Д.В. Подсистема защиты авторского права в программном обеспечении. Выполнен анализ проблемы защиты авторского права на программное обеспечение и описана работа программной подсистемы защиты ПО с использованием криптографических алгоритмов шифрования.

Ключевые слова: система лицензирования, программное обеспечение, криптография, клиент-серверная архитектура, база данных, лицензия.

Постановка проблемы. В настоящее время нелегальное копирование и распространение программного обеспечения наносит огромный ущерб разработчику. Использование такого пиратского ПО не приносит прибыли. В большинстве случаев, если программное обеспечение содержит подсистему защиты от нелегального копирования, то разработке такой подсистемы уделяется недостаточно сил. Спустя некоторое время хакеры взламывают такую ненадёжную защиту, и программный продукт попадает на рынок пиратского ПО [1]. В связи с этим возникает необходимость реализации подсистемы защиты таким образом, чтобы программный продукт не мог

функционировать без подтверждения лицензии, а процесс взлома такой подсистемы был максимально затруднён.

Цель статьи — описать работу разработанной подсистемы защиты авторского права в программном обеспечении, продемонстрировав её функциональные возможности.

Описание подсистемы защиты. Современные подсистемы защиты ПО подразумевают активацию программного продукта (с целью подтверждения права на его использование) посредством взаимодействия с сервером лицензий. После проверки правильности лицензионных данных, предоставленных пользователем через пользовательский интерфейс подсистемы защиты, сервер должен дать подтверждение, что лицензия действительна, и подсистема защиты открывает доступ пользователю к функциональным возможностям ПО. Однако данная идея защиты содержит существенный недостаток — можно реализовать собственный сервер лицензий, который всегда будет подтверждать действительность лицензии, или модифицировать саму подсистему защиты таким образом, чтобы проверка активации программного продукта даже не осуществлялась.

Надёжная подсистема защиты должна быть основана не на предоставлении права на функционирование ПО, а на предоставлении некоторых функциональных возможностей ПО, то есть некоторой части программного кода, без которой продукт не может работать и выполнять важные для пользователя функции. В этом случае ни модификация подсистемы защиты, ни реализация пиратского сервера лицензий не позволяют получить доступ к необходимым функциям программы.

Схема функционирования разработанной подсистемы защиты представлена на рисунке 1.

После приобретения программного продукта в базе данных сервера добавляется запись о лицензии. Затем пользователь получает файл лицензии и использует его для активации программы. Пример содержимого файла лицензии приведён на рисунке 2.

Лицензия представляет собой файл формата XML [2] с расширением «.dls». Тег «HardwareId» содержит MD5-хеш [3] серийного номера первого жёсткого диска компьютера. Тег «Version» содержит версию ПО, для которой эта лицензия действительна. В случае обновления ПО пользователю необходимо будет получить новую лицензию. Тег «Files» содержит набор файлов программы в корневом каталоге программы (имя и MD5-хеш файла). В случае обнаружения несоответствия имени или хеша файла подсистема защиты уведомляет пользователя о повреждении файлов программы.

Теги «ActivateDate» и «ExpiryDate» позволяют контролировать начальную и конечную даты, когда лицензия может быть использована. Теги «Signature» и «PublicKey» содержат электронную цифровую подпись

(ЭЦП) [4] данных лицензии и публичный ключ для её проверки соответственно.

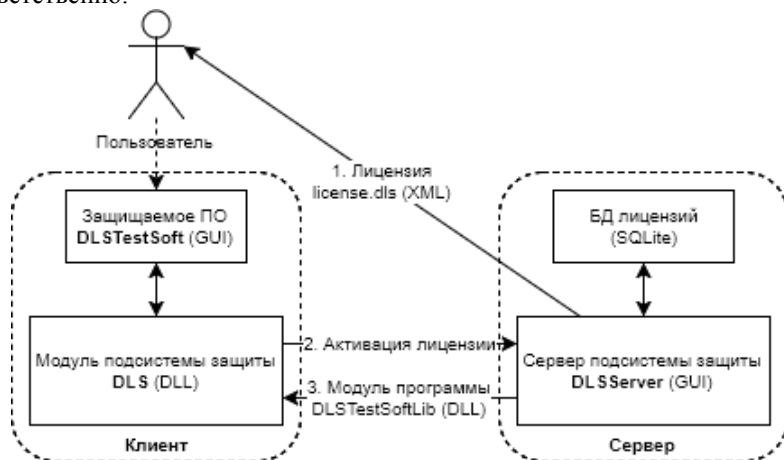


Рисунок 1 – Схема функционирования подсистемы защиты

```

<DLS>
  <Id>2</Id>
  <Name>Denis Kubashevskiy</Name>
  <Email>dehax12@gmail.com</Email>
  <HardwareId>0oAJwuz+4bdyNFguagNjhg==</HardwareId>
  <Version>1.0.0.0</Version>
  <Files>
    <File Name="log4net.dll" Hash="9ktzPq5EymZiFzhtwg8r8A==" />
  </Files>
  <ActivateDate>2017-01-01</ActivateDate>
  <ExpiryDate>2018-01-01</ExpiryDate>
  <Signature>be20fuNz+kFTPIzzX77a7MuyYRk8UfcEqgqa5MNUvphY2F==</Signature>
  <PublicKey>BgIAAAAIABEU1MxAAQAAN0oIQF8dL3YE+wgqQ/eJweoRR==</PublicKey>
</DLS>
  
```

Рисунок 2 – Пример файла лицензии

При первом запуске защищаемого программного обеспечения подсистема защиты запрашивает у пользователя файл лицензии. Пользователь предоставляет файл, после чего подсистема защиты проверяет данные лицензии и ЭЦП. Если проверка пройдена успешно, подсистема защита обращается к серверу для подтверждения лицензии и получения модуля полнофункционального режима работы программы. Пример работы тестового

защищаемого ПО в полнофункциональном режиме после активации лицензии представлен на рисунке 3.

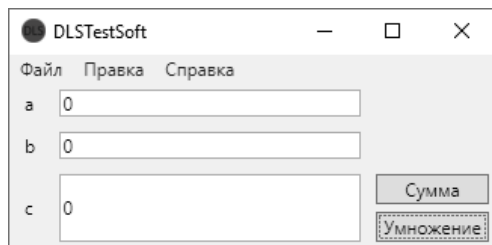


Рисунок 3 – Полнофункциональный режим после активации лицензии

В случае отсутствия Интернет-соединения или при ошибке проверки лицензии защищаемое ПО может продолжить работу, но только в функционально ограниченном демонстрационном режиме. Пример работы программы в функционально ограниченном демонстрационном режиме представлен на рисунке 4.

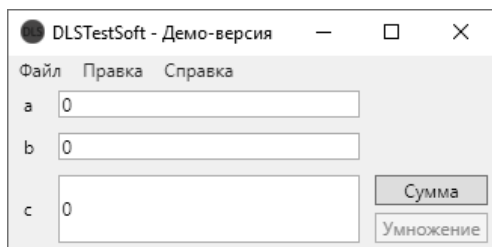


Рисунок 4 – Ограниченный демонстрационный режим

В базе данных сервера подсистемы защиты содержится информация о пользователях ПО в таблице «Users», о версиях ПО в таблице «Versions», о файлах в корневом каталоге программы в таблице «Files» и о лицензиях на использование ПО в таблице «Licenses». Схема базы данных приведена на рисунке 5.

Согласно такой организации БД у каждого пользователя может быть несколько лицензий, которые могут, в свою очередь, относиться к нескольким версиям ПО. Для каждой версии ПО имеется определённый набор файлов в корневом каталоге программы. Также каждая версия содержит код защищённого программного модуля, который представлен в таблице «Versions» полем «ProtectedCode».

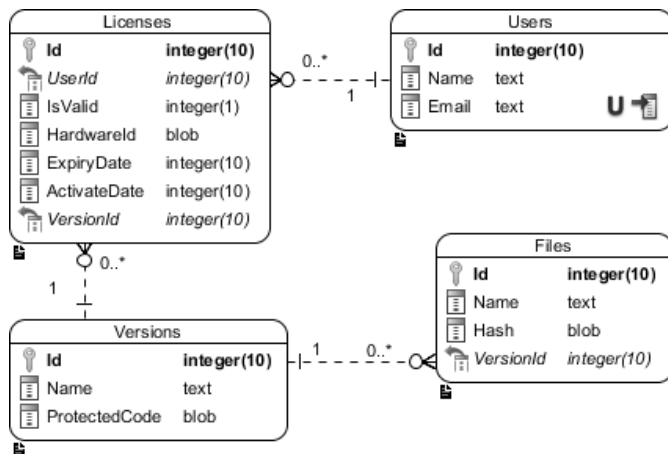


Рисунок 5 – Схема базы данных сервера лицензий

Выводы. Выполнен анализ проблемы защиты авторского права на программное обеспечение. Описана работа и функциональные возможности разработанной программной подсистемы защиты ПО, использующей криптографические алгоритмы шифрования. На данный момент реализованная подсистема защиты предусматривает работу с одним программным продуктом, однако, в дальнейшем планируется расширение до системы защиты ПО, которая будет работать с несколькими программными продуктами и поддерживать распределённые базы данных [5] для оптимизации взаимодействия модуля системы защиты с серверами лицензий и уменьшения на них нагрузки при большом количестве пользователей защищаемого ПО.

Список литературы

1. Нарушение авторского права [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: https://ru.wikipedia.org/wiki/Нарушение_авторского_права
2. XML [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: <https://ru.wikipedia.org/wiki/XML>
3. MD5 [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: <https://ru.wikipedia.org/wiki/MD5>
4. Электронная подпись [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: https://ru.wikipedia.org/wiki/Электронная_подпись
5. Распределённая база данных [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: https://ru.wikipedia.org/wiki/Распределённая_база_данных

УДК 004.056

ЗАЩИТА ИНФОРМАЦИИ В ПРИЛОЖЕНИЯХ ДЛЯ МГНОВЕННОЙ ПЕРЕДАЧИ СООБЩЕНИЙ

О.С. Грищенко, А.В. Чернышова

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
alla@donntu.org
olleg1996@mail.ru

Аннотация

Грищенко О.С., Чернышова А.В. Защита информации в приложениях для мгновенной передачи сообщений. Представлен обзор существующих мессенджеров, поддерживающих защиту на уровне сообщений, сформированы требования к авторской программе-мессенджеру. Проведено проектирование и реализация авторской программы-мессенджера.

***Ключевые слова:** мессенджер, шифрование, криптографический протокол, защита данных.*

Постановка проблемы. Решение о создании системы было принято в связи с актуальностью проблемы защиты персональных данных и востребованностью приложений для защищенного обмена сообщениями. Необходимость обеспечения безопасности персональных данных в наше время очень важна. Информация всегда имела большую ценность, но в связи со стремительным развитием информационных технологий защита личных данных выходит на новый уровень. Из-за этого приложения, созданные для защиты данных, сейчас обретают все большую популярность.

Цель работы – разработка приложения для обмена мгновенными сообщениями, предоставляющего защиту пользовательских данных и удовлетворяющего современным требованиям к безопасности пользовательской переписки. Проектирование приложения осуществляется на основе информации, полученной при анализе наиболее защищенных существующих приложений. Для реализации защиты данных было решено использовать стандартные криптографические алгоритмы шифрования и электронной цифровой подписи.

Обзор существующих мессенджеров, поддерживающих защиту на уровне сообщений

Chatsecure [1] – приложение для обмена мгновенными сообщениями с поддержкой шифрования. Реализован для iOS и Android (более не поддерживается). Разработчиком является GuardianProject. Шифрование

обеспечивается с помощью протокола OTR (Off-the-Record)[2]. Передача сообщений осуществляется по протоколу XMPP[3]. ChatSecure не сохраняет отправленные сообщения в памяти устройства. ChatSecure также позволяет организовывать групповые чаты. Приложение позволяет передавать мультимедиа-сообщения, фотографии и файлы. ChatSecure является бесплатным open source проектом.

CryptoCat [4] – приложение для обмена мгновенными сообщениями с поддержкой шифрования. Является открытым. Может работать в браузерах Chrome, Firefox, Safari и Opera, а также в качестве отдельного клиента в Apple OS X и на iPhone. Использует шифрование данных через протокол OTR и передает зашифрованные данные по XMPP. Сообщения шифруются с помощью AES-256[5]. CryptoCat предоставляет пользователям возможность обмениваться документами и мультимедийными файлами (при этом файлы шифруются с помощью AES-256). Сообщения не хранятся и после окончания диалога удаляются автоматически в течение нескольких минут. При повторном входе в тот же диалог никто из участников не сможет просмотреть ранее отправленные сообщения. Файлы хранятся в течение 30 дней.

Telegram [6] – кроссплатформенный мессенджер, позволяющий обмениваться текстовыми сообщениями и файлами размером до 1,5 Гб. Обеспечивает защиту данных с помощью протокола MTProto, который предполагает использование нескольких криптографических протоколов. При авторизации и аутентификации пользователя приложение использует алгоритмы RSA-2048[7] и Диффи-Хеллмана [8]. Для шифрования сообщений использует AES. В режиме секретных чатов реализует end-to-end шифрование[9]. Также позволяет организовывать аудио и видео конференции.

Signal[10] позволяет обмениваться сообщениями и звонками с абонентами из телефонной книги мобильного устройства. Приложение предоставляет возможность отправлять переписки в архив. Приложение обеспечивает криптографическую защиту телефонных разговоров от прослушки, записи и перехвата. Для шифрования данных используется алгоритм AES с 256-битным ключом. Signal способно обеспечить шифрование голосовой связи на всем пути прохождения сигнала. Передача данных осуществляется через Wi-Fi подключение или мобильное соединение. Для защиты от прослушки на экране отображаются два случайных слова, которые должен повторить собеседник.

TextSecure[10] – мессенджер для безопасной передачи мгновенных сообщений, файлов и мультимедиа данных. TextSecure использует открытый протокол Signal, который реализует end-to-end шифрование данных. Ключи, используемые для шифрования, хранятся только на устройстве пользователя. Для проверки подлинности собеседника предусмотрено ручное сравнение отпечатков ключей на устройствах, либо сканирование QR-кода. TextSecure предоставляет также возможность создания групповых чатов,

которые также шифруются. При этом серверы не имеют доступа к метаданным группы.

Проектирование авторской программы-мессенджера

После анализа существующих защищенных мессенджеров было принято решение о разработке нового мессенджера с поддержкой защиты передаваемых данных. Основываясь на анализе механизмов защиты существующих продуктов, к разрабатываемому приложению были предъявлены следующие требования:

- обеспечение надежного механизма согласования ключей;
- шифрование всех передаваемых данных;
- хранение сообщений в зашифрованном виде;
- отсутствие доступа к передаваемым данным при передаче через сервер;
- проверка целостности сообщений.

На основе описанных требований было произведено проектирование разрабатываемой системы.

На рисунке 1 представлена диаграмма вариантов использования разрабатываемого приложения.

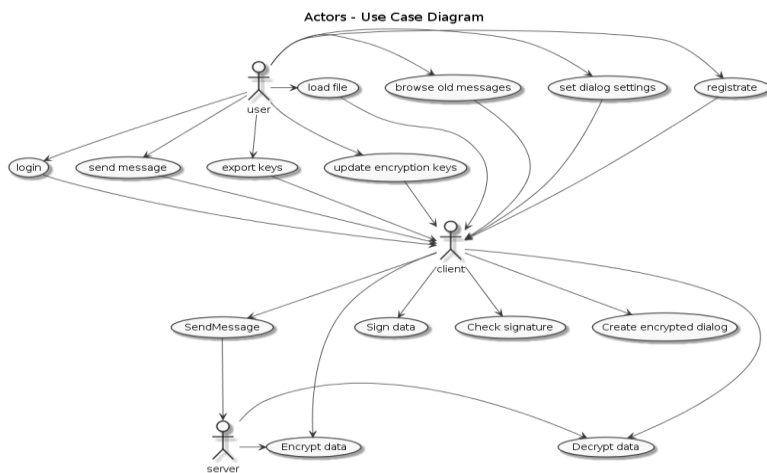


Рисунок 1 – Диаграмма вариантов использования.

Приложение использует собственный протокол передачи данных. Для согласования ключей протокол использует алгоритм RSA. Для шифрования передаваемых данных используется алгоритм AES, для проверки целостности сообщений используется электронно-цифровая подпись. Протокол предполагает два уровня шифрования: шифрование канала между клиентом и сервером и шифрование передаваемых сообщений непосредственно между

клиентами. Таким образом, сообщения клиентов будут храниться на сервере в зашифрованном виде, а ключи для расшифровки сообщений будут находиться только на пользовательских компьютерах. На транспортном уровне используется протокол TCP.

Разрабатываемое приложение состоит из трех частей: клиента, сервера и библиотеки, инкапсулирующей работу с протоколом передачи данных.

На рисунке 2 представлен фрагмент диаграммы классов приложения, содержащий классы, обеспечивающие защиту и передачу данных, а также их основные методы и атрибуты.

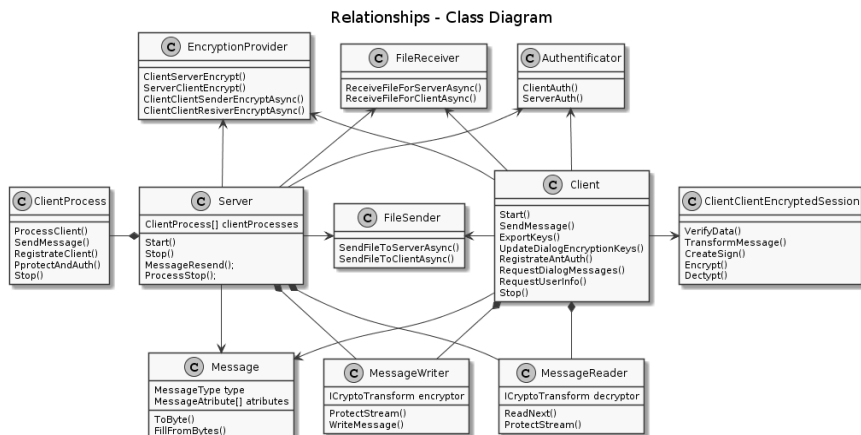


Рисунок 2 - Фрагмент диаграммы классов.

Для серверной части приложения была разработана база данных и использован MS SQL Server. Схема данных базы данных представлена на рисунке 3.

На данный момент приложение реализует следующую функциональность:

- создание защищенных диалогов;
- отправка зашифрованных сообщений с ЭЦП;
- просмотр ранее отправленных сообщений;
- передача зашифрованных файлов с ЭЦП;
- защищенная авторизация и аутентификация пользователя;
- обновление ключей шифрования диалога;
- регистрация пользователя.

Выводы. Был проведен обзор существующих мессенджеров, поддерживающих защиту на уровне сообщений. Были выделены требования к разрабатываемому продукту. В рамках данной работы было проведено

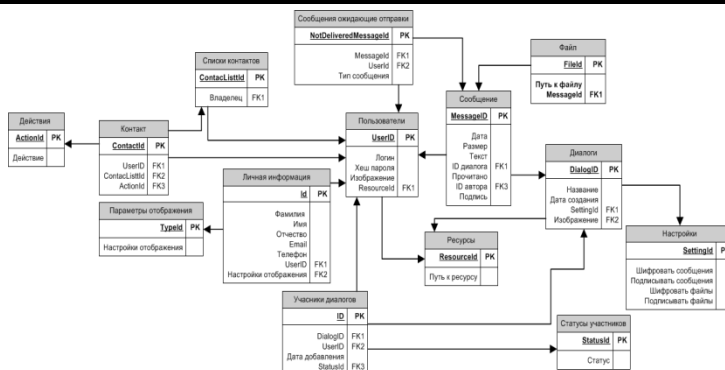


Рисунок 3 – Схема базы данных сервера.

проектирование программы-мессенджера и реализация ее основной функциональности. В дальнейшем планируется расширить функциональность приложения возможностью организации защищенных групповых чатов, а реализовать защищенные аудио и видео звонки.

Список литературы

1. Chatsecure// hatsecure.org [Электронный ресурс]. – Режим доступа: <https://chatsecure.org/>
2. Off-the-RecordMessaging// Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Off-the-Record_Messaging.
3. XMPP // Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/XMPP>
4. Cryptocat // crypto.cat [Электронный ресурс]. – Режим доступа: <https://crypto.cat/>
5. AdvancedEncryptionStandard// Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Advanced_Encryption_Standard.
6. Telegram // tlgrm.ru. [Электронный ресурс]. – Режим доступа: <https://tlgrm.ru/techfaq>
7. RSA // Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/RSA>
8. Протокол Диффи—Хеллмана// Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Протокол_Диффи_—_Хеллмана
9. End-to-endencryption// Википедия [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/End-to-end_encryption
10. Open whisper systems// whispersystems.org. [Электронный ресурс].– Режим доступа: <https://whispersystems.org/>

УДК 004.043

ИСПОЛЬЗОВАНИЕ СРЕДСТВ РЕФЛЕКСИИ ЯЗЫКА JAVA ПРИ РЕАЛИЗАЦИИ ПРОТОКОЛОВ ПРИКЛАДНОГО УРОВНЯ

В. А. Полетаев, А.С. Кузьмичева

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
poletaev.vladislav@gmail.com

Аннотация

Полетаев В.А., Кузьмичева А.С. Использование средств рефлексии языка Java при реализации протоколов прикладного уровня. Рассмотрены возможные методы реализации приложений, осуществляющих обмен информацией с удаленными устройствами. Обосновано использование Java Reflection API для организации отношений компонентов системы.

Ключевые слова: гибкая архитектура, средства рефлексии, сериализация, обработка данных

Общая постановка проблемы. Разнообразие задач, которые требуют использования средств передачи данных в компьютерных сетях, приводит к необходимости разработки и реализации новых сетевых протоколов, наилучшим образом соответствующих требованиям программного продукта. При этом в связи с необходимостью дальнейшей поддержки протокола и возможными изменениями требований и используемых средств передачи данных возникает вопрос о гибкости архитектуры реализации протокола, то есть возможности быстро вносить изменения в структуру и формат передаваемых сообщений.

Это требование можно удовлетворить, если на этапе проектирования разработать модульную архитектуру с низкой сопряженностью компонентов [1]. В случае если модуль реализует некоторый публичный интерфейс и инкапсулирует большую часть своей логики, то внесение изменений в модуль, не приведет к необходимости изменения других компонентов системы, или эти изменения окажутся минимальными.

Рассмотрим возможный вариант проектирования компонента системы, выполняющего обмен сообщениями с удаленным приложением (рис. 1).

Функция обмена информацией с удаленным устройством во многих случаях может быть реализована средствами стандартной библиотеки многих современных языков программирования. Для этого могут быть использованы транспортные протоколы стека TCP/IP, беспроводная передача данных с использованием Bluetooth, передача данных по проводному соединению (USB

и др.) или передача информации в пределах одной системы средствами межпроцессного взаимодействия. Большинство этих способов подразумевают передачу сообщений в виде последовательности октетов.



Рисунок 1 – Многоуровневая архитектура приложения

Рассмотрим процесс преобразования данных в формат, удобный для обработки другими модулями приложения. Во многих случаях, невозможно разработать способ отображения данных на единственную структуру в памяти, которая могла бы быть использована для обработки любого сообщения.

Проблемы отображения данных на динамические структуры данных. Одним из способов решения этой проблемы является десериализация сообщения в некоторую динамическую структуру данных, которая позволяет установить соответствие между кодом поля сообщения и его значением. Для этого необходимо разработать базу данных, содержащую коды сообщений и типы полей. Пример фрагмента базы данных в формате YAML, которая содержит информацию о сообщениях протокола, приведен на рисунке (рис. 2).

```

name: initialization
key: 1
states: [initialization]
side: client
properties:
- name: authorization_key
  key: 1
  type: byte_array
  size: 32
- name: version
  key: 2
  type: int
- name: password
  key: 3
  type: string
  optional: true
---
```

```

name: initialization_responce
```

Рисунок 2 – Фрагмент формата базы данных сообщений протокола

Этот подход позволяет успешно абстрагироваться от средства взаимодействия, однако он обладает существенными недостатками:

- при обращении к полям сообщения программист должен помнить символьные идентификаторы полей. Проверка корректности идентификаторов не производится на этапе компиляции и может привести к возникновению ошибок;

- при изменении структуры сообщения невозможно использование средств реорганизации кода, доступных в современных IDE;

- при использовании объектно-ориентированного подхода и представлении полученных данных в виде объектов, в системе образуется два экземпляра схемы данных, синхронизация которых требует дополнительных затрат и может привести к возникновению ошибок которые не будут обнаружены на этапе компиляции.

Этих проблем можно избежать путем объединения информации о структурах данных и соответствующем им формате передачи. Для этого необходимо использовать средства рефлексии, доступные во многих современных языках программирования.

Сериализация данных. Средства рефлексии Java позволяют определить для любого класса, интерфейса или члена класса метаданные, которые могут быть получены и обработаны программой на этапе выполнения [2]. Эта возможность позволяет объединить информацию базы данных, описанной выше с определениями полей класса, на которые данные сообщения будут отображены. При отправке сообщения, подлежащего сериализации, средствами интроспекции программа может определить тип сериализуемых данных и получить информацию об аннотированных полях класса.

Для десериализации сообщения необходимо определить, какой класс соответствует конкретному типу сообщения. Для этого в формат каждого сообщения можно добавить код типа. Классу, соответствующему этому типу можно прикрепить аннотацию, параметром которой является код типа. Это позволяет установить взаимно-однозначное соответствие между сообщением и классом.

Для преобразования данных в формат сообщения необходимо определить алгоритм сериализации и десериализации для каждого используемого типа данных. При этом класс, инкапсулирующий эту операцию можно определить на этапе выполнения, используя средства рефлексии.

Реализация модуля сериализации. Рассмотрим пример реализации модуля сериализации, организованного с использованием рефлексии и интроспекции.

Программные модули, реализующие сериализацию и десериализацию типов данных, могут быть организованы в виде классов, связанных

отношением обобщения с абстрактным классом `ByteConverter`. Эти классы должны переопределить абстрактные методы `serialize` и `deserialize`. Они получают и записывают последовательность октетов, используя потоки вывода и ввода соответственно. Для определения типа данных, которые подлежат преобразованию, класс-наследник должен сопровождаться одной или несколькими аннотациями `@SerializableType`, параметром которой является тип сериализуемого значения.

Диаграмма, иллюстрирующая фрагмент модуля сериализации, включающего поддержку целочисленного типа представлена на рисунке (рис. 3). Стандарт UML не поддерживает добавление метаданных к элементам диаграммы классов, однако существуют исследования, направленные на разработку возможных расширений стандарта, обеспечивающих возможность проектирования приложений с использованием аннотаций Java и атрибутов .NET Framework [3]. Один из этих вариантов используется в тексте данной статьи.

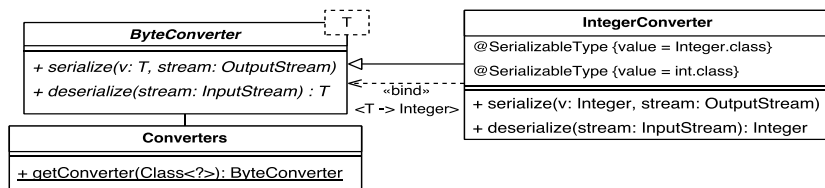


Рисунок 3 – Диаграмма модуля сериализации, включающего поддержку целочисленного типа

Сериализация и десериализация скалярных типов, таких как целочисленные переменные, значения с плавающей запятой, строки происходит чтением или записью в поток данных определенного количества октетов, в зависимости от типа.

Преобразование составных типов может быть реализовано вызовом метода сериализации или десериализации для каждого элемента типа. Если количество элементов переменное, то к структуре сообщения добавляется поле, содержащее количество элементов. Если количество известно на этапе проектирования, то его можно указать в качестве параметра аннотации свойства и не добавлять к двоичному представлению данных.

Необходимо отметить, что информация о параметрах типов в Java удаляется на этапе компиляции. Это связано с использованием механизма стирания типов (`type erasure`) [4], обеспечивающего обратную совместимость с программными модулями, разработанными до введения параметризованных типов в стандарт Java. Проблема невозможности определения параметра типа на этапе компиляции решается передачей типа в конструктор создаваемого модуля сериализации.

Не составляет труда реализовать метод, выполняющий поиск класса, инкапсулирующего логику сериализации для заданного типа. Аналогичным образом можно реализовать метод поиска класса сообщения по его коду.

Обработка сообщений. Средства рефлексии могут быть использованы в модуле, реализующем бизнес-логику приложения, при обработке полученных сообщений. Для поиска программного модуля, который отвечает за обработку полученных данных можно использовать аннотации, добавленные к методам класса-обработчика. Пример диаграммы, демонстрирующей реализацию модуля обработки сообщений, представлен на рисунке (рис. 4).

Введение дополнительных метаданных позволит реализовать новые правила бизнес-логики: проверки на корректность данных, автоматической обработки данных без необходимости дублирования кода.

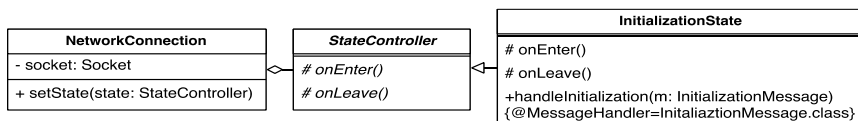


Рисунок 4 – Диаграмма реализации модуля обработки сообщений

Выводы. На основании анализа существующих программных решений, обеспечивающих сериализацию и десериализацию данных, разработанных для выполнения на платформе Java (Google Gson, XStream), встроенных средств сериализации платформы .NET, а также решений, обеспечивающих использование событийно-ориентированной архитектуры (Flask, PyQt) были определены преимущества использования метаданных в виде аннотаций Java при разработке программного обеспечения.

В статье была продемонстрирована возможность использования средств рефлексии Java при проектировании модульной архитектуры программного продукта с низкой сопряженностью компонентов для модуля, реализующего взаимодействие с удаленным процессом.

Приведенные в статье методы организации программного решения могут быть использованы в приложениях, написанных на других языках программирования высокого уровня, таких как C#, Python и др.

Список литературы

1. Макконнелл С. Совершенный код. Мастер-класс / С. Макконнелл / Пер. с англ. — М.: Издательство «Русская редакция», 2010. — 896 стр.: ил.4
2. Trail: The Reflection API (The Java™ Tutorials) [Electronic resource] / Интернет-ресурс. – URL: <https://docs.oracle.com/javase/tutorial/reflect/>. - Загл. с экрана.

3. Vasian Cera Representing Explicit Attributes in UML / Vasian Cera, Sven Kloppenburg // 7th International Workshop on Aspect-Oriented Modeling: материалы VII международной конференции, 2 октября 2005г., г. Montego Bay, Jamaica. –, 2005.

4. Type Erasure (The Java™ Tutorials > Learning the Java Language > Generics (Updated)) [Electronic resource] / Интернет-ресурс. - URL: <https://docs.oracle.com/javase/tutorial/java/generics/erasure.html>. - Загл. с экрана.

УДК 004.056

КЛИЕНТ-СЕРВЕРНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ БЕЗОПАСНОГО ОБЛАЧНОГО ХРАНИЛИЩА ФАЙЛОВ

В.Е. Лебедев, А.В. Чернышова

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
alla@donntu.org, alant.1301@gmail.com

Аннотация

Лебедев В.Е., Чернышова А.В. Клиент-серверное программное обеспечение безопасного облачного хранилища файлов. В данной работе представлен обзор существующих облачных хранилищ, поддерживающих шифрование файлов, сформированы требования к авторскому программному продукту. Проведено проектирование и реализация авторской программы.

Ключевые слова: облачное хранилище, шифрование, клиент, сервер, защита данных.

Постановка проблемы.Целесообразность создания подобной системы заключается в том, что несмотря на наличие аналогов, готовые решения не удовлетворяют всем требованиям, выдвигаемым пользователями. Помимо обеспечения доступности данных в любое время и в любом месте, информация пользователя должна быть защищена от кражи/подмены. На данный момент существуют средства для реализации подобных систем, а в связи с повсеместным распространением облачных технологий подобное программное обеспечение будет востребовано [1].

Цель работы – разработка клиент-серверного приложения, реализующего возможность облачного хранилища [2], обеспечивающего защиту пользовательских данных с использованием шифрования и электронно- цифровой подписи. Проектирование приложения осуществляется на основе информации, полученной при анализе существующих программ-

аналогов. Для реализации будут использоваться стандартные криптографические алгоритмы.

Обзор существующих облачных хранилищ, поддерживающих шифрование пользовательских данных

Dropbox [3] — файловый хостинг компании DropboxInc., включающий персональное облачное хранилище, синхронизацию файлов и программу-клиент. Dropbox позволяет пользователям создать специальную папку на своих компьютерах, которую Dropbox синхронизирует таким образом, что она имеет одинаковое содержимое независимо от того, какое устройство используется для просмотра[6]. Файлы, размещённые в этой папке, также доступны через веб-сайт Dropbox и мобильные приложения. Dropbox в связке с BoxCryptor, который надёжно шифрует файлы перед синхронизацией их с облаком, обеспечивает конфиденциальность данных в Dropbox. Данный сервис предлагает тарифы, представленные в таблице 1.

Таблица 1 – Размер облачного хранилища Dropbox.

Ежемесячная плата	Бесплатно	\$9,99	\$15
Объем	2-16 ГБ	1 ТБ	безлимит

MEGA (MEGA Encrypted Global Access) [4] — файлообменник КимаДоткома. Открыт 19 января 2013 года. Особенностью данного сервиса является то, что Mega шифрует весь контент прямо в браузере с помощью алгоритма AES [5]. Пользователи могут передавать друг другу файлы в зашифрованном виде, при этом все данные хранятся в «облаке». Ключи доступа к файлам не публикуются в открытом доступе, а распространяются по схеме Friend-to-Friend, между доверяющими друг другу пользователями. Однако сервис критиковался экспертами по криптографии из-за фактического отсутствия доверенного объекта на стороне пользователя. Хотя шифрование происходит на стороне клиента при помощи JavaScript, этот код каждый раз заново загружается с сервера и в любой момент может быть подменен без предупреждения, например, в результате взлома сервера третьей стороной. Данный сервис предлагает тарифы, представленные в таблице 2.

Таблица 2 – Размер облачного хранилища MEGA.

Ежемесячная плата	Бесплатно	€9,99	€19,99	€29,99
Объем	50 ГБ	500 ГБ	2 ТБ	4ТБ

Проектирование авторской программы

Проанализировав существующие системы облачного хранения данных, оказалось, что только в МЕГАреализовано шифрование файлов перед отправкой на сервер, а также VoxСтуртреализует защиту файлов перед отправкой в Dropbox.

После анализа были сформулированы следующие требования:

- клиент, реализующая управление файлами, загрузку/скачивание файлов;
- возможность структурирования файлов по папкам;
- разграничение прав доступа для разных пользователей;
- шифрование файлов перед отправкой их на сервер;
- выделение квот на загрузку файлов.

На основе описанных требований было произведено проектирование разрабатываемой системы.

На рисунке 1 представлена диаграмма вариантов использования разрабатываемого приложения.

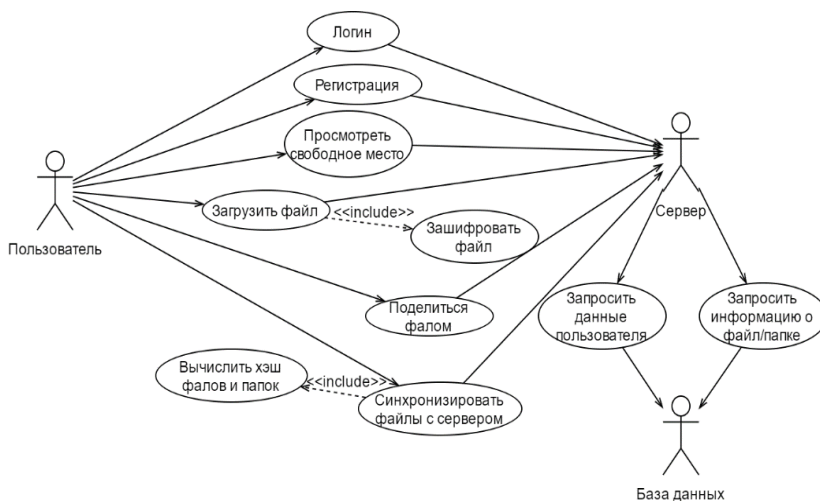


Рисунок 1 – Диаграмма вариантов использования.

Для передачи данных используется протокол HTTP. Для обмена ключами используется собственный алгоритм. Для шифрования ключа используется алгоритм RSA [6]. Для шифрования передаваемых данных используется алгоритм AES. В качестве алгоритма электронно-цифровой подписи используется DSA [7]. Спроектированная система предполагает шифрование

всего трафика между клиентом и сервером. Для этого реализован алгоритм обмена ключами, включающий:

- генерацию RSA-пары ключей;
- отправку публичного ключа;
- генерацию AES-ключа для шифрования трафика;
- отправку на клиент AES-ключа, зашифрованного открытым RSA-ключом;
- сохранение расшифрованного ключа на клиенте для последующего шифрования трафика от клиента.

Файлы перед загрузкой на сервер шифруются и хранятся на сервере в зашифрованном виде.

Разрабатываемое приложение состоит из трех частей: клиента, сервера и библиотеки, инкапсулирующей криптографические функции.

Для работы сервера была спроектирована база данных с использованием СУБД MySQL. Схема базы данных представлена на рисунке 2.

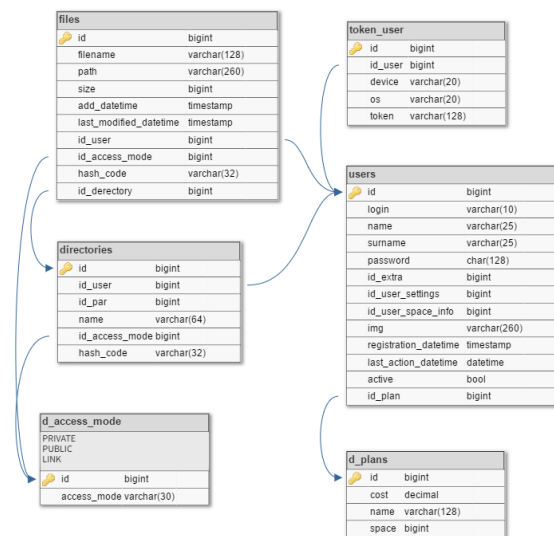


Рисунок 2 - Схема базы данных сервера.

На рисунке 3 представлен фрагмент диаграммы классов приложения, содержащий классы, обеспечивающие работу облачного хранилища и защиту данных.

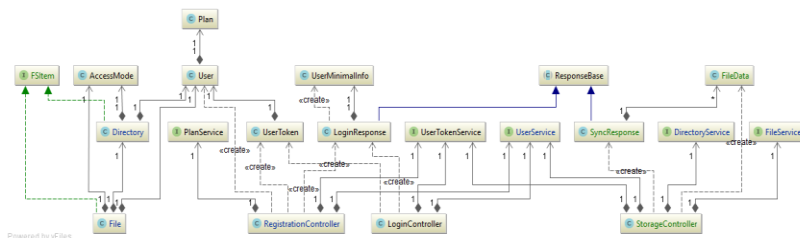


Рисунок 3 – Схема базы данных сервера.

На данный момент в приложении реализованы следующие функции:

- регистрация пользователя с созданием его родительской директории;
- вход в систему с выдачей токена для последующих запросов;
- создание и работа с директориями;
- синхронизация файлов с сервером;
- загрузка файла на сервер с шифрованием;

Выводы. В результате выполнения работы был проведен обзор существующих аналогов, реализующих шифрование пользовательских файлов. Были сформулированы требования к разрабатываемому продукту. В процессе работы было проведено проектирование клиент-серверного приложения и реализация его основной функциональности. В дальнейшем планируется расширить функциональность приложения для ускорения синхронизации: переименование, перенос, копирование, удаление файлов.

Список литературы

1. Онлайн-овые хранилища данных // ComputerBild : журнал. — 2010. — № 4. — С. 62—67.
2. Облачное хранилище данных // Википедия — свободная энциклопедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Dropbox>
3. Dropbox // Википедия — свободная энциклопедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Dropbox>
4. Mega // Википедия — свободная энциклопедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Mega>
5. AdvancedEncryptionStandard// Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Advanced_Encryption_Standard.
6. RSA // Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/RSA>
7. DSA // Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/DSA>

УДК 004.932

АНАЛИЗ СИСТЕМ РАСПОЗНАВАНИЯ НОМЕРНЫХ ЗНАКОВ АВТОМОБИЛЕЙ

Пикалёв Я.С., Дорохина Г.В.

Государственное учреждение

«Институт проблем искусственного интеллекта»

E-mail: pikaliov@gmail.com, sgv_iai@mail.ru

Аннотация

Пикалёв Я.С., Дорохина Г.В. *Анализ систем распознавания номерных знаков автомобилей. Рассмотрены современные средства распознавания номерных знаков автомобилей. Дана их краткая характеристика, продемонстрированы результаты их работы на наборе изображений.*

Ключевые слова: *распознавание, автомобили, сравнительный анализ*

Общая постановка проблемы

Повсеместно растёт потребность в надёжных системах автоматического распознавания номерных знаков, в том числе и в Донецкой Народной Республике.

Задачей распознавания изображений занимается достаточно много коллективов. Появились технологии, заявляющие о хороших результатах распознавания. Некоторые программные средства распознавания изображений, в том числе номерных знаков, доступны для использования как бесплатные или условно бесплатные. Цель работы – исследование доступных современных программных средств, которые могли бы быть использованы в системе распознавания номерных знаков автомобилей.

Распознавание номерных знаков включает в себя две взаимосвязанные задачи: детекция номерного знака и распознавание регистрационного номера автомобиля. Исследуем результаты их решения программами и программными библиотеками.

Исходные данные и методика сравнительного анализа

Для сравнительного анализа OCR были протестированы на изображениях, к которым предъявлялись следующие требования:

- номер автомобиля должен размещаться в кадре целиком;
- угол вертикального наклона номера - не более 40°;
- изображения должны быть четкими и не размытыми;
- номер не должен быть грязным или плохо читаемым;
- отсутствие встречной засветки изображения фарами автомобиля;

– другие помехи природного или техногенного происхождения, влияющие на качество изображения.

Для демонстрации полученных результатов отобраны три изображения (рис. 1, 2).



а)



б)

Рис. 1. Исходные данные: а) изображение №1; б) изображение №2.



Рис. 2. Исходные данные: изображение №3.

Изображение №1 – цветное (640x480, jpg); две машины с европейскими номерными знаками; содержит надпись в правом нижнем углу и множество ооконтуренных прямоугольных областей (рис. 1а). Изображение №2 – цветное

(1280x720, png) с изменённой цветовой гаммой, полученной в результате эффекта «выцветание»; машина с украинскими номерными знаками (рис. 16). Изображение №3 – градации серого (1280x1024, png); машина, имеющая номерной знак Украины, у которого в поле флага и герба расположена соответствующая символика ДНР; номерной знак имеет наклон к горизонту около 10°.

При анализе программ и библиотек распознавания продемонстрируем результаты их работы на исходных данных. В случае отрицательного результата распознавания, дополнительно рассмотрим работу системы распознавания на вручную вырезанной из исходного изображения области с номерным знаком.

Анализ результатов распознавания специализированными программами

Рассмотрим результаты распознавания номерных знаков на исходных изображениях с помощью программ, предназначенных для выполнения именно этой задачи: OpenALPR API и iANPR FREE.

OpenALPR API [1] – библиотека, совместимая с языками C/C++, C#, Java и Python. Анализирует изображения и видеопотоки для идентификации номерных знаков. Детектор OpenALPR [2] основан на Local Binary Pattern алгоритме [3]. Основой распознавателя [4] является Tesseract OCR 3.

OpenALPR содержит специализированные training data для различных номерных знаков, включая северно-американские номерные знаки (12x6 дюймов), европейские номерные знаки (52x11 см), австралийские, корейские и др. Также в репозитории программы есть инструментарий с подробными инструкциями для обучения детектора и распознавателя. Исходники программы находятся под Affero GPLv3 лицензией (является commercial-friendly лицензией).

Тестирование библиотеки проводилось при помощи скомпилированного исходного кода, а также реализации OpenALPR Cloud API [5]. При тестировании на наборе изображений библиотека показала высокую точность детектирования и распознавания.

К недостаткам системы можно отнести следующее.


OpenALPR Cloud API распознает лишь один номерной знак. требует Интернет-подключения, а также имеет ограничения для бесплатной версии в 2000 распознаваний в месяц. Результаты работы программы на исходном наборе изображений:

- изображение №1: SAM343 – 88,74%;
- изображение №2: П7777П – 80,68%;
- изображение №3: АН9257IE – 93,12.

Приложение OpenALPR API способно выделить более одного номерного знака, но на некоторых изображениях тестового набора не детектируются номерные знаки, к недостаткам также можно отнести его условную

доступность. Результаты работы этого приложения для изображений №1 и №2 приведены в таблице 1. Для изображения №3 программа не дала результата, при ручном выделении – также нет результата

Таблица 1 – Распознавание номерных знаков посредством библиотеки OpenALPR API.

Изображение №1	Изображение №2
<p>Matched License Plates:</p> <pre>-- Plate #1 -- SAM323 87,9% Fals SAM323 84,6% Fals SAM32 81,9% Fals SAM23 80,9% Fals SAM325 78,7% Fals SAM32 78,6% Fals SAMS23 78,4% Fals SAM32B 78,3% Fals -- Plate #2 -- SAM343 88,0% Fals SAM34 81,6% Fals SAM34B 78,9% Fals SAM343 78,3% Fals SAM345 78,0% Fals SAM43 78,0% Fals SAMB43 77,3% Fals SAM34G 76,8% Fals License Plate ROI: SAM 323 SAM 343</pre>	<p>Matched License Plates:</p> <pre>-- Plate #1 II7777II 95,2% Fa License Plate ROI: II 7777 II</pre> 

iANPR SDK [6] – это комплект средств разработки для распознавания автомобильных номеров на основе библиотеки компьютерного зрения OpenCV. Возможности библиотеки включают обработку изображений в формате OpenCV и выдачу результата распознавания: автомобильные номера, попавшие в кадр и их координаты. Количество номеров не ограничено. Версия 1.5 библиотеки поддерживает распознавание базовых и транзитных российских номеров, номеров прицепов, номеров такси и полиции, военных номеров, номеров мотоциклов и тракторов. А также ряд типов номеров для Беларуси, Казахстана, Туркменистана, Украины, Молдовы, Польши, Литвы, Латвии, Эстонии.

Недостатки: алгоритм распознавания в бесплатной версии библиотеки искусственно замедлен, а сама библиотека не предназначена для

использования в коммерческих целях. Результаты, полученные на исходных изображениях приведены в таблице 2.

Таблица 2 – Результаты распознавания iANPR SDK.

Изображение №1	Количество найденных номеров: 2 Номер: В443АО88 (Координаты: 114,242,116,41) Номер: Е883ТВ?? (Координаты: 458,278,79,43)
Изображение №2	нет результата, при ручном выделении: нет результата
Изображение №3	Количество найденных номеров: 1 Номер: А992ЕТ?? (Координаты: 558,729,270,155)

Выводы

Анализ специализированных программных средств сторонних разработчиков, которые могут быть использованы для распознавания номерных знаков автомобилей указывает на наличие проблем распознавания даже при хорошем качестве исходных изображений. Отсутствие возможности выполнить их дообучение или настройку для улучшения качества распознавания заставляет искать альтернативные средства для построения систем распознавания номерных знаков автомобилей.

Список литературы

1. <https://github.com/openalpr/openalpr>
2. <https://github.com/openalpr/train-detector>
3. http://www.scholarpedia.org/article/Local_Binary_Patterns 4
4. <https://github.com/openalpr/train-ocr>
5. <http://www.openalpr.com/cloud-api.html>
6. <http://intbusoft.com/rus/products/iANPR/>

УДК 004.42

РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ КОМПЬЮТЕРНОЙ СИСТЕМЫ ПСИХОФИЗИОЛОГИЧЕСКОЙ ДИАГНОСТИКИ

Ушаков А.И.¹, Дорохина Г.В.^{1,2}

¹Государственное образовательное учреждение высшего профессионального образования

«Донецкий национальный университет»
кафедра компьютерных технологий

²Государственное учреждение «Институт проблем искусственного интеллекта»

E-mail: sssusha56@gmail.com, sgv_iai@mail.ru

Аннотация

Ушаков А.И. Дорохина Г.В. Разработка клиент-серверного приложения компьютерной системы психофизиологической диагностики. Разработано клиент-серверное приложение компьютерной системы психофизиологической диагностики на основе компьютерного тестирования, в которой возможно добавление новых тестов.

Ключевые слова: диагностика, приложение

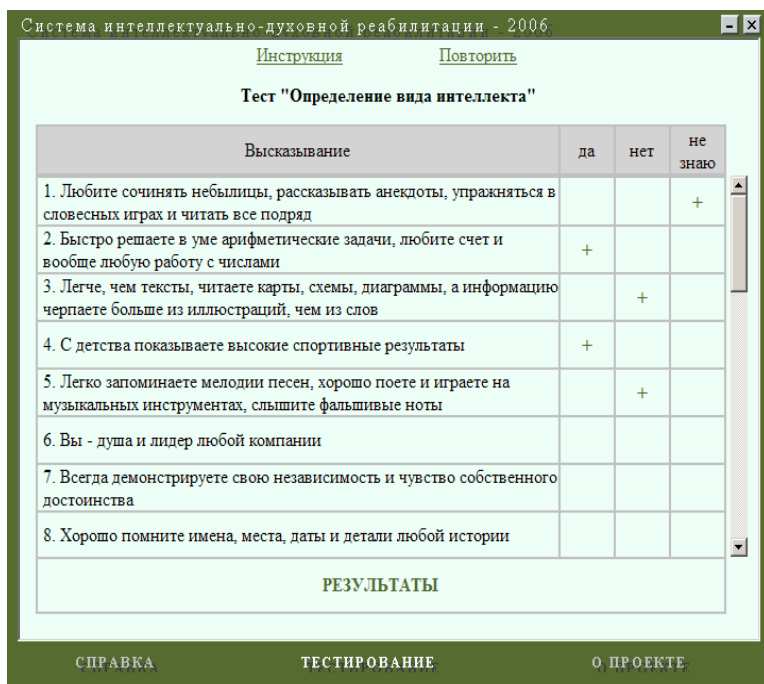
Общая постановка проблемы

В мировой науке и практике продолжает расти интерес к диагностике устойчивых индивидуальных психофизиологических особенностей и психических состояний человека [1]. Эта задача находит применение в области медицины, профориентации, вопросах техники безопасности и охраны труда. Некоторые виды профессий предъявляют жесткие требования к психофизиологическим качествам работника: эмоциональная уравновешенность, самообладание, выдержка, способность к сосредоточению, психическая выносливость, переключаемость внимания и т.п. Эти качества связаны со свойствами нервных процессов (физиологическими особенностям человека) и текущим функциональным состоянием организма. Функциональные состояния организма тесно взаимосвязаны с психическими состояниями человека [1]. Существуют диагностические методики для оценки физиологических особенностей и психических состояний человека.

К физиологическим особенностям человека, которые влияют на его способности, относят доминирующее полушарие, тип темперамента, силу нервной системы. Психические состояния – это целостные, временные и динамические характеристики психической деятельности человека, отражающие особенности протекания психических процессов и оказывающие

влияние на своеобразие функционирования интеллектуальной, волевой и эмоциональной сферы личности.

В качестве средства выполнения диагностики физиологических особенностей и психического состояния человека в данной работе используется тестирование. Программный комплекс [2, 3], послуживший прототипом разрабатываемой системы, представляет собой Windows-приложение, экранная форма которого приведена на рисунке 1. В работе также использовано информационное обеспечение упомянутой системы, а именно, собранные в ней стандартные тесты и опросники с алгоритмами оценки ответов пользователя.



Система интеллектуально-духовной реабилитации - 2006

[Инструкция](#) [Повторить](#)

Тест "Определение вида интеллекта"

Высказывание	да	нет	не знаю
1. Любите сочинять небылицы, рассказывать анекдоты, упражняться в словесных играх и читать все подряд			+
2. Быстро решаете в уме арифметические задачи, любите счет и вообще любую работу с числами	+		
3. Легче, чем тексты, читаете карты, схемы, диаграммы, а информацию черпаете больше из иллюстраций, чем из слов		+	
4. С детства показываете высокие спортивные результаты	+		
5. Легко запоминаете мелодии песен, хорошо поете и играете на музыкальных инструментах, слышите фальшивые ноты		+	
6. Вы - душа и лидер любой компании			
7. Всегда демонстрируете свою независимость и чувство собственного достоинства			
8. Хорошо помните имена, места, даты и детали любой истории			

РЕЗУЛЬТАТЫ

СПРАВКА ТЕСТИРОВАНИЕ О ПРОЕКТЕ

Рис. 1. Экранная форма приложения-прототипа

Формирование требований к системе

Для охвата большей аудитории пользователей разрабатываемую систему целесообразно реализовать в виде клиент-серверного приложения. Также важно использовать в базу данных для сохранения результатов диагностики пользователей.

В психофизиологической диагностике применяются разные виды тестов (например, текстовые, графические), отличаются способы отображения предлагаемых пользователю вариантов ответов и алгоритмы их оценок.

В ряде случаев, результат одного теста отображается на нескольких шкалах. Например, тест по методике САН [2] позволяет оценить психологическое состояние человека по шкалам: «Самочувствие», «Активность», «Настроение», а опросник на «выгорание» Маслача и Джексона [2] – по шкалам: «Эмоциональное истощение», «Деперсонализация», «Редукция личных достижений».

Необходимо разработать способ хранения тестов, позволяющий хранить в базе данных и тесты, и данные необходимые для подсчёта результатов. Это позволит в компьютерную систему психофизиологической диагностики добавлять новые тесты при наличии в базе данных теста того же типа.

Разрабатываемое приложение должно работать с четырьмя классами пользователей, в работе с которыми система должна выполнять следующие функции:

- незарегистрированный пользователь (проходить тесты, зарегистрироваться);
- зарегистрированный пользователь (проходить тесты, смотреть свои результаты за текущий сеанс);
- специалист по диагностике (проходить тесты, просматривать результаты пользователей за определенный период; добавлять и редактировать тесты);
- администратор (проходить тесты; просматривать свои результаты за текущий сеанс; изменять права доступа зарегистрированных пользователей; удалять пользователей).

Разработка системы

Для хранения в базе данных и тестов, и данных необходимые для их отображения пользователю и подсчёта результатов потребовалось классифицировать собранные в работе [2, 3] тесты. Для реализации в системе выбраны два класса, позволяющие описать восемь тестов и опросников:

- Методика САН (самочувствие, активность, настроение);
- «Измерение степени выраженности сниженного настроения – субдепрессии»;
- «Шкала ситуативной тревожности»;
- Методика «Определение состояния фрустрации»;
- Методика «Диагностика уровня социальной фрустрированности»;
- «Модифицированная шкала личностной соревновательной тревожности Р. Мартенса»;
- Методика «Накопление эмоционально-энергетических зарядов, направленных на самого себя»;

– Опросник на «выгорание» Маслача и Джексона.

Данные приложения будем хранить в реляционной базе данных, логическая модель которой представлена на рисунке 2.

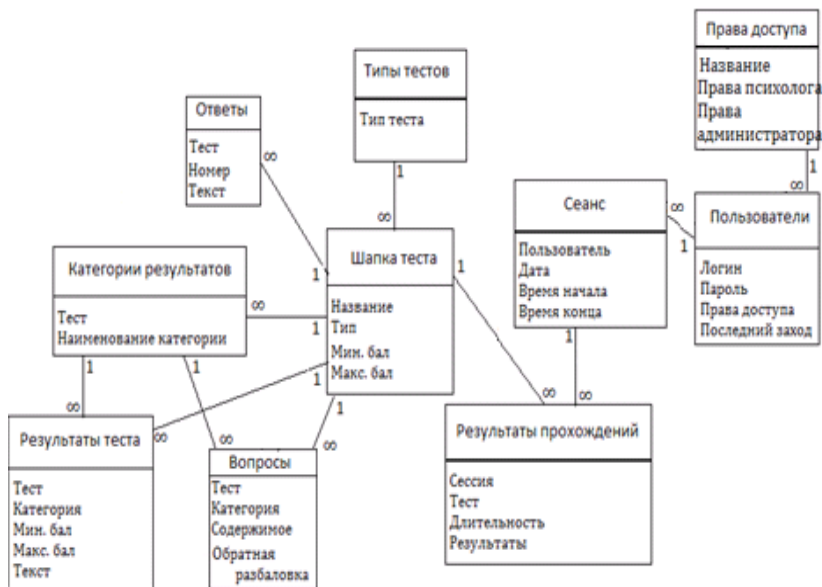


Рисунок 2 – Логическая модель базы данных

Она учитывает следующие сущности и их атрибуты: шапка теста (заголовок, тип теста, информация, необходимая для оценивания ответов); ответы (номер теста, номер и текст ответа); категория результатов; вопрос (номер теста; номер категории, к которой относится вопрос; текст вопроса; порядок начисления баллов за ответы: прямой или обратный); результат теста; пользователь; права доступа пользователей; сеанс (идентификатор пользователя, дата/время начала и окончания сеанса, пройденные тесты и их результаты).

Результирующее клиент-серверное приложения создано на языке программирования PHP, которое использует web-сервер Apache и сервер управления базами данных MySQL.

Выводы

Разработано клиент-серверное приложение компьютерной системы психофизиологической диагностики на основе компьютерного тестирования. Приложение позволяет пользователю определять свои психофизические параметры путем компьютерного тестирования, просматривать результаты

тестирования за определенный период. Специалист по диагностике, дополнительно может редактировать и добавлять тесты, а администратором – изменять права доступа и удалять пользователей.

Дальнейшее развитие системы связано с улучшением внешнего вида приложения, усовершенствованием системы безопасности и добавлением и обработкой новых видов тестов.

Список литературы

1. Мантрова И.Н. Методическое руководство по психофизиологической и психологической диагностике./ И.Н.Мантрова – ООО "Нейрософт" (Россия, Иваново). – 2007 г. – 216 с. URL: <https://kognoreg.files.wordpress.com/2015/03/psyhotest2.pdf> (дата обращения: 27.04.2017 г.).

2. Сальников И.С. Современные методы и методики изучения и диагностирования интеллектуально-психофизиологических состояний человека и способы их компьютерной аудио-визуальной терапии / И.С. Сальников, Р.И. Сальников // «Искусственный интеллект». – 2006. – №4. – С. 548-554.

3. Шевченко А.И. Разработка и построение комплексной компьютерной системы психофизиологического диагностирования, интеллектуально-духовной реабилитации и безмедикаментозной терапии / А.И. Шевченко, И.С. Сальников, Р.И. Сальников, А.В. Дьяченко // «Искусственный интеллект». – 2007. – №3. – С. 6-16.

УДК 004.9

ПРОГРАММНАЯ СИСТЕМА ДЛЯ БИСЕРОПЛЕТЕНИЯ

И.А. Коломойцева, Н.О. Сторожук
Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
bolatiger@gmail.com, storozhuk2909@mail.ru

Аннотация

Коломойцева И.А., Сторожук Н.О. Программная система для бисероплетения. Целью данной работы является описание проектирования и частично реализации программной системы «CrochetBeadMaster». Для этого сформулированы требования, определены функциональные модули, а также реализованы базовые алгоритмы и основные экраны интерфейса.

Ключевые слова: бисер, схема, озвучивание, программная система, файл, модуль, навигация, требование, алгоритм, дерево.

Введение. Постоянно растущая популярность бисероплетения не могла не коснуться информационных технологий, в отдельную категорию следует выделить программы, предоставляющие возможность самостоятельно составлять схемы для вязания бисером и воспроизводить их звуковой аналог для набора бисера рукодельницей.

Создание готового изделия из бисера без использования программных средств имеет ряд недостатков:

- используется чужая схема, что исключает уникальность изделия;
- своя схема может быть создана только от руки на бумаге либо с использованием ПО другого назначения, что является неудобным, трудоемким и исключает возможность создания качественной схемы;
- при самостоятельном наборе последовательности бисера повышается риск некорректного набора, увеличивается нагрузка на человека;
- диктовка другого человека повышает скорость набора, однако требует дополнительный человеческий ресурс;
- ручной просчет материалов требует от пользователя знаний определенных статистических данных;
- при расчете стоимости бисера вручную повышается риск ошибочных вычислений, субъективности расчетов;
- при расчете вручную может возникнуть необходимость повторного перерасчета в случае утери старого либо незначительных изменениях (длины, цвета и т.п.).

В [1] были рассмотрены достоинства и недостатки существующих

программ, оптимизирующих работу со схемами вязания бисером, сформулированы базовые требования, определены функциональные характеристики и модули программного продукта CrochetBeadMaster.

Целями написания статьи являются:

- формирование итоговых требований к данным и функционалу программы, интерфейсу и аппаратуре, на которой будет использоваться разрабатываемая программная система;
- обзор основных данных, обрабатываемых в программе;
- описание архитектуры разрабатываемой системы;
- обзор используемых алгоритмов и методов;
- анализ проделанной работы а также планирование дальнейшей деятельности, направленной на доработку программного продукта.

Обзор требований к программной системе. В результате анализа существующих систем бисероплетения и требований пользователя к таким системам определены функции, которые должны быть реализованы:

- поддержка форматов входных файлов – dbb, jbb;
- поддержка два языка интерфейса – русский, английский;
- наличие калькулятора для подсчета количества исходных материалов для заданной схемы;
- наличие инструкции;
- эргономичность интерфейса;
- обработка ошибок и исключительных ситуаций программы;
- наличие 3 схем – образцов;
- возможность полной замены цвета в схеме;
- возможность прикрепить фото готового жгута и просмотреть изображение из программы;
- прокрутка образца жгута;
- выбор скорости чтения;
- возможность установки паузы;
- отображение количества бисерин в диаметре;
- отображение количества необходимых раппортов;
- отображение времени набора;
- подсчет количества бисера каждого цвета для заданной длины;
- подсчет необходимой длины нити для заданной длины жгута.

Интерфейс приложения должен соблюдать следующие базовые требования:

- максимально комфортное взаимодействие программы и пользователя;

- минимальное количество времени для освоения программного приложения;
- вывод информации о необходимых материалах;
- использование общепринятых обозначений для универсальных кнопок;
- наличие раздела «Справка» с детализацией предназначения элементов управления;
- ограничение на ввод некорректных данных;
- обработка исключительных ситуаций;
- запрос подтверждения для удаления данных.

Каждый экран приложения должен иметь конкретное предназначение и предоставлять доступ к необходимым данным и функциям. Таким образом, выделены 6 основных экранов. Рассмотрим каждый из них.

Экран «Мои раппорты» должен отображать краткую информацию в табличном виде обо всех схемах, которые добавлены в программу: название схемы, её диаметр, число цветов и миниатюру с фотографией, если она прикреплена. По нажатию на строку с определенной схемой, пользователь должен увидеть экран с детализацией информации о выбранной схеме, а также иметь возможность добавлять схему в программу из файловой системы.

Экран «О схеме» должен содержать следующую информацию о схеме: название;

- диаметр;
- число цветов;
- базовый цвет;
- комментарий (изменяемое поле);
- ссылка на фото, которое можно просмотреть, удалить, заменить другим.

Экран «Набор» предполагает собой наличие:

- таблицы в виде «число бисерин – цвет – готовность»;
- кнопок «Начать/Пауза», «Стоп» для озвучивания раппорта;
- значения скорости и кнопки для её изменения;
- возможности изменять цвета перед чтением;
- показателей длительности набора всего и оставшуюся длительность;
- показателя готовности одного раппорта в процентах.

Экран «Калькулятор» должен предполагать ввод информации о длине жгута, а также размере и стоимости бисера, объёме упаковки. На основании этой информации и данных из схемы, должны быть рассчитаны следующие значения:

- число требуемых бисерин каждого цвета;
- объём требуемых бисерин каждого цвета в граммах;

- количество упаковок бисера каждого цвета, округленных вверх;
- общая стоимость бисера на вес и в упаковках;
- длина нити, требуемой для выполнения жгута.

Экран «Справка» должен содержать краткое описание функционала программы, а также элементов управления.

Экран «О программе» должен вмещать информацию о текущей версии приложения, его авторе, используемых библиотеках и лицензиях на них.

В программе должна присутствовать удобная панель для навигации между окнами, кнопка «Выход», вспомогательные окна для просмотра фотографии и добавления схемы из файловой системы в программу, иконка приложения, описывающая тематику программы либо индикатор автора программы.

Программа должна корректно функционировать на смартфонах, которые удовлетворяют следующим требованиям:

- ОС Android с API старше 4.4.4 (19);
- частота не менее 1.4 Гц;
- оперативная память не менее 1,5 Гб;
- исправный сенсорный экран;
- исправный динамик или исправный разъём для наушников и наушники.

Архитектура программной системы. На основании полученных требований, а также анализа входов и выходов проектируемой системы, были определены прецеденты, соответствующие основным сервисам системы.

Диаграмма прецедентов (диаграмма вариантов использования) в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне [2]. Большое значение диаграмма прецедентов имеет для специфицирования, визуализации и документирования поведения системы. Используя ее, разработчику легче понимать систему, подсистему или классы, а также взглянуть извне на преимущества использования элементов для того или иного контекста. Подобная UML-диаграмма представляет особую важность для проведения тестирования исполняемых систем при прямом проектировании а также для лучшего понимания их внутреннего устройства [3].

На рисунке 1 представлена диаграмма вариантов использования для описываемой системы. Актёрами является пользователь, файловая система устройства, а также устройство вывода звука. Следует отметить, что некоторые прецеденты слабо детализированы по причине избыточности информации на диаграмме. Так прецедент «Работа с фото» предполагает «Просмотр фото», «Добавление фото», «Удаление фото» и «Замена фото».

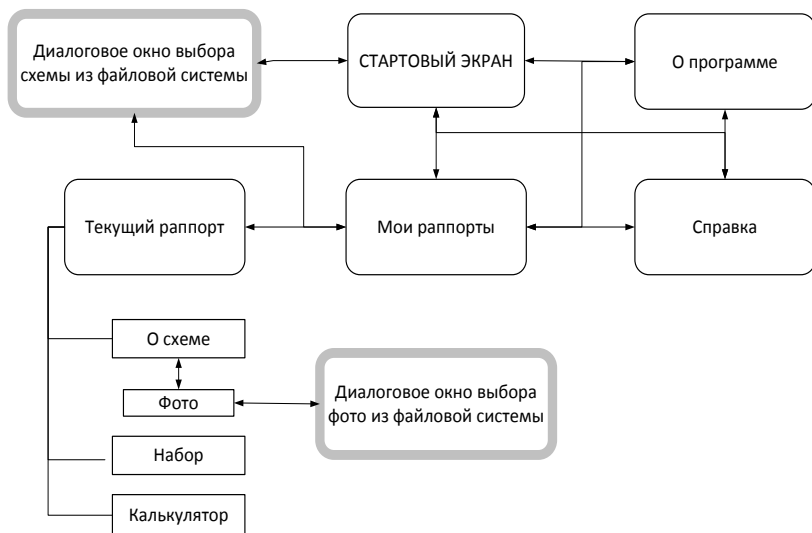


Рисунок 2 – Карта навигации между экранами

Обзор файла с расширением .jbb и алгоритма для работы с ним.

Основным используемым типом данных в программе CrochetBeadMaster является схема изделия из бисера. Так как такие схемы могут быть использованы или созданы в сторонних программах, необходимо изучить и использовать стандартное представление информации о планируемом изделии. Рассмотрим структуру стандартного файла схемы (рис. 3).

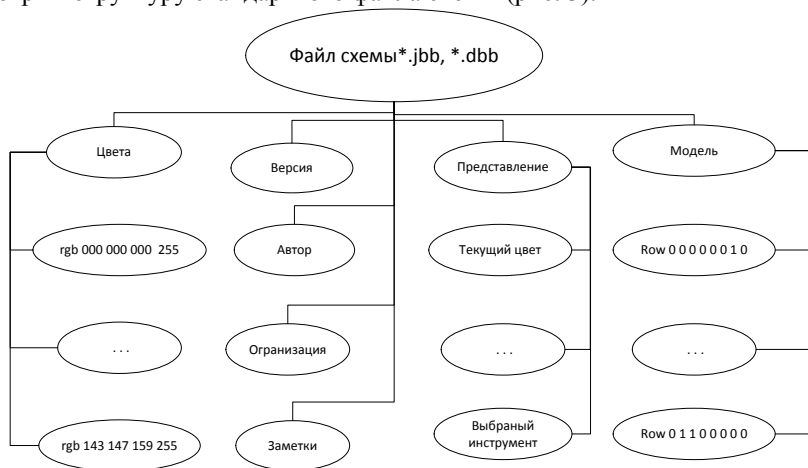


Рисунок 3 – Структура файла

Заголовок файла содержит информацию общего характера, такую как:

- версия схемы (version) – по умолчанию устанавливается «1», может изменяться по желанию пользователя после редактирования;

- автор схемы (author) – логин присваивается автоматически при сохранении схемы на сайте CrochetBeadPaint [4], в противном случае поле остаётся пустым;

- организация (organization) – заполняется тогда, когда доступ к готовой схеме на сайте CrochetBeadPaint ограничен, схема не будет открыта (чаще всего - в платных схемах);

- заметки (notes) - раздел не обязателен для заполнения и содержит любые комментарии пользователя.

Далее следует список цветов (colours) в rgb-формате для возможности их дальнейшего конвертирования и использования в разных программах для работы с бисером.

Следующий раздел под названием «представление (view)» содержит характеристики, описывающие последнее состояние программы перед сохранением. Данные показатели обязательно используются, если схема открывается в одной и той же программе. Если пользователь использует схему в нескольких приложениях, данные параметры могут быть учтены не полностью, так как интерфейс и возможности программ различны. Рассмотрим диапазон значений и характеристику каждого показателя:

- использование панели эскиза (draft-visible) – показатель фиксирует наличие/отсутствие панели для рисования во время последнего сеанса, принимает состояния true и false;

- использование панели скорректированного эскиза (corrected-visible) – аналогичный предыдущему, но скорректированный эскиз отображает бусины уже со смещением для вязки;

- использование симулирования готового изделия (simulation-visible) – показатель может принимать значения true или false в зависимости от того, было ли использовано представление готового изделия;

- использование отчета (report-visible) – индикатор фиксирует наличие/отсутствие отчета по схеме во время последнего сеанса, принимает состояния true и false;

- выбранный инструмент (selected-tool) – показатель, фиксирующий последний выбранный во время рисования схемы инструмент. Может применять значения, такие как: «fill» - заливка для выделенного фрагмента текущим цветом, «pencil» - карандаш, инструмент, предназначенный для окраски одной клетки поля в текущий цвет и т.д.;

- масштабирование (zoom) – показатель принимает значения от 0 до 7, по умолчанию установлено значение 2. Используется для фиксации изменения масштаба схемы;

– прокрутка (scroll) – показатель фиксирует строку, которая остается самой нижней на панели эскиза в момент завершения работы;

Алгоритм выделения информации из дерева модели

```
Считать файл из памяти в массив байт
Считать из массива байт в дерево схемы
текущий корень = "version"
Получить версию
текущий корень = "author"
Получить автора
текущий корень = "organisation"
Получить организацию
текущий корень = "notes"
Получить комментарий

Очистить массив цветов
текущий корень = "color"
пока (лист = rgb)
{
    текущий корень = лист
    Получить в red значение первого листа
    Получить в blue значение второго листа
    Получить в green значение третьего листа
    Если ( четвертый лист)
    {
        Получить в alpha значение четвертого листа
    }
    Добавить цвет в массив
}
текущий корень = "model"
высота = число листов
текущий корень = первый лист "row"
диаметр = число листов корня
массив данных схемы = массив [высота*диаметр]
текущий корень = "model"
пока (лист = row)
```

```

{
    текущий корень = лист
    пока (лист)
    {
        ...      в массив данных Добавить лист
    }
}
Конец

```

Рисунок 4 – Алгоритм выделения информации из дерева модели

– сдвиг (shift) – значение показателя находится в диапазоне от 0 до числа бисерин в диаметре жгута и показывает, на сколько бисерин сдвинут макет готового изделия по отношению к эскизу;

– рисование с помощью цветов (draw-colors) – идентификатор принимает значения true или false, и показывает, были ли использованы стандартные цвета. По умолчанию установлен в true;

– рисование с помощью символов (draw-symbols) – идентификатор принимает значения true или false, и показывает, были ли использованы стандартные символы. По умолчанию установлен в false, так как предоставляемой цветовой гаммы обычно достаточно для удобной работы;

– символы (symbols) – здесь перечислены стандартные символы, которые доступны для использования при необходимости. Как правило после индикатора следует следующий список символов: ".abcdefghijklmnopqrstuvwxyz+~\|*"

Заключительный раздел «модель (model)» содержит описание схемы. Данные представлены в виде строк, начинающихся со слова «row». Каждая строка содержит ряд цифр, разделяющихся пробелами, в количестве бисерин в диаметре готового изделия. В такой строке каждое число обозначает номер цвета из раздела «цвета», которым будет окрашена бисерина в этом ряду. Таким образом, данный раздел представляет «развертку» готового жгута с закодированными в номера цвета.

После выбора пользователем файла, система считывает байты файла с дальнейшим преобразованием в символы. Далее символы группируются в «фразы», т.е. смысловые фрагменты содержимого файла: скобки, числа, слова или другие символы. Следующим этапом является анализ полученного списка элементов и формирование дерева, грубую визуализацию которого можно видеть на рисунке 3. Рисунок 4 представляет собой псевдокод функции, реализующий получение значений основных параметров схемы из дерева в переменные. Следует отметить, что раздел «Представление (view)» пропускается, так как приложение не предполагает использование инструментов для рисования, основные параметры которых представлены в

этом разделе.

Выводы. В ходе работы рассмотрены базовые требования к системе и устройству, на котором система будет запущена, рассмотрена базовая структура данных и кратко описан принцип работы с ней а также некоторые аспекты архитектуры системы в виде диаграммы прецедентов и навигации между экранными формами. Последующими направлениями работы являются:

- проработка качественного дизайна интерфейса и его реализация;
- усовершенствование разработанных алгоритмов;
- обработка исключений и тестирование программы.

Список литературы

1. Коломойцева И.А., Сторожук Н.О. Проектирование программной системы для бисероплетения «CrochetBeadMaster» // Программная инженерия: методы и технологии разработки информационно – вычислительных систем (ПИИВС-2016): сборник научных трудов I научно – практической конференции (студенческая секция). – Донецк, ГОУ ВПО «Донецкий национальный технический университет», 2016 – 39-43 с.

2. Построение диаграммы прецедентов [электронный ресурс] // Life – Prog: [сайт]. [2016]. URL: http://life-prog.ru/1_16788_postroenie-diagrammi-pretседentov.html

3. UML – диаграмма прецедентов. [электронный ресурс] // FB.ru: [сайт]. [2016]. URL: <http://fb.ru/article/91646/uml-diagramma-pretседentov>

4. Инструкция. [электронный ресурс] // CrochetBeadPaint: [сайт]. [2016]. URL: <http://crochetbeadpaint.info>

УДК 004.9

ПРОГРАММНАЯ СИСТЕМА ДЛЯ ПЕРЕВОДА ТЕКСТА С ПОМОЩЬЮ КАМЕРЫ МОБИЛЬНОГО УСТРОЙСТВА

И.А. Коломойцева, Д.А. Власюк

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
bolatiger@gmail.com, dimugan@ukr.net

Аннотация

Коломойцева И.А., Власюк Д.А. Программная система для перевода текста с помощью камеры мобильного устройства. В работе сформированы итоговые требования к разрабатываемой системе, приведено описание основных алгоритмов, функциональных возможностей, интерфейса приложения и дальнейшее планирование при выполнении дипломного проекта.

Ключевые слова: распознавание символов, перевод, оптимизация, словари, точность, интерфейс, программный продукт, прецедент.

Введение. Данная статья посвящена описанию собственного приложения, использующего камеру мобильного устройства для перевода текста с одного языка на другой. В [1] выполнен сравнительный анализ аналогов и определены их достоинства и недостатки, представленные в виде таблицы. Необходимо отметить, что качество работы приложения зависит от многих воздействующих факторов, например, от камеры мобильного устройства, освещения, в котором выполняется снимок и т. д. Полный список данных факторов представлен в [1].

В результате анализа аналогичных программ, представленного в [1], наиболее успешным признано приложение GoogleTranslate, которое имеет высокий процент распознавания символов, быструю скорость перевода, и поддерживает большое количество языков. Также неплохие результаты показали CameraTranslate и CamTraslator, которые немного уступают предыдущему приложению, однако являются достойными представителями данной области.

К недостаткам существующих систем, относится использование большого объема памяти, необходимого для скачивания обновлений, а также пакетов словарей. Отсутствие офлайн перевода ограничивает пользователя в возможностях работы с приложением без наличия Интернета, что является минусом некоторых существующих приложений. Также страдает скорость работы, стабильность распознавания и возможность получения перевода как больших объемов текста, так и маленьких. Еще одним недостатком некоторых готовых приложений было отсутствие русифицированной версии, однако

данное замечание касается только определенного региона проживания. Такое приложение как CamTranslatorPro оптимизировано только под iOS и не имеет версии для Android, несмотря на значимость этой платформы и количество пользователей [1]. Получив все необходимые сведения, об аналогах, сформировалось виденье разрабатываемого программного продукта, который должен соответствовать необходимым требованиям и не уступать уже существующим приложениям.

Целями написания статьи являются:

- составление окончательных требований к собственной программной системе;
- описание программной системы, используемых алгоритмов и функциональных возможностей;
- описание интерфейса приложения;
- обнаружение дополнительных замечаний к дальнейшей разработки и дальнейшее планирование.

Требования к разрабатываемой программной системе. После изучения и анализа существующих приложений, использующих камеру мобильных устройств для ввода исходной информации, а также после прочтения отзывов пользователей к данным продуктам, были сформированы требования к функционалу будущего приложения.

Приложение должно обладать следующими характеристиками:

- оптимизированная скорость анализа и перевода текста;
- поддержка нескольких языков перевода;
- достаточный уровень устойчивости и стабильности распознавания текста;
- простой и удобный интерфейс пользователя;
- возможность выбора изображения для перевода;
- работа с камерой мобильного устройства;
- вывод результата в удобной для просмотра форме;
- взаимодействие между окнами интерфейса;
- работу системы не должен влиять размер экрана устройства;
- возможность правки распознанного текста с целью более точного перевода;
- качество перевода не менее 80% анализируемого текста.

Система должна включать в себя модуль распознавания символов из изображения, а также модуль перевода полученного текста на необходимый язык и, собственно, интерфейс приложения. Данная программа должна быть разработана с помощью среды AndroidStudioи средств языка Java.

Описание программной системы. Работа с данными является основой разрабатываемого приложения. В качестве исходных данных выступает изображение с текстом, который необходимо распознать. Также для обеспечения возможности успешного распознавания, используются специальные словари, по которым и происходит получение связанных слов и предложений из набора символов.

Для начала работы пользователю необходимо выбрать высококачественное изображение из галереи, или же сделать моментальный снимок интересующего текста. После чего, система проверит, если ли на мобильном устройстве необходимые словари для выполнения распознавания. В случае их отсутствия, произойдет создание каталога и копирование необходимых файлов на телефон или планшет. Далее, программа проанализирует выбранный файл и попытается выдать распознанный текст. Диаграмма состояний для работы модуля распознавания символов представлена на рисунке 1.

Перевод – деятельность по интерпретации смысла текста на одном языке и созданию нового эквивалентного ему текста на другом языке [2]. Перевод будет осуществляться с использованием трех языков: русский, английский и немецкий. Для получения переведенного текста необходимо реализовать базу данных, в которой и будут содержаться необходимые слова. При переводе будет производиться проверка наличия слова в базе, и в случае отсутствия нужного слова, оно будет найдено с помощью Интернета. После чего будет производиться сохранение полученной информации.

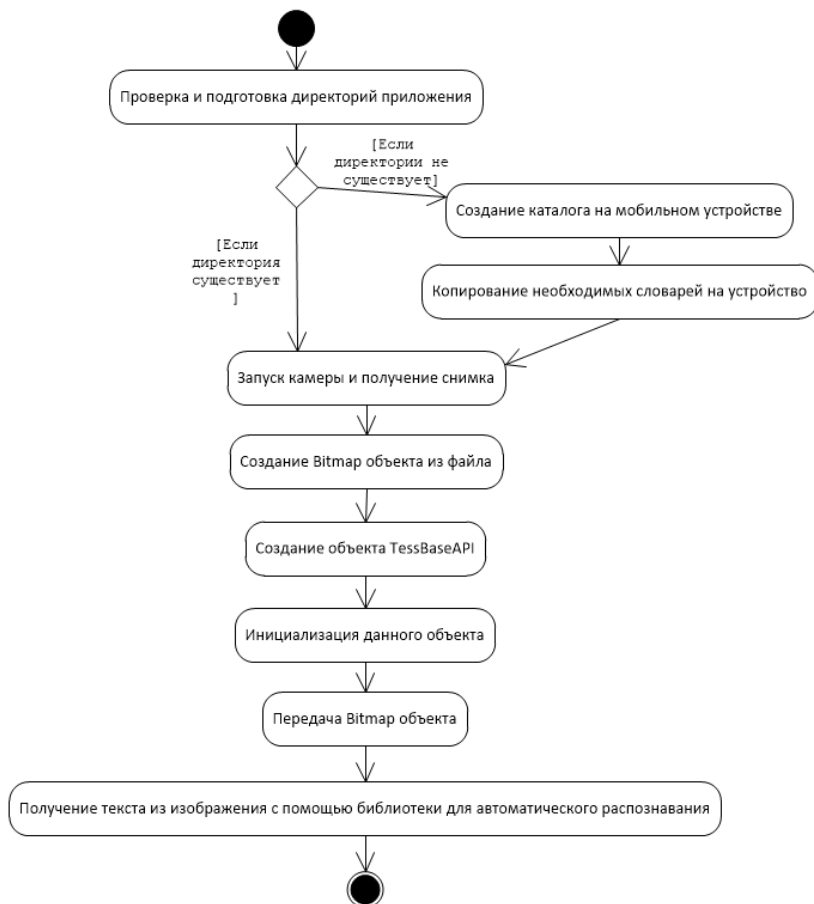


Рисунок 1 – Диаграмма состояний для модуля распознавания текста

На рисунке 2 представлена диаграмма прецедентов программной системы, основными актерами которой являются пользователь, камера и файловая система.

Диаграмма прецедентов (диаграмма вариантов использования) – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели, позволяющей описать систему на концептуальном уровне [3].

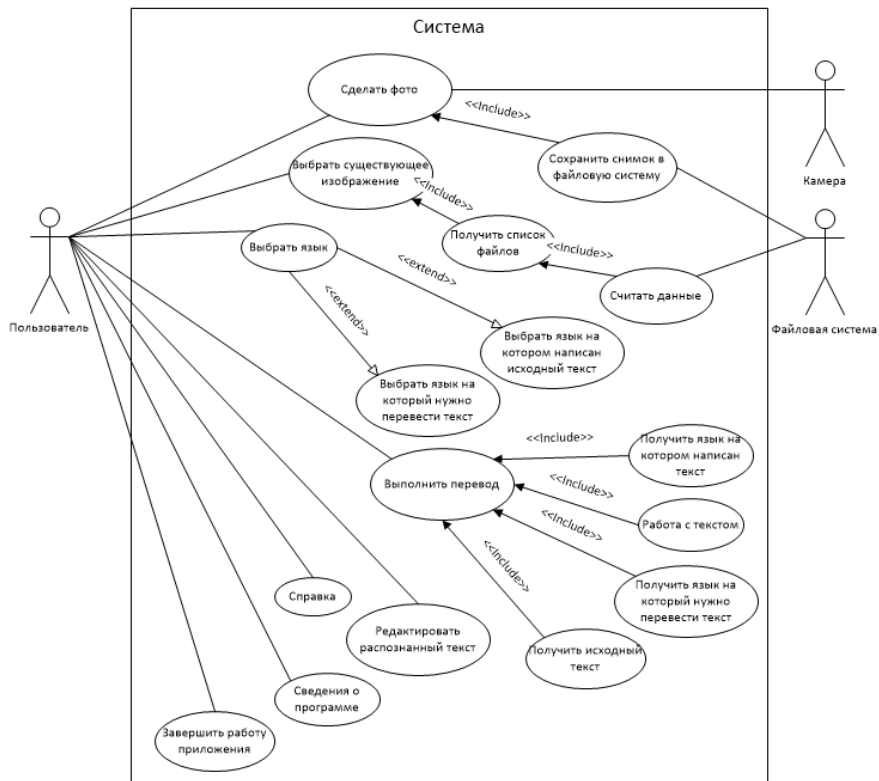


Рисунок 2 – Диаграмма прецедентов

Интерфейс и экранные формы приложения. Интерфейс программы – это то, с чем непосредственно имеет дело пользователь, он взаимодействует с приложением именно благодаря интерфейсу. Поэтому пользовательский интерфейс приложения должен быть удобным и простым для понимания[4]. Данное приложению содержит два экрана, на рисунке 3 показаны экранные формы программной системы.

На главном экране расположены два выпадающих списка, с помощью которых пользователь может выбрать язык, с которого и на который нужно выполнить перевод. Также здесь расположены кнопки для выбора существующих файлов из галереи, и для создания моментального снимка, а также кнопка для перехода к второму экрану на котором и будет отображаться переведенный и исходный тексты.

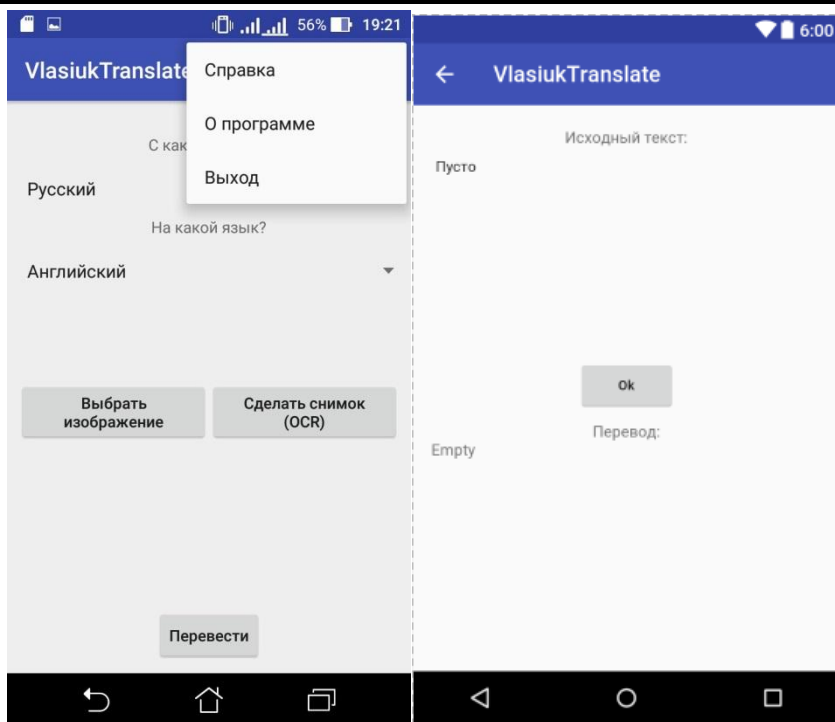


Рисунок 3 – Экранные формы приложения

На втором экране присутствуют поля для распознанного и для переведенного текста. Поле для исходного текста позволяет редактировать полученные с изображения данные. Это необходимо для того, чтобы пользователь смог исправить возможные ошибки и получить более качественный перевод. Помимо этого на экране находится кнопка для повторного перевода отредактированного текста и кнопка возврата на главную форму.

Все возможности приложения должны быть интуитивно понятны для обычного пользователя. Однако, справа в верхнем углу находится специальное меню. На вкладке «О программе» описаны основные сведения о приложении, версия, автор и прочая дополнительная информация. Пункт «Справка» предназначен для справки по работе приложения, его функционалу и возможностях для пользователя. Также здесь находится кнопка для завершения приложения.

Выводы. В ходе выполнения данной работы сформированы окончательные требования к разработке программной системы для перевода текста с использованием камеры мобильного устройства. Также представлено описание данной системы, показан алгоритм работы отдельного модуля и интерфейс программы.

В ходе дальнейшей разработки необходимо обеспечить быстрый и качественный перевод полученного текста, после чего приложением можно свободно пользоваться в различных областях, например, в путешествиях или же при прочтении книг на иностранном языке.

Список литературы

1. Коломойцева И.А., Власюк Д.А «Приложение для перевода текста с помощью камеры мобильного устройства «VlasiukTranslate»» //Программная инженерия: методы и технологии разработки информационно – вычислительных систем (ПИИВС-2016): сборник научных трудов I научно – практической конференции (студенческая секция). – Донецк, ГОУ ВПО «Донецкий национальный технический университет», 2016 – 34-38 с.
2. Перевод. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Перевод>
3. Диаграмма прецедентов. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Диаграмма_прецедентов
4. Требования к дизайну пользовательского интерфейса. [Электронный ресурс]. – Режим доступа: http://www.okd.mdk.ksue.edu.ua/index.php?option=com_content&view=article&id=174&Itemid=90

УДК 004.021, 004.5, 004.8

АНАЛИЗ СЛОВАРЕЙ АССОЦИАЦИЙ В МЕТА-ПОИСКОВОЙ СИСТЕМЕ ПОИСКА ЦИТАТ

О.А. Серёженко, И.А. Коломойцева

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии
seryozhenko-94@mail.ru

Аннотация

Серёженко О.А., Коломойцева И.А. Анализ словарей ассоциаций в мета-поисковой системе поиска цитат. В данной статье описаны проблема поиска текста в сети Интернет, в частности цитат, инструментарий онлайн-словарей ассоциаций на основе примеров проектов «Сеть словесных ассоциаций» и «Карта слов».

Ключевые слова: *поисковая система, мета-поисковая система, словарь ассоциаций, ассоциация, виртуальная библиотека, цитата.*

Введение. На сегодняшний день актуальной проблемой для многих людей каждый день является проблема поиска информации в Интернете. На сегодняшний день существует крайне мало решений, обеспечивающих поиск в Интернете цитат из произведений.

Поиск цитат — это поиск текста по заданному фрагменту. Пользователь, задавший такой запрос, вероятнее всего, хочет найти происхождение цитаты — то есть либо увидеть произведение, из которого она взята, либо хотя бы узнать автора и название этого произведения [1].

Задача поиска цитат относится к поиску извлечения знаний.

Постановка проблемы. В настоящий момент не существуют ярко выраженных центров сосредоточения знаний. Традиционные источники получения информации: библиотеки, базы данных, архивы, становятся огромными, что воспринимаются не как отдельные информационные узлы, а как совокупность множества распределенных источников информации.

Однако объединение распределенных источников информации несет в себе серьезные проблемы, связанные с быстрым и релевантным поиском и классификацией необходимых информационных ресурсов. Кроме того, пользователи не всегда знают электронные адреса серверов с необходимой для них информацией.

Таким образом в образовательной и научной деятельности, при поиске библиографической информации, невозможно эффективное использование

новых информационных сред без применения специализированных поисковых механизмов.

Каждая поисковая система имеет свое множество документов, которые доступны для поиска. Ни одна из подобных систем не сможет охватить всех ресурсов сети Интернет, поэтому в любой момент может возникнуть ситуация, когда информационные потребности пользователя не смогут быть удовлетворены. Как правило, в этом случае пользователь переходит на другую поисковую систему и пытается искать то, что ему нужно, там.

Каждая поисковая машина использует собственный алгоритм поиска и индексирования Интернет-ресурсов. Поисковых систем много, базы данных (или индексы) поисковых систем далеко не являются одинаковыми. Поэтому возникла идея для удобства поиска объединить результаты выдачи по тому или иному запросу сразу из нескольких систем в одном интерфейсе.

Решение проблемы. Для решения данной проблемы и расширения возможности поиска, были созданы системы, названные мета-поисковыми. Они не имеют собственных поисковых баз данных, не содержат никаких индексов и при поиске используют ресурсы множества поисковых систем. За счет этого полнота поиска в таких системах максимальна и вероятность нахождения нужной информации очень высока.

Архитектура мета-поисковой системы. Мета-поисковая система строится на принципах клиент-агент-серверной архитектуры с ультратонким необслуживаемым клиентом, где клиентом является стандартный Web-браузер, агентом – мета-поисковая система, а сервером - Web-сервера так называемой “Виртуальной библиотеки”, к чьим поисковым механизмам обращается агент [2]. Виртуальная библиотека объединяет в себе электронные каталоги, систему интеллектуального поиска и клиентские места.

Принцип мета-поисковой системы. Особенностью мета-поисковой системы является то, что обращение идет не непосредственно к электронным каталогам, а механизмам их обслуживающим, это позволяет не изменять структуру и принципы работы Web-серверов “Виртуальной библиотеки”.

Большинство мета-поисковых систем отображают результаты поиска нескольких поисковиков в одном объединенном списке, из которого удалены повторяющиеся записи. Каждая подобная система имеет свой собственный метод определения релевантности, поэтому трудно скомбинировать их всех в общую категорию того, как они работают, хотя они придерживаются одних и тех же базовых принципов.

У мета-поисковых систем нет собственной базы данных веб-страниц, они просто действуют как посредник между пользователем и другими поисковыми системами. Как только пользователь вводит свой поисковый запрос в мета-

поисковый механизм, они затем передают его основным поисковым системам и предоставляют вам результаты из всех баз данных, опрошенных быстро. Поскольку основные поисковые системы часто приводят к очень разным результатам, системы мета-поиска особенно полезны, когда у пользователя мало времени, и он хочет получить как можно больше доступной информации [3].

Словарь ассоциаций. Важной частью мета-поисковой системы поиска цитат являются словари ассоциаций, синонимов, реже, антонимов. Особенно стоит обратить внимание на словарь ассоциаций, так как человек вспоминает какую-либо фразу или цитату по ассоциациям. Ассоциация – возникающая в опыте человека закономерная связь между двумя содержаниями сознания (ощущениями, представлениями, мыслями, чувствами и т. п.), которая выражается в том, что появление в сознании одного из содержаний влечет за собой и появление другого [4].

Примеры ассоциаций приведены на рисунке 1.

Там на неведомых тропинках/дорожках...
Храните деньги в банке/сберегательной кассе...
Либо я веду ее в ЗАГС, либо она меня ведет к прокурору/тюрьму.

Рисунок 1 – Примеры ассоциаций

Можно использовать существующие готовые словари ассоциаций, которые описаны ниже. Отправляя запросы программными средствами, пользователь получает набор слов-ассоциаций по введённому слову.

Проект «Сеть словесных ассоциаций». Проект «Сеть словесных ассоциаций» позволяет просматривать ассоциации, возникающие у человека, с заданным словом. В основе процесса формирования списка ассоциаций лежит программный модуль, анализирующий классические и современные произведения русской и зарубежной литературы с использованием принципов системного подхода. По утверждению авторов проекта, сформированный список ассоциаций может рассматриваться как усредненный результат при проведении лингвистического ассоциативного эксперимента

Проект «Сеть словесных ассоциаций» может оказаться полезным для литературоведов, начинающих писателей, журналистов, лингвистов, социологов, маркетологов, менеджеров по рекламе, а также любознательных людей, желающих узнать новое о русском языке и об окружающем мире [5].

Достоинствами вышеуказанного проекта являются:

- яркий, приятный дизайн;

- сопровождение заданного слова статьями из толковых словарей Д. Н. Ушакова, В. И. Даля и т. д.;
 - формирование списков ассоциаций по частям речи;
 - возможное сопровождение заданного слова картинками;
 - сопровождение поиска афоризмами, цитатами известных людей.
- Недостатками проекта являются:
- отсутствие примеров использования заданного слова;
 - отсутствие списка синонимов, антонимов заданного слова.

Ознакомиться с проектом «Сеть словесных ассоциаций» можно по адресу <https://wordassociations.net/ru/>. На рисунках 2-3 показаны соответственно главная страница проекта и пример работы с ним.



Рисунок 2 – Главная страница проекта «Сеть словесных ассоциаций»

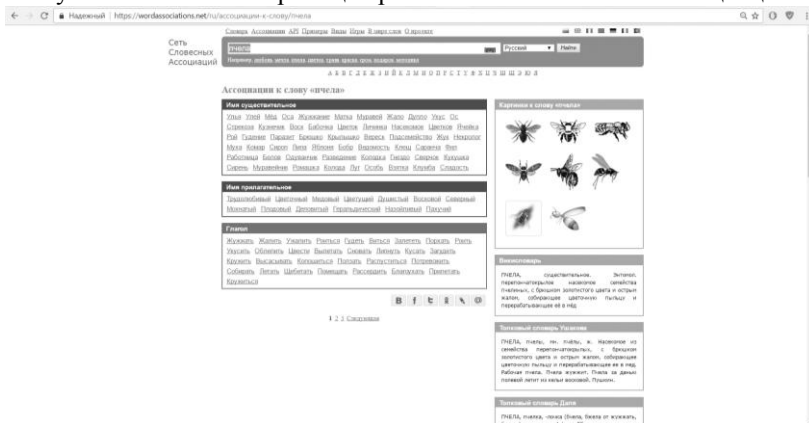


Рисунок 3 – Результат работы поиска ассоциаций к заданному слову

Проект «Карта слов». Проект «Карта слов» – большая онлайн-карта русского языка, целью которой является сделать современный, открытый и доступный всем пользователям Рунета онлайн-словарь, где можно будет найти справочную информацию (ударения, склонение имён и спряжение глаголов), а также ассоциации и синонимы, примеры употребления слов в контексте и т.д.

Достоинствами данного проекта являются:

- формирование списка ассоциаций по мере использования слов;
- приведение примеров использования заданного слова в словосочетаниях и предложениях;
- наличие списка синонимов;
- наличие дополнительных возможностей (например, правильное написание, склонение, разбор по составу заданного слова).

Недостатками проекта является:

- отсутствие разбиения на списки по частям речи;
- отсутствие списка антонимов.

Ознакомиться с проектом «Карта слов» можно по адресу <https://kartaslov.ru/>. На рисунках 4-5 показаны соответственно главная страница проекта и пример работы с ним.

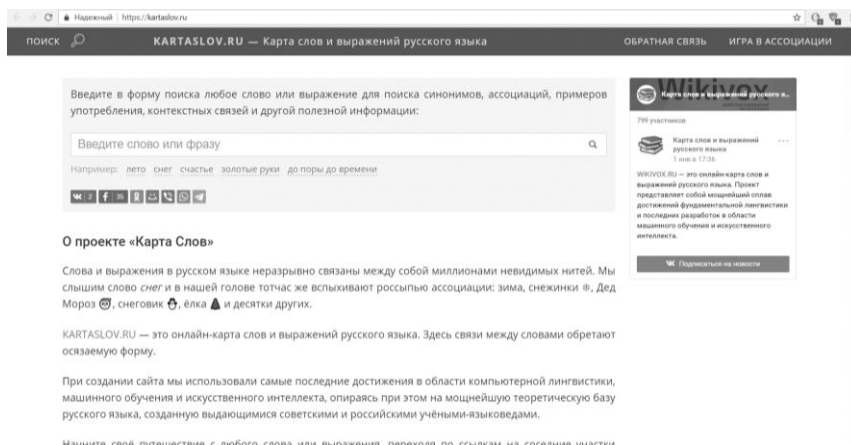


Рисунок 4 – Главная страница проекта «Карта слов»

Выводы. В ходе написания статьи были проанализированы:

- проблемы поиска цитат в сети Интернет;
- инструментарии существующих онлайн-словарей ассоциаций, приведены их характеристики.

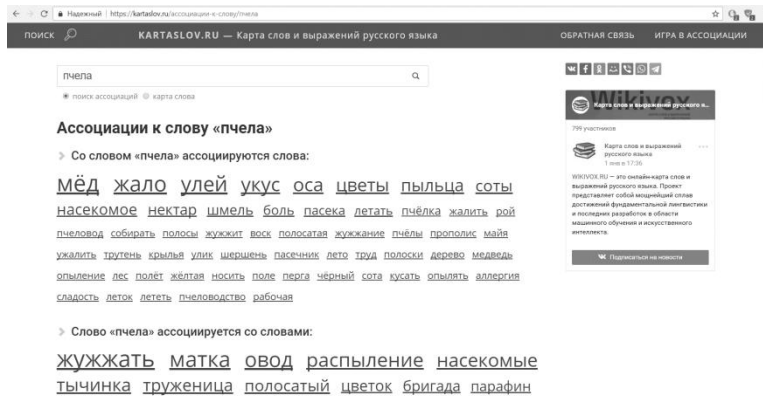


Рисунок 5 – Результат работы поиска ассоциаций к заданному слову

Словари ассоциаций являются важной частью мета-поисковой системы поиска цитат так как человек вспоминает какую-либо фразу или цитату по возникающим ассоциациям.

Рассмотренные существующие онлайн-словари ассоциаций, которые можно использовать в разрабатываемой мета-поисковой системе поиска цитат с помощью сгенерированных программных запросов.

Список литературы

1. Серёженко О.А., Коломойцева И.А. Применение мета-поиска к решению задач поиска цитат // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2016): сборник научных трудов I научной-практической конференции (студенческая секция). 16-17 ноября 2016 г. – Донецк, ГОУ ВПО «Донецкий национальный технический университет», 2016. – с. 194-200.
2. Буханов А.Н., Саркисова И.О. Увеличение эффективности механизмов поиска информации с помощью постобработки данных. / Международная конференция "Информационные средства и технологии" в 3-х т.т. Т.2 – М.: Станкин, 1998. – с.121-126.
3. Автоматизация поиска неиндексируемых ресурсов в распределенных компьютерных сетях [электронный ресурс]. – Режим доступа: http://magazine.stankin.ru/arch/n_10/14/index.html
4. Ассоциация [электронный ресурс]. – Режим доступа: <https://ru.wiktionary.org/wiki/ассоциация>.
5. Сеть Словесных Ассоциаций. О проекте [электронный ресурс]. – Режим доступа: <https://wordassociations.net/ru/about>.

УДК 004.056.5

ОСОБЕННОСТИ МОДЕЛИ АУТЕНТИФИКАЦИИ НА ОСНОВЕ ЦВЗ ДЛЯ РАЗРАБОТКИ СИСТЕМ ГРАФИЧЕСКОГО ПАРОЛЯ

И.А. Ефремченко, Н.Е. Губенко

Донецкий национальный технический университет, г. Донецк

кафедра компьютерного моделирования и дизайна

efr-inessa@yandex.ru

Аннотация

Ефремченко И.А., Губенко Н.Е. Особенности модели аутентификации на основе ЦВЗ для разработки систем графического пароля. В данной статье проведен анализ методов стеганографии, которые позволяют повысить стойкость систем к методам взлома систем с парольной аутентификацией пользователей. Анализируется метод аутентификации пользователя на основе ЦВЗ.

***Ключевые слова:** методы аутентификации, графический пароль, цифровой водяной знак, авторизация, информационная система, пользователь.*

Постановка проблемы. С появлением мониторов и различных устройств с сенсорными экранами ведутся разработки систем графических паролей, которые создаются, чтобы избавить пользователя от сложных паролей и упростить авторизацию, но при этом обезопасить доступ к ресурсам. Как известно, самым уязвимым звеном безопасности является непосредственно сам пользователь, который не всегда запоминает несколько сложных паролей. До сих пор в большинстве случаев используется известное слово или имя, в лучшем случае снабжённое несколькими дополнительными цифрами. Словарные атаки, а также система автоматического подбора всех возможных комбинаций знаков позволяет вскрыть такой пароль за короткий промежуток времени. Таким образом, проблема безопасности защищаемых ресурсов при прохождении идентификации/аутентификации пользователей средствами систем графического пароля остается актуальной. Решение этой проблемы можно обеспечить с помощью применения стенографических методов, которые позволяют повысить стойкость систем к большинству известных атак и методов взлома на парольные системы.

Цель статьи – провести анализ методов использования цифровых водяных знаков для совершенствования систем графического пароля.

Цифровые водяные знаки.

В настоящий момент компьютерная стеганография включает в себя несколько областей исследования:

1. Встраивание информации с целью ее скрытой передачи;

2. Встраивание цифровых водяных знаков (ЦВЗ), в частности для защиты авторских прав на электронную продукцию, такую как видео, аудио и графические файлы в электронном виде;

3. Встраивание заголовков;

4. Встраивание идентификационных номеров.

Наибольший интерес вызывают ЦВЗ. Они применяются как для маркирования электронных файлов, так и для встраивания и передачи различной информации по каналам связи[1].

ЦВЗ также могут быть трех видов:

- хрупкие;
- полухрупкие;
- робастные.

Хрупкие ЦВЗ разрушаются при незначительной модификации заполненного контейнера. Они применяются для аутентификации сигналов. Отличие от средств электронной цифровой подписи заключается в том, что хрупкие ЦВЗ все же допускают некоторую модификацию контента. Это важно для защиты мультимедийной информации, так как законный пользователь может, например, пожелать сжать изображение. Другое отличие заключается в том, что хрупкие ЦВЗ должны не только отразить факт модификации контейнера, но также вид и местоположение этого изменения.

Полухрупкие ЦВЗ устойчивы по отношению к одним воздействиям и неустойчивы по отношению к другим. Полухрупкие ЦВЗ специально проектируются так, чтобы быть неустойчивыми по отношению к определенному роду операциям. Например, они могут позволять выполнять сжатие изображения, но запрещать вырезку из него или вставку в него фрагмента.

Под робастностью понимается устойчивость ЦВЗ к различного рода воздействиям на стегоконтейнер. Робастные ЦВЗ могут быть 3-х типов. Это ЦВЗ, которые могут быть обнаружены всеми желающими, хотя бы одной стороной, либо это могут быть ЦВЗ которые трудно модифицировать или извлечь контента (контейнера) [2].

Таким образом, можно выделить актуальную задачу по применению ЦВЗ, которая состоит в аутентификации пользователя при условии устойчивости к навязыванию ложных сообщений нарушителем и воздействию случайных и преднамеренных ошибок в каналах связи. В ходе поиска решения данной задачи необходимо использование методов контроля подлинности на основе ЦВЗ и применении как одного, так и нескольких видов цифровых водяных знаков.

Модель аутентификации пользователя на основе ЦВЗ системы представлена на рис.1:

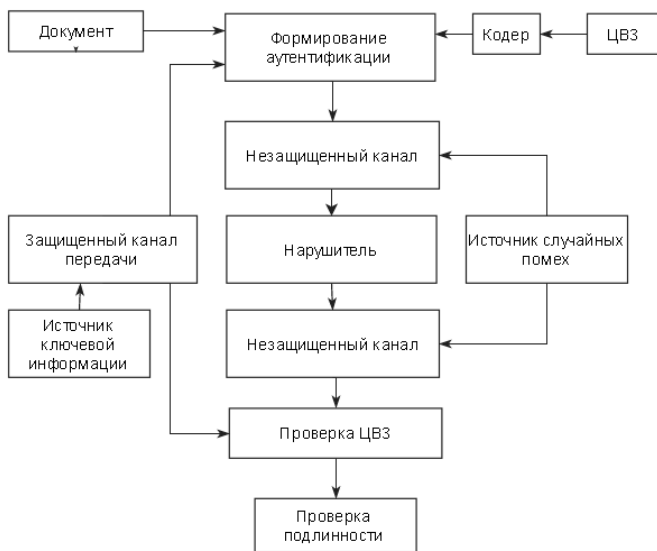


Рисунок 1 - Модель аутентификации пользователя на основе ЦВЗ

Алгоритм аутентификации пользователя на основе ЦВЗ:

1. Отправитель создает документ, накладывает цифровой водяной знак, индивидуальный для каждого отправителя, который преобразовывается в кодере к удобному виду для встраивания в заверяемое сообщение.

2. В формирователе заверенных сообщений конструкция водяного знака встраивается в документ, используя конфиденциальный ключ.

3. В канале связи на заверенное сообщение воздействуют нарушитель, а также случайные и преднамеренные помехи. В результате этого воздействия на приеме в устройство проверки водяных знаков поступает модифицированное сообщение.

4. По алгоритму обнаружения цифрового водяного знака формируется его оценка. Подлинность документа определяется в соответствии с этой оценкой.

При этом возможны решения следующего вида:

- подлинность сообщения подтверждена;
- подлинность сообщения не подтверждена;
- фрагмент скорее всего подлинный;
- фрагмент скорее всего навязан или искажен помехами передачи.

При формировании оценки водяных знаков могут возникнуть ошибки их обнаружения получателем сообщения[3].

В данной модели под ЦВЗ понимается встраивание определенной информации о пользователе и его пароле к определенному ресурсу. Информация кодируется и встраивается в последовательность графических файлов. Пользователь выбирает определенную последовательность файлов и подтверждает свой выбор. Система из этой последовательности извлекает ЦВЗ, в которых содержатся имя пользователя и пароль, если последовательность верна, то пользователю открывается ресурс, в обратном случае доступ к защищенному ресурсу будет закрыт, до тех пор, пока последовательность будет правильной.

Однако данная модель обладает рядом недостатков:

- не предусмотрены ограничения на количественный выбор последовательности, то есть если, к примеру, последовательность была выбрана не правильно несколько раз, то доступ пользователя блокируется автоматически;
- использование только одного метода встраивания ЦВЗ, что приводит к недостаточной устойчивости ко взлому.

Выводы. Благодаря использованию ЦВЗ в системах графических паролей атаки на данные системы становятся более сложными в реализации и значительно отличаются от атак на часто используемые символьные системы.

Рассмотренная модель является достаточно распространенной, однако для повышения ее эффективности целесообразно провести следующие доработки:

- ввести количественные ограничения на ввод изображений для подбора парных изображений;
- для реализации модели целесообразно использовать различные типы ЦВЗ (хрупкие, полухрупкие, робастные) для каждого файла из последовательности, причем методы для встраивания могут выбираться в произвольной последовательности.

Список литературы

1. Введение в цифровую стеганографию [электронный ресурс] // Портал магистров ДонНТУ: [сайт]. [2006]. URL: <http://masters.donntu.org/2006/fvti/khotov/library/stego.htm>
2. Цифровые водяные знаки [электронный ресурс] // Образовательная социальная сеть KazEdu.kz: [сайт]. [2009-2017]. URL: <https://www.kazedu.kz/referat/133581/2>
3. Шокарев А.В. Графические пароли с использованием методов стеганографии // Инновационные технологии и экономика в машиностроении: Труды VII Всеросс. Научно-практ. конф. С междунар. участием. – Томск: Изд-во Томского политехнического университета, 2009. – С. 293–299.

УДК 004.92

ИНФОРМАЦИОННАЯ СИСТЕМА «GRABTHE TRADE»

М.Г. Титаренко, И.А. Коломойцева

Донецкий национальный технический университет, г. Донецк

кафедра компьютерных наук и технологий

timigen2010@yandex.ua

Аннотация

Титаренко М.Г., Коломойцева И.А. Информационная система «GrabTheTrade». В данной статье описана постановка задачи, а также реализация некоторых элементов информационной системы «GRABTHE TRADE». Сформулированы основные требования, структура БД и используемый шаблон проектирования.

Ключевые слова: парсер, шаблон, анализ достоверности, Django, MTV, MVC, БД.

Введение. Ежедневно в современном мире человек получает большое количество информации о торгово-экономических связях между государствами, торговых «войнах», экономической зависимости одной страны от другой, эмбарго и прочем.

Для осуществления анализа данной информации ему необходимо посетить несколько сайтов с данными по тому или другому государству. В русскоязычном интернете такая информация отсутствует. Однако на англоязычных ресурсах, таких как TradeMap [2] или официальный сайт ВТО, необходимые данные также находятся довольно тяжело, так как они расположены вперемешку с другими экономическими данными и экономическими отчетами.

В связи с этим пользователь должен приложить немало усилий для анализа и систематизации необходимой ему информации. Однако, учитывая психологию человеческого восприятия, он (человек) отставит попытки разобраться в этом аспекте.

Следовательно, необходимо предоставить пользователю возможность быстрого и наглядного получения относительно достоверной информации по торгово-экономическим связям между государствами, их взаимозависимости по этим связям, а также возможность прогнозирования их будущих торговых отношений на основании полученных ранее данных.

Кроме интерфейса для конечных пользователей необходимо создать администраторскую панель. Необходимо предусмотреть также масштабируемость системы с возможностью будущей автоматизации получения информации.

Опишем основные задачи данного проекта.

Получение данных с заданных интернет источников. В качестве маски для получения данной информации будут использованы шаблоны. В качестве альтернативы необходимо задание используемого на сайте API. Решение данной проблемы позволит упростить задание администратором данных с необходимых для чтения сайтов.

Анализ достоверности полученных данных. При совмещении данных с нескольких сайтов, они могут не совпадать (согласно разным оценкам). Поэтому после добавления новых данных необходимо проверять, насколько значительны расхождения в них. Причем пересмотру подлежит каждый интернет ресурс в отдельности по отношению к совокупности других ресурсов. Решение этой проблемы позволит отсеять заведомо абсурдную информацию.

Анализ полученной информации, выявление наиболее объективной. После получения необходимой информации и оценки ее достоверности необходимо проанализировать совокупность данных и сделать наиболее объективную оценку отношений. Кроме того, отдельно должен высчитываться коэффициент зависимости государств друг от друга. Решение данной задачи позволит использовать данные результата для отображения пользователю.

Автоматическое периодическое обновление данных. Ежеквартально в автоматическом режиме, без участия администратора необходимо проверять наличие новой информации на указанных ресурсах и обновлять данные. В результате решения данной проблемы снижается нагрузка на администратора системы.

Вычисление прогноза отношений по полученным данным. Кроме вывода полученной и проанализированной информации необходимо также на основании истории взаимоотношений выстроить предположение о дальнейшем развитии данных отношений. В случае, если поступят данные, по которым был сделан прогноз, необходимо оценить их сходство и на основании этого улучшить прогнозирование в перспективе. Решение данной задачи позволит пользователю примерно оценивать, в каком направлении движутся отношения двух стран на этом этапе.

При решении задачи парсинга сайтов будет использована библиотека lxml, анализ которой проводился в предыдущей статье [1].

Основные требования к продукту.

К разрабатываемому продукту были выдвинуты следующие требования:

- парсинг указанного интернет ресурса по шаблону или API;
- анализ достоверности полученных результатов;
- автоматическое периодическое обновление данных;
- анализ сохраненной информации и вычисление коэффициента

зависимости;

- наглядный вывод полученной и обработанной информации.

Также были выделены требования к интерфейсу приложения.

Интерфейс приложения должен быть разделен на 2 части – клиентскую и администраторскую.

Клиентская часть приложения должна содержать единственную страницу на которой будут располагаться основные элементы:

- карта мира с возможностью выбора страны;
- элементы управления для выбора страны, партнера, направления торговли и прочего;
- один график для демонстрации текущего положения дел, а также прогноза дальнейшего развития;
- таблица с конкретными данными по стране за определенные года.

Время отзывчивости интерфейса не должно превышать 1 секунды.

Прототип элемента с картой представлен на рисунке 1.



Рисунок 1 – Элемент интерфейса «карта»

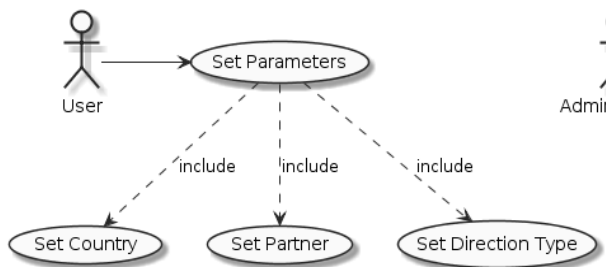


Рисунок 2 – Прецеденты целевого пользователя

По внешним требованиям была составлена диаграмма прецедентов, представленная на рисунках 1-2.

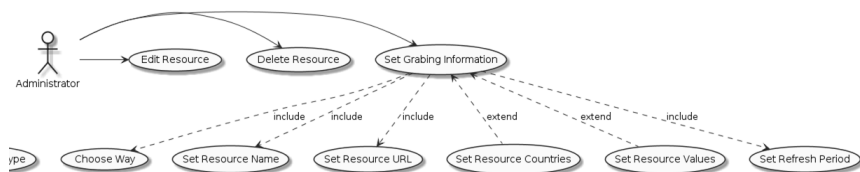


Рисунок 3 – Прецеденты администратора

Для удовлетворения изложенных выше требований был выбран шаблон проектирования MVC и фреймворк Django.

Шаблон проектирования MTV(MVC).

Для разработки системы применяется фреймворк Django, который использует шаблон проектирования MTV(MVC). Django следует шаблону MVC в такой мере, что его можно было бы назвать MVC-фреймворком.

Опишем примерное соответствие между буквами M, V, C и концепциями Django.

M – часть, касающаяся доступа к данным; соответствует уровню работы с базой данных в Django.

V – часть, касающаяся решения о том, что и как отображать, соответствует представлениям и шаблонам.

C – часть, которая передает управление некоторому представлению в зависимости от того, что ввел пользователь, реализована самим фреймворком с помощью конфигурации URL, которая говорит, какую функцию представления вызывать для данного URL.

Поскольку буква «C» реализована самим фреймворком, а самое интересное происходит в моделях, шаблонах и представлениях, то поэтому Django стали называть MTV-фреймворком, где приняты следующие обозначения.

M означает «Model» (модель), то есть уровень доступа к данным. На этом уровне сосредоточена вся информация о данных: как получить к ним доступ, как осуществлять контроль, каково их поведение, каковы отношения между данными.

T означает «Template» (шаблон), уровень отображения. Здесь принимаются решения, относящиеся к представлению данных: как следует отображать данные на веб-странице или в ином документе.

V означает «View» (представление), уровень бизнес-логики. На этом уровне расположена логика доступа к модели и выбора подходящего шаблона (или шаблонов). Можно сказать, что это мост между моделями и шаблонами.

Если брать другие MVC-фреймворки в веб-разработке, например Ruby on Rails, то можно считать, что представления Django – это «контроллеры», а шаблоны Django – «представления». [5]

Структура БД.

При проектировании была разработана структура базы данных для данного проекта, которая представлена на рисунке 4.

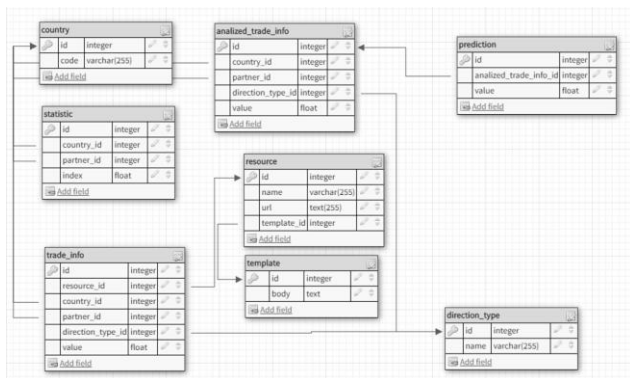


Рисунок 4 – Структура БД

По назначению база данных разделяется на две категории.

В первую категорию попадают таблицы для хранения информации как есть. Это так называемая «накопительная» часть базы. Эта часть базы используется непосредственно при сохранении распарсенной информации интернет ресурса. Используется преимущественно администраторской частью системы. К данной категории относятся таблицы country, resource, direction_type, template и trade_info.

Ко второй категории относятся таблицы, в которых хранятся данные после обработки. Эти данные уже могут использоваться для вывода пользователю. Используется преимущественно клиентской частью системы. К данной категории относятся таблицы analyzed_trade_info, statistic и prediction.

Кратко рассмотрим назначение таблиц (таблица 1).

Выводы. На данный момент спроектирована и реализована база данных системы. Реализован алгоритм парсинга интернет ресурсов, а также частично реализован интерфейс клиентской части приложения. В дальнейшем планируется реализация алгоритма проверки согласованности данных, реализация прогнозирования данных, а также доработка интерфейса клиентской и администраторской части системы. Планируется создание автоматизации обновления данных.

Таблица 1 – Назначение таблиц БД

Название таблицы	Назначение
country	Содержит буквенный код страны. Полное название страны содержится в языковых файлах.
resource	Содержит данные о ресурсах для осуществления парсинга.
direction_type	Хранит набор типов данных, по типу экспорт, импорт, реэкспорт и тд.
template	Содержит данные о шаблоне.
trade_info	Содержит непосредственно экономические данные о взаимоотношении стран, представленных на определенном ресурсе.
analized_trade_info	Содержит проанализированные экономические данные о взаимоотношении стран.
statistic	Хранит индекс взаимозависимости двух стран.
prediction	Содержит предположительные экономические данные о взаимоотношении стран, которые были сгенерированы в результате прогнозирования.

Список литературы

1. Титаренко, М.Г. «Информационная система «Торгово-политическая карта мира»» / Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС - 2016) – Донецк. «Цифровая типография», 2016.
2. TradeMap [Электронный ресурс]. – Режим доступа: http://www.trademap.org/Bilateral_TS.aspx - (24.10.2016)
3. Кузнецов, И.Н. Информация: сбор, защита, анализ. Учебник по информационно-аналитической работе – М. Яуза, 2001.
4. TheIxml.etree [Электронный ресурс]. - Режим доступа: <http://lxml.de/> - (12.05.2017).
5. WebDevelop.Info [Электронный ресурс]. - Режим доступа: <http://webdevelop.info> - (12.05.2017).
6. Django [Электронный ресурс]. - Режим доступа: <https://www.djangoproject.com/> - (12.05.2017)
7. Python [Электронный ресурс]. - Режим доступа: <https://www.python.org/> - (12.05.2017)

УДК 004.056.53

СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ВЫБОР АЛГОРИТМОВ ХЕШИРОВАНИЯ ДЛЯ ОРГАНИЗАЦИИ ПАРОЛЬНОЙ ЗАЩИТЫ ИГРОВЫХ ПРИЛОЖЕНИЙ

Н.Е.Губенко, А.В.Сирант

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
an147job@gmail.com

Аннотация

Губенко Н.Е., Сирант А.В. Сравнительный анализ и выбор алгоритмов хеширования для организации парольной защиты игровых приложений. Рассматриваются проблемы защищенности игрового веб-приложения. Был проведен анализ существующих решений для парольной защиты с учетом существующих видов атак.

***Ключевые слова:** криптография, хеширование, кибер-безопасность, игровое приложение, веб-приложение.*

Постановка проблемы. Современная игровая индустрия занимает значительный сегмент рынка программных продуктов и является потребителем и стимулятором графических систем. Создателям цифровых видео-игр постоянно приходится сталкиваться с читерством, которое базируется на взломе исходного кода игры (например, путем реверсного инжиниринга) либо других ее составляющих. При этом четко прослеживается закономерность: чем популярнее игровое приложение, тем больше шансы быть взломанными [1]. Продажа читов к играм является целым подпольным бизнесом, ведь каждый хочет получить игровое преимущество, которое легально можно получить только за реальные деньги (имеются в виду игры, распространяемые по модели «free to play»).

Другая проблема состоит в защите данных каждого конкретного пользователя, в распоряжении которого может находиться собственное «цифровое имущество». Этой проблеме уделяется в данной статье основное внимание.

Анализ существующих видов атак. Проведем анализ различных атак на пользовательскую базу данных. Согласно принципу Керкгоффса [2] при оценке надёжности криптосистемы необходимо предполагать, что злоумышленник знает об используемой системе шифрования всё, кроме ключей. В данном случае ключами являются пароли пользователей, которые

хранятся в виде хешей, а сама СУБД, соответственно, была взломана и база данных целиком попала к злоумышленнику.

Самый простой метод взлома это попытка подобрать пароль. Таких методов взлома два: это атака по словарю и brute-force атака (простой лексикографический перебор всех возможным вариантов). При данных атаках берется текущий предполагаемый пароль, хешируется той же функцией, что и пароли в БД, а затем полученный хеш сравнивается с хешем в БД. Если хеши совпали, значит предполагаемый пароль верен. Процесс подбора пароля описанными методами показан ниже на рис. 1. От данных атак сложно защититься, но есть способы сделать их менее эффективными.

Dictionary Attack	Brute Force Attack
Trying apple : failed	Trying aaaa : failed
Trying blueberry : failed	Trying aaab : failed
Trying justinbeiber : failed	Trying aaac : failed
...	...
Trying letmein : failed	Trying acdb : failed
Trying s3cr3t : success!	Trying acdc : success!

Рисунок.1. Взлом пароля с помощью атаки по словарю и brute-force атаки.

Следующий метод взлома - это поиск по таблице. Это чрезвычайно эффективный метод для взлома множества хешей одного типа. Суть метода заключается в построении таблицы с заранее вычисленными значениями хешей и соответствующими значениями пароля. При этом просто производится поиск хеша подвергаемого взлому, и затем из таблицы берется исходное значение пароля. Данный процесс показан на рис. 2.

```
Searching: 5f4dcc3b5aa765d61d8327deb882cf99: FOUND: password5
Searching: 6cbe615c106f422d23669b610b564800: not in database
Searching: 630bf032efe4507f2c57b280995925a9: FOUND: letMEin12
Searching: 386f43fab5d096a7a66d67c8f213e5ec: FOUND: mcd0nalds
Searching: d5ec75d5fe70d428685510fae36492d9: FOUND: p@ssw0rd!
```

Рисунок.2. Взлом пароля с помощью заранее созданной таблицы хешей.

Похожий метод взлома называется обратным поиском по таблице. Это очень эффективный метод при больших объемах пользовательской БД. Эта атака позволяет злоумышленнику применять метод перебора паролей ко множеству хешей одновременно. Сначала создается поисковая таблица, где в соответствие каждому возможному значению хеша ставится один или более пользователей, имеющие данный хеш. Затем злоумышленник перебирает все

возможные варианты пароля, хеширует их и делает поиск пользователей с таким же значением хеша. Данная атака продемонстрирована ниже на рис. 3. Эффективность данной атаки еще выше из-за того факта, что несколько пользователей в большой БД обычно могут иметь один и тот же пароль.

```
Searching for hash(apple) in users' hash list... : Matches [alice3, 0bob0, charles8]
Searching for hash(blueberry) in users' hash list... : Matches [usr10101, timmy, john91]
Searching for hash(letmein) in users' hash list... : Matches [wilson10, dragonslayerX,
joe1984]
Searching for hash(s3cr3t) in users' hash list... : Matches [bruce19, knuth1337,
john87]
Searching for hash(z@29hjja) in users' hash list... : No users used this password
```

Рисунок.3. Взлом пароля методом обратного поиска.

Последний, наиболее изощренный и прогрессивный способ взлома – это использование так называемых радужных таблиц [3]. Радужная таблица является вариантом обычной таблицы поиска, но хранит в себе цепочки хешей, полученные с помощью чередования функции хеширования со специальной функцией редукции. Данный метод замедляет поиск совпадений в хеше, так как многие элементы таблицы приходится вычислять динамически. Плюс данной таблицы в снижении требования на необходимую память: при объеме для обычных таблиц в N слов для радужных нужно всего порядка N в степени $2/3$. Уже существуют таблицы, позволяющие легко взломать любой пароль длиной до 8 символов, зашифрованный при помощи хеш функции MD5 [4].

Анализ объекта защиты и выбор инструментов для обеспечения безопасности. Остановимся подробнее на особенностях данных, которые необходимо шифровать для собственно разработки многопользовательской игры. При ее создании была использована СУБД MongoDB. Все объекты хранятся в БД в виде JSON-объектов, и они не обязательно должны иметь одинаковый размер, что добавляет гибкости при дальнейшей разработке приложения. В базу данных при регистрации нового пользователя заносится новый объект, содержащий хеш пароля, так называемую соль (затравку для функции хеширования) и токен пользователя – еще один хеш. Запись структуры данных показана ниже на рис. 4.

```
{
  "_id": {
    "$oid": "58d153bcc2ef165bbfc76b8d"
  },
  "name": "MyPasswordIs1111",
  "nickname": "MyPasswordIs1111",
  "passwordHash":
  "3b780f6d2ede6ed47a8897c8031885b21e73b7f63ea5e036997f7f17b68a398c",
```

```

"r":
"cEtRWW18uznbZ5WTWWjHw2f0O00XjawxTj623i6pMe3djNJUN5GtcgdDX1B
zw2Qf",
"token":
"$2y$14$4t2RZU0DVv5MG3czFEjgeodyCV2uL3KuagkOc01F/ZdpP1shCEAW",
"id": 6,
"survey": {
  "e-mail": "an147job@gmail.com",
  "age": 0,
  "gender": "",
  "source": ""
},
"registerDate": "2017-03-21 19:24:27"
}

```

Рисунок.4. Структура данных о пользователе игрового приложения.

Как можно увидеть из рис. 4, основанная информация о пользователе это логин, электронный адрес почты, хэш пароля (элемент passwordHash) , соль хеша пароля (элемент r) и токен (элемент token).

Рассмотрим на этом примере, как сделать криптографически стойкое хеширование паролей. Согласно множеству источников [4,5] MD5 и SHA1 уже изжили свой век и не являются достаточно криптостойкими функциями хеширования. SHA2 «на очереди», однако, для простоты и скорости вычислений при авторизации пользователей был выбран именно этот алгоритм. Именно этим алгоритмом хешируется пароль пользователя при регистрации и каждый раз при попытке авторизации. Возможно, он будет заменен на более надежный и медленный алгоритм в дальнейшем. Однако, также намного более криптостойким хэш делает использование затравки.

Для любой функции хеширования в обязательном порядке рекомендуется добавлять затравку. Затравка – это просто случайная строка, которая добавляется перед хешируемым сообщением, или после него (от позиции затравки криптостойкость не зависит). Использование затравки дает намного больше преимуществ, чем кажется на первый взгляд: оно делает невозможными атаки по таблицам любого рода, оставляя злоумышленнику только возможность перебирать пароли в лексикографическом порядке, либо по словарю. Пример применения затравки показан ниже на рис. 5.

Таким образом, злоумышленник не может предугадать, какая затравка будет использована в каждом хеше пароля. Это влечет за собой невозможность построения радужных и иных таблиц для комбинаций всех возможных паролей со всеми возможными затравками.

Типичные ошибки при работе с затравками это использование одной и той же затравки и использование слишком короткой затравки. В первом случае

смысл использования затравки в целом пропадает, во втором – злоумышленник все же может построить таблицы для всех возможных значений слишком короткой затравки. Например, если в затравке всего 3 ASCII символа, то имеется всего $95 \cdot 95 \cdot 95 = 857,375$ возможных комбинаций. На первый взгляд это очень много, но если каждая таблица содержит около 1 Мб наиболее часто используемых паролей, то в сумме они займут 836 Гб. Это не так уж и много, если учесть, что сегодня можно приобрести жесткий диск на 1Тб по цене дешевле 100\$.

```

hash("hello") =
2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
hash("hello" + "QxLUF1bgIAdeQX") =
9e209040c863f84a31e719795b2577523954739fe5ed3b58a75cff2127075ed1
hash("hello" + "bv5PehSMFv11Cd") =
d1d3ec2e6f20fd420d50e2642992841d8338a314b8ea157c9e18477aaef226ab
hash("hello" + "YYLmfY6IehjZMQ") =
a49670c3c18b9e079b9cfaf51634f563dc8ae3070db2c4a8544305df1b60f007
    
```

Рисунок.5. Хеши сообщения hello без затравки и с разными затравками.

Для генерации случайных строк, как, собственно, для генерации любых случайных значений для криптосистемы, нужно пользоваться криптостойкими псевдогенераторами чисел, чтобы последовательность случайных величин была непредсказуемой, иначе криптостойкость системы падает. Такие генераторы можно найти в свободно распространяемых библиотеках на любом языке, например, в PHP 7 уже встроена функция `random_int` [6].

Стоит также отметить, что затравку как таковую нет смысла шифровать или прятать, это только усложнит процесс аутентификации пользователей с точки зрения приложения.

Для повышения уровня защищенности при использовании пары «логин - пароль» в веб-запросах в разрабатываемом игровом приложении, было решено создавать для каждого пользователя по токену. Токен в данном конкретном случае – это зашифрованная пара: логин и хеш пароля.

При этом возникает проблема, которая заключается в том, что приложение использует RESTful API для обращения к серверам с БД. Одно из требований RESTful API – это независимость от состояния, согласно источнику [7]. Таким образом, невозможно ввести понятие сессии, и приходится передавать данные для идентификации пользователя каждый раз при новом запросе к БД. Это обыденная практика для современных облачных сервисов. При регистрации пользователю выдается специальный хеш – секретный API ключ, который не следует разглашать. Этот ключ используется в каждом запросе к сервису, таким образом идентифицируя клиента. Пример подобного ключа можно увидеть ниже на рис. 6.

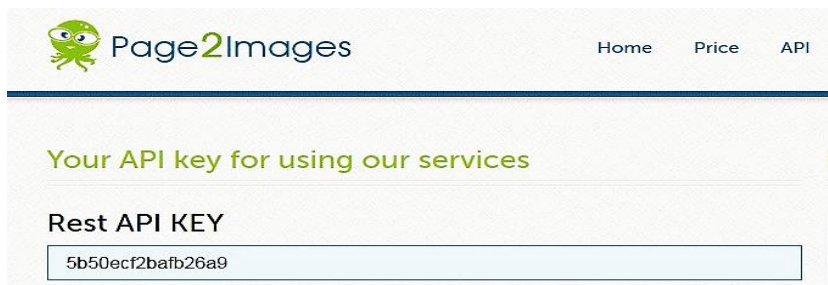


Рисунок.6. Ключ для идентификации клиента API веб-сервиса Page2Images.

Самый простой способ сделать RESTful запросы безопасными – использовать защищенное соединение [8]. В разрабатываемом игровом приложении это, конечно, обязательный пункт. Однако SSL/TLS не является панацеей, поэтому были предприняты дополнительные меры по безопасности, а именно передача токена вместо пары «логин-хеш пароля». При запросе к серверу в БД пользователей ищется пользователь с таким же токеном, и если он найден, то этот запрос идет от имени конкретного пользователя. Таким образом, даже если HTTPS запрос к RESTful API будет расшифрован при перехвате сообщения, злоумышленнику придется расшифровывать токен, чтобы иметь доступ к логину и хешу пароля. В целях безопасности токен создается лишь единожды при регистрации пользователя, и не меняется при смене пароля, следовательно, нельзя будет сделать запрос на присвоение токена определенному пользователю.

Так как операция хеширования (создания) токена проводится единожды, есть смысл применить более тяжелые современные алгоритмы хеширования, которые применяют технику «key stretching». Согласно [9], данная техника призвана сделать любой даже самый слабый пароль более криптостойким просто увеличив время на хеширование.

Рассмотрим данную технику на примере алгоритма bcrypt, который был выбран для создания токенов. Данный алгоритм принимает целочисленный аргумент, который отражает количество итераций алгоритма хеширования. Количество итераций получается возведением двойки в степень аргумента N , т.е. 2^N . Этот алгоритм адаптируется под любые нужды, например, при создании токена было задано достаточно большое количество итераций, $2^{14} = 16384$ итерации. Данную цифру можно изменять в зависимости от нагрузки на сервер, чтобы при регистрации множества пользователей не страдала производительность [10]. Например, если приложение весьма популярно и в секунду регистрируется 10 пользователей, то есть смысл подобрать такое количество итераций, чтобы хеширование занимало 100 мс.

Для создаваемого игрового приложения, с учетом его популярности, на сегодня достаточно оставить время хеширования в пределах 0.5 секунды. Для злоумышленника, перебирающего все возможные пароли, время отводимое на перебор будет просто неприемлемым, даже если у него в десятки раз мощнее оборудование. Действительно, если пароль состоит из всего четырех ASCII символов, имеется $256^4 = 42,949,967,296$ комбинаций. Допустим, злоумышленник располагает аппаратными ресурсами в 100 крат мощнее, чем веб-сервер, на котором происходит формирование токена, тогда каждая попытка займет около 5 мс. Для перебора всех возможных вариантов ему потребуется 21,474,836 секунд, или 248,5 суток (8 месяцев). При использовании трех символов перебор возможен немного менее, чем за сутки, поэтому минимальную длину пароля следует строго ограничивать.

Кроме рассмотренного алгоритма были проанализированы алгоритмы PBKDF2, crypt и scrypt, и оценена целесообразность их применения. Эти алгоритмы также используют технику key stretching. Однако, bcrypt немного лучше того же PBKDF2 по той причине, что рассчитан на использование злоумышленником графического процессора (GPU), который в некоторых случаях может давать прирост по сравнению с CPU в несколько десятков раз. Bcrypt настроен на обращения к таблице подстановок, которая постоянно меняется во время выполнения алгоритма, тем самым заставляя все ядра GPU конкурировать за общую память в шине [11]. Отсюда следует, что прирост производительности от использования GPU не будет таким огромным.

Формирование политики безопасности. Кроме рассмотренных факторов на безопасность игрового сетевого приложения естественно влияет и длина пароля. Как и в других сетевых приложениях, минимальная длина пароля должна быть ограничена, так как при небольших паролях эффективность защиты сводится к нулю, как сказано выше. Обычно этот минимум равен 8 символам, что является разумным и не сильно утруждает пользователей.

Политика парольной защиты игрового приложения основывается на рекомендациях, приведенных в [4]. При попытках авторизации целесообразно не делать разделения на ошибку в логине и ошибку в пароле, а рассматривать их как единую пару. Не имея на руках БД пользователей, злоумышленник не сможет через интерфейс перебором подобрать логин и пароль [4]. Политика восстановления пароля должна быть основана на временных токенах. При получении письма, пользователю должна быть предоставлена ссылка с данным токеном, где он может поставить себе новый пароль. Рекомендуется привязывать токен к конкретному аккаунту, чтобы злоумышленник не мог имея на руках «свой» токен поменять чужой пароль. Время действия токена должно истекать в течение 15 минут со времени запроса на восстановление пароля. Также токен должен аннулироваться, когда данный пользователь авторизовался в системе (он вспомнил пароль). В токене не должна быть

отражена информация о таймауте и прочие параметры, необходимые системе, так как злоумышленник будет способен заменить данные параметры на нужные. Никогда не следует посылать новый пароль пользователю по почте. Также нужно использовать новую затравку для хеширования нового пароля.

Выводы. Таким образом, в статье рассматриваются проблемы повышения защищенности игрового веб-приложения с учетом различных видов атак на пользовательскую БД и проведен анализ существующих решений для обеспечения надежной парольной защиты. Также была предложена политика безопасности в отношении паролей пользователей. Данные решения были внедрены в разрабатываемое сетевое приложение.

Список литературы

1. «Борьба с читерами в онлайн-играх: 22 «нужно» и «нельзя»» // Habrahabr. [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/post/320602/>
2. «Kerckhoffs's principle» // Wikipedia. [Электронный ресурс]. Режим доступа: https://en.wikipedia.org/wiki/Kerckhoffs's_principle
3. «Rainbow table» // Wikipedia. [Электронный ресурс]. Режим доступа: https://en.wikipedia.org/wiki/Rainbow_table
4. «Salted Password Hashing - Doing it Right» // CrackStation. [Электронный ресурс]. Режим доступа: <https://crackstation.net/hashing-security.htm>
5. «Why not use MD5 for password hashing?» // StackOverflow. [Электронный ресурс]. Режим доступа: <http://stackoverflow.com/questions/30496061/why-not-use-md5-for-password-hashing>
6. «PHP 7 Documentation: random_int» // PHP.net. [Электронный ресурс]. Режим доступа: http://www.php.net/random_int
7. «RFC 7235. Hypertext Transfer Protocol (HTTP/1.1): Authentication» // IETF.org. [Электронный ресурс]. Режим доступа: <https://tools.ietf.org/html/rfc7235#section-4.2>
8. «How do I secure my REST API?» // security.stackexchange.com. [Электронный ресурс]. Режим доступа: <https://security.stackexchange.com/questions/19930/how-do-i-secure-my-rest-api>
9. «Key stretching» // Wikipedia. [Электронный ресурс]. Режим доступа: https://en.wikipedia.org/wiki/Key_stretching
10. «Recommended # of iterations when using PKBDF2-SHA256?» // security.stackexchange.com. [Электронный ресурс]. Режим доступа: <https://security.stackexchange.com/questions/3959/recommended-of-iterations-when-using-pkbf2-sha256/3993#3993>
11. «Do any security experts recommend bcrypt for password storage?» // security.stackexchange.com. [Электронный ресурс]. Режим доступа: <https://security.stackexchange.com/questions/4781/do-any-security-experts-recommend-bcrypt-for-password-storage?rq=1>

УДК 004.942

ИССЛЕДОВАНИЕ ПРОГРАММНЫХ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ ПРЯМОГО РАСПРОСТРАНЕНИЯ

Медгаус С.В., Федяев О.И.

Донецкий национальный технический университет
кафедра программной инженерии

E-mail: medgaus-sergey@yandex.ru, fedyayev@donmtu.org

Аннотация

Медгаус С.В., Федяев О.И. Исследование программных моделей нейронных сетей прямого распространения. В данной статье рассмотрены наиболее распространённые стратегии обучения нейронных сетей прямого распространения, а также приведены программные модели нейросетей, для которых применены эти алгоритмы и приведён анализ их эффективности.

Ключевые слова: нейронные сети

Введение

Многие задачи, которые решаются в настоящее время с помощью компьютерных систем, с точки зрения формализма разработки алгоритма решения, относятся к классу трудно формализуемым (а иногда и к неформализуемым) задачам. Для них невозможно чётко определить логику функционирования программы для решения поставленной задачи. К таким задачам относятся задачи распознавания классификации образов, сжатия данных, идентификации и управления сложными объектами, аппроксимации и интерполяции, прогнозирования динамики процессов и т. д. [1]. Для решения таких задач стали успешно применяться искусственные нейронные сети благодаря их способности к обучению и к обобщению полученных знаний. В настоящее время разработано много типов нейронных сетей для указанного класса прикладных задач.

Среди множества существующих типов сетей в качестве важнейших можно выделить нейронные сети прямого распространения (с прямой связью). Согласно теоретическим результатам нейронные сети с прямой связью и с сигмоидальными функциями активации являются универсальным средством для приближения (аппроксимации) произвольных функций. Однако нет никакого правила, позволяющего найти оптимальную топологию сети для решаемой задачи. Таким образом, задача построения адекватной нейронной сети является нетривиальной. Вопросы о том, сколько нужно взять скрытых слоёв, сколько нейронов в каждом из них, какие связи, в имеющейся литературе, как правило, трактуются бездоказательно и предлагаются решать путём простого перебора различных архитектур.

Другая проблема связана с правильной организацией процесса обучения нейронной сети [2]. Насколько качественно он будет организован, зависит способность сети решать поставленные перед ней задачи во время штатного функционирования. Ключевым вопросом в обучении является выбор алгоритма настройки нейросети. Все подобные алгоритмы основаны на методах оптимизации нелинейной многомерной функции ошибки, поэтому длительность процесса обучения и его качество зависит не только от применяемого метода оптимизации, но и от других факторов (особенностей структуры нейросети, обучающего множества). Разработано уже более сотни разных обучающих алгоритмов, отличающихся друг от друга стратегией оптимизации и критерием ошибок. Для задачи многомерной оптимизации предложен большой спектр методов, начиная от градиентного метода наискорейшего спуска и заканчивая методом Монте-Карло и генетическими алгоритмами.

Описанные проблемы приводят к тому, что при практической работе с нейронными сетями прямого распространения приходится экспериментировать с большим числом различных сетей, порой обучая каждую из них по нескольку раз и сравнивая полученные результаты. Главным показателем качества результата является здесь контрольная ошибка. При этом, в соответствии с общесистемным принципом, из двух сетей с приблизительно равными ошибками контроля имеет смысл выбрать ту, которая проще.

Несмотря на значительное количество уже известных практических приложений нейросетей прямого распространения, возможности их дальнейшего использования для обработки сигналов не изучены окончательно. Поэтому цель данной научной работы состоит в разработке программных моделей нейросетей прямого распространения и исследовании их ключевых свойств (представления и обучения) для разработки на их основе качественных программных агентов с нейросетевой архитектурой. Для этого рассмотрены две типовые структуры сетей прямого распространения и алгоритмы их обучения по стратегии «обучение с учителем».

1. Однослойная сеть прямого распространения (однослойный персептрон)

Типовая структура однослойного персептрона, ориентированная на решение задачи распознавания графических символов показана на рис. 1.

Для обучения нейросети применялась стратегия контролируемого обучения: «учитель» подаёт на вход сети оцифрованный образ буквы, а на выход сообщает желаемое значение результата распознавания (буква гласная или согласная). Такой набор контрольных примеров был сгруппирован в обучающее множество $\{(X, Y)\}$. В ходе обучения по алгоритму дельта-правило

вычисляется ошибка δ (дельта), как разность между требуемым сигналом и фактическим выходом.

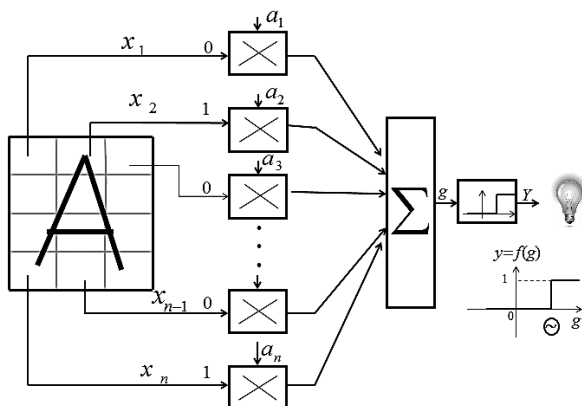


Рисунок 1 – Модель персептрона

$$\delta = TARGET_OUT - ACTUAL_OUT \quad (1)$$

Полученная ошибка служит основой для корректировки весов нейрона. После получения значения ошибки для каждого из нейронов (если их несколько в слое) вычисляется значение новых весов на основании этой ошибки и параметра скорости обучения (2).

$$a_j(i + 1) = a_j + \eta \delta x_j \quad (2)$$

где i - номер текущей итерации (эпохи) обучения персептрона; j – номер синаптической связи; η - коэффициент скорости обучения, позволяет управлять средней величиной изменения весов ($0 < \eta < 1$); x_j - величина входа соответствующая a_j синаптическому весу.

Сходимость алгоритма (1-2) исследовал Ф.Розенблатт [3], а также другие авторы в более поздних публикациях. Данный алгоритм обучения хороший, но он применим только для однослойных нейросетей, в которых ошибка непосредственно влияет на единственные веса, ведущие ко входам в нейронной сети [2].

Для исследования данной нейросети была разработана её программная модель, которая реализует один нейрон с одним выходом и многими входами с соответствующими весовыми коэффициентами [3] (см. рис. 1). В обучающее множество включены изображения первых 10 букв с разным начертанием и задан признак гласности каждой из букв. После этого НС была обучена по

рассмотренному алгоритму на данных примерах с приемлемым уровнем точности (до 100% распознавания) (см. рис. 2).

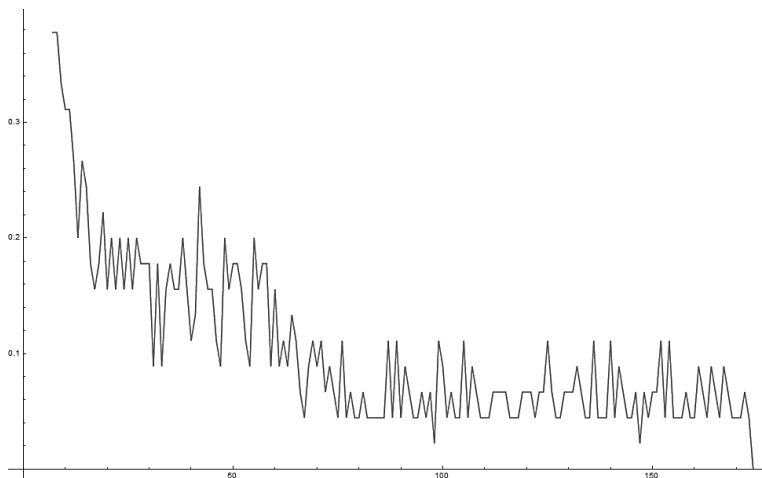


Рисунок 2 – Уменьшение ошибки распознавания в ходе обучения перцептрона

При тестировании программной модели нейросети на её вход подавались новые образы букв, незнакомые системе, и система выдавала в 80% правильные результаты распознавания (см. рис. 3 - 4).

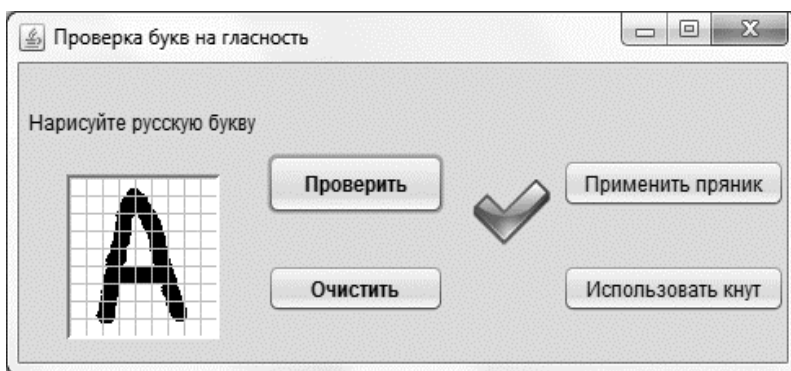


Рисунок 3 – Работа системы по проверке гласности. Буква А

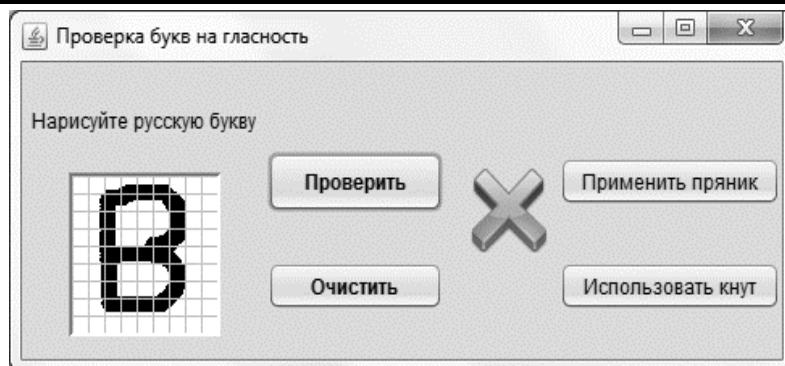


Рисунок 4 – Работа системы по проверке гласности. Буква В

Также разработанная программная модель позволяет «дообучать» нейросеть с новыми образами, для этого пользователю необходимо нарисовать новый образ и запустить проверку. Если система выдаёт правильный результат, то «одобрить» её вычисления. Иначе нужно «пожурить» её, чтобы в следующий раз при обучении она не делала неправильных вычислений.

2. Многослойная сеть прямого распространения (многослойный персептрон)

Для обучения многослойных нейронных сетей был программно реализован эффективный алгоритм обратного распространения ошибки (Backpropagation), представляющий собой развитие обобщённого дельта-правила [4,5]. Этот алгоритм считается наиболее известным и чаще всего применяемым в искусственных нейронных сетях прямого распространения. Эта стратегия обучения применима как для однослойных, так и для многослойных НС. Ошибка из выходного слоя НС последовательно распределяется и передаётся внутренним скрытым слоям, в итоге достигая первого скрытого слоя [4]. Основные действия при обратном распространении ошибки представлены на рис. 5 [5].

Для исследования этого типа нейросети также была разработана программная модель. Структура многослойной нейросетипоказана на рис. 6. Она представляет собой трёхслойную НС, которая использует для обучения алгоритм обратного распространения ошибки. Качество данной нейросети оценивалось на решении задачи прогнозирования динамического процесса.

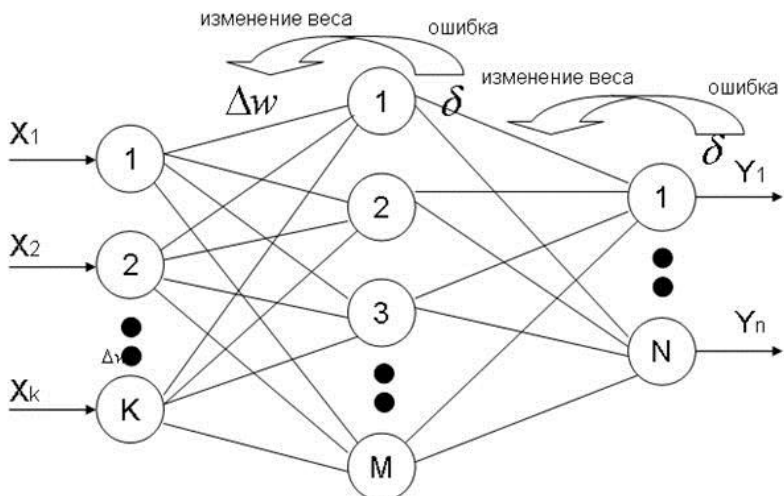


Рисунок 5- Основные действия НС при обратном распространении

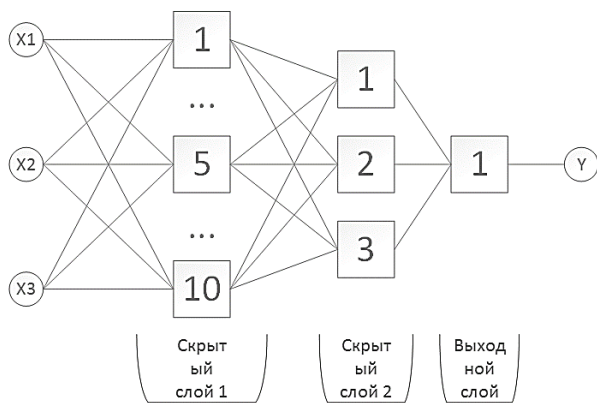


Рисунок 6 – Структура многослойной сети

Динамика процесса описывалась синусоидальной функцией. В эксперименте использовалась функция $\sin(x)$. Для обучения НС функция была разбита на отрезки и с помощью метода окон (Windowing) было построено обучающее множество[6]. При обучении была достигнута ошибка в 0.005%. С целью ускорения процесса обучения сети предложены многочисленные модификации алгоритма обратного распространения ошибки, связанные с

использованием различных функций ошибки, процедур определения направления и величины шага.

На рис. 7 представлена динамика обучения нейронной сети в зависимости от количества эпох обучения.

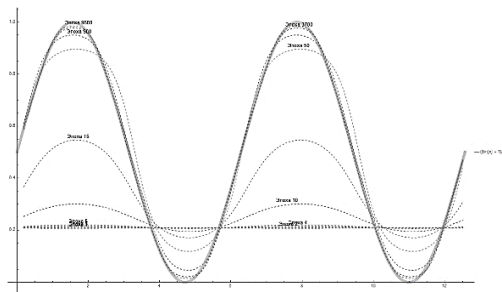


Рисунок 7- Результатынастройки многослойной нейронной сети на синусоиду по эпохам

Выводы. В ходе исследования были проанализированы различные структуры нейронных сетей и методы их обучения. Также были разработаны программные модели вышеуказанных нейросетей с целью проведения экспериментов по анализу их эффективности.

В результате был подтвержден тезис о способности нейросетей распознавать образы и выявлены основные достоинства и недостатки различных алгоритмов обучения нейросетей прямого распространения.

Список литературы

1. Искусственные нейронные сети / Интернет-ресурс. - Режим доступа: [www/ URL: http://www.neuralnet.info/глава-1-введение/](http://www.neuralnet.info/глава-1-введение/).
2. Алгоритм обучения НС прямого распространения по дельта-правилу / Интернет-ресурс. - Режим доступа: [www/ URL: http://www.aiportal.ru/articles/neural-networks/perceptron-learning.html](http://www.aiportal.ru/articles/neural-networks/perceptron-learning.html).
3. Структура персептрона. - Режим доступа: [www/ URL: http://www.neuralnet.info/глава-3-персептроны/](http://www.neuralnet.info/глава-3-персептроны/).
4. Алгоритм обратного распространения ошибки в многослойных НС. - Режим доступа: [www/ URL: http://www.gotai.net/documents/doc-nn-009-06.aspx](http://www.gotai.net/documents/doc-nn-009-06.aspx).
5. Основные действия при обратном распространении ошибки. - Режим доступа: [www/ URL: http://apsheronk.bozo.ru/Neural/Lec3.files/image030.jpg](http://apsheronk.bozo.ru/Neural/Lec3.files/image030.jpg)
6. Метод плавающего окна. - Режим доступа: [www/ URL: http://masters.donntu.org/2006/kita/kiryan/library/art06.htm](http://masters.donntu.org/2006/kita/kiryan/library/art06.htm)

УДК 004

РАЗРАБОТКА КЛИЕНТСКОГО ПРИЛОЖЕНИЯ ДЛЯ ШИФРОВАНИЯ И ДЕШИФРОВАНИЯ ФАЙЛОВ С ИСПОЛЬЗОВАНИЕМ СЕТИ ФЕЙСТЕЛЯ

Э.Э. Ворожбитов, В.А. Павлий

Донецкий национальный технический университет
кафедра компьютерного моделирования и дизайна
r.r.1911@mail.ru

Аннотация

Ворожбитов Э.Э., Павлий В.А. Разработка клиентского приложения для шифрования и дешифрования файлов с использованием сети Фейстеля. В данной статье представлены понятия шифрования и алгоритма сети Фейстеля. Рассмотрены аналоги и общий их недостаток. Даны описания алгоритма и приложения. В конце подведены выводы.

***Ключевые слова:** шифрование, дешифрование, алгоритм, сеть Фейстеля, клиентское приложение.*

Постановка задачи. В современном мире с множеством разнообразной информации существует угроза её похищения или раскрытия. Для предотвращения этого одним из лучших способов является шифрование данных для сохранения конфиденциальности информации и несанкционированного ознакомления с ними.

Практически все существующие программы для шифрования имеют общий минус – невозможность детальной конфигурации шифра.

Исходя из этого, является необходимой программа, в которой может быть произведена индивидуальная для каждого пользователя конфигурация шифрования.

Аналогов, использующих для шифрования сеть Фейстеля, в открытом доступе нет. Существует приложения, которые используют модифицированные от сети Фейстеля шифры или совсем иные алгоритмы[1], но у всех них нет возможности подробной настройки шифрования.

Описание алгоритма. Сеть Фейстеля — один из методов построения блочных шифров. Сеть состоит из ячеек, называемых ячейками Фейстеля. На вход каждой ячейки поступают данные и ключ. На выходе каждой ячейки получают изменённые данные и изменённый ключ. Все ячейки однотипны, и говорят, что сеть представляет собой определённую многократно повторяющуюся (итерированную) структуру. Ключ выбирается в зависимости от алгоритма шифрования/расшифрования и меняется при переходе от одной

ячейки к другой. При шифровании и расшифровании выполняются одни и те же операции; отличается только порядок ключей[2].

Полученные данные делятся на блоки фиксированной длины, которая, как правило, равна значению степени двойки.

Первый блок делится на два одинокого размера подблока – “левый” (L_0) и “правый” (R_0).

Затем “левый” подблок обрабатывается в функции $f(L_0, K_0)$, где K_0 – ключ, и производит операцию “XOR” с “правым” подблоком R_0 . Результат данного преобразования ставится на место нового “левого” подблока L_1 . На место нового “правого” подблока R_1 ставится без изменений “левый” подблок L_0 .

После этого данные действия производятся $N-1$ раз, где N – заданное в алгоритме количество раундов. При этом ключ K изменяется с каждым раундом по определённому правилу.

Дешифрование данных производится аналогично шифрованию, с одним отличием: ключи будут идти в противоположном порядке, то есть от последнего к первому.

Таким алгоритмом зашифровывается весь файл. Если длина выбранного блока меньше заданного размера, тогда блок увеличивается до нужной длины посредством дописывания сзади нулей.

Общая структура алгоритма сети Фейстеля показана на рисунке 1.

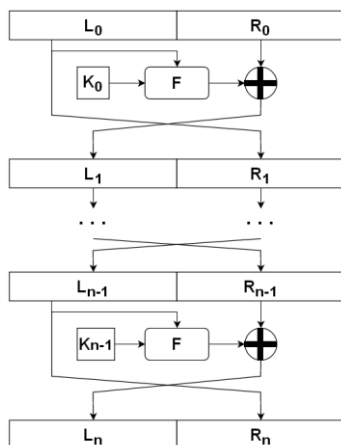


Рис.1 – общая структура алгоритма сети Фейстеля

Описание приложения. Шифрование – это преобразование информации, делающее ее нечитаемой для посторонних. При этом доверенные лица могут провести дешифрование и прочесть исходную информацию. Существует множество способов шифрования/дешифрования, но секретность данных

основана не на тайном алгоритме, а на том, что ключ шифрования (пароль) известен только доверенным лицам[3].

Для разработки приложения используется язык программирования C#. Также с помощью него был реализован интерфейс, который позволяет пользователю загружать в программу файлы для шифрования, выбирать и настраивать метод шифрования, прогресс выполнения которого можно наблюдать, и выгружать зашифрованный файл.

Для описания возможных взаимодействий пользователя с приложением и указания функционального назначения программы существует диаграмма прецедентов, которая показана на рисунке 2.

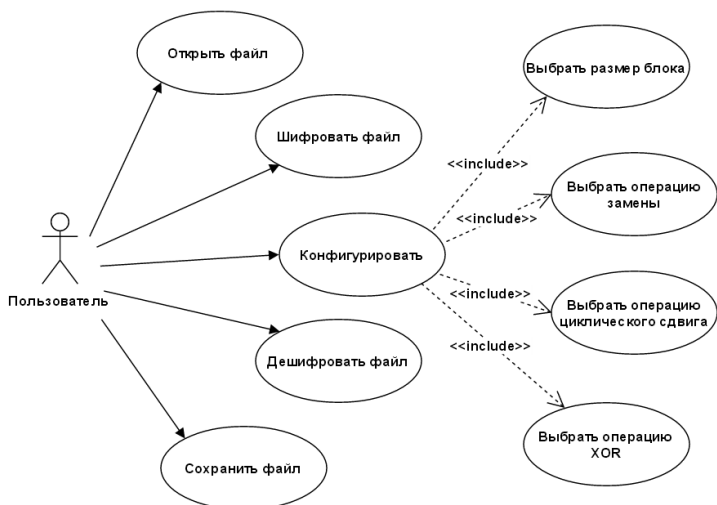


Рисунок 2 – Диаграмма прецедентов

В начале работы с программой выбирается существующий файл. В случае выбора, не существующего файла, появится ошибка, сообщающая об этом.

Затем производится конфигурирование:

- выбор размера блока, который должен быть равен значению степени двойки;
- выбор одного или нескольких методов шифрования;
- выбор операции замены, определив какие именно элементы будут подвергаться данному действию;
- выбор операции циклического сдвига, с указанием, на какое количество и в какую сторону необходимо сдвигать;

- выбор операции “XOR”, обозначив элемент, и то с чем он будет выполнять данную операцию: с иным элементом или с каким-либо символом или знаком.

После этого происходит процесс шифрования файла, используя сеть Фейстеля с пользовательскими настройками алгоритма.

Для дешифровки файлов необходимо совершить аналогичные действия, только выбрать соответствующую данной операции функцию программы – дешифрование.

В программе предусмотрен визуальный показатель прогресса шифрования и дешифрования файлов, который демонстрируется одновременно с выполнением соответствующей его обозначению операцией.

Общий алгоритм работы приложения показан на рисунке 3.



Рисунок 3 – Общий алгоритм работы приложения

Выводы. Защита информации является одной из самых важных и первостепенных задач, стоящих перед IT-специалистами. Именно для этой цели разрабатывается приложение для шифрования и дешифрования файлов.

Конечно, технологии, как и время, не стоят на месте, поэтому в дальнейшем планируется развитие данной программы.

Список литературы

1. Криптография. Алгоритмы симметричного шифрования [Электронный ресурс] – URL: http://alexinternetcllc.ru/Kriptograf_1.php
2. Википедия – Сеть Фейстеля [Электронный ресурс] – URL: https://ru.wikipedia.org/wiki/Сеть_Фейстеля
3. Блог Лаборатории Касперского. Зачем нужно шифрование? [Электронный ресурс] – URL: <https://blog.kaspersky.ru/encryption-reasons/879/>

Секция № 3
Системы
автоматизированного
проектирования



УДК 62-529

3D-ФРЕЗЕРНЫЙ СТАНОК С ЧПУ ПОД УПРАВЛЕНИЕМ LABVIEW

Кухта С.С., Бобелюк М.Б., Мезенцев Д.А., Харитонов А.Ю.
Донецкий национальный технический университет, г. Донецк
Кафедра компьютерного моделирования и дизайна
kukhtass@mail.ru

Аннотация

Кухта С.С., Бобелюк М.Б., Мезенцев Д.А., Харитонов А.Ю. 3D-фрезерный станок с ЧПУ под управлением LabVIEW. На базе конструктора Lego Mindstorm собран 3D-фрезерный станок. Разработано программное обеспечение для управления работой станка на базе LabVIEW.

Ключевые слова: фрезерный станок, числовое программное управление, визуальное программирование, LabVIEW, Lego NXT.

Постановка задачи. Фрезерный станок – станок, использующийся при обработке деревянных, металлических и других изделий. Поскольку данное оборудование дорогостоящее и громоздкое, в рамках данного проекта был создан более портативный вариант станка. В ходе работы над проектом были поставлены следующие задачи:

- на базе конструктора Lego Mindstorm собрать полноценный 3D-фрезерный станок;
- разработать программное обеспечение для управления работой на базе NI LabVIEW.

Цель статьи – описать конструкцию данного станка, рассмотреть возможные варианты реализации программного обеспечения и написать программу для управления работой данного устройства.

Описание конструкции. Станок представляет собой трёхуровневое основание станка, свободно перемещающееся по двум осям, а также рамку, к которой присоединяется фреза. Нижняя платформа неподвижно присоединена к основанию. По краям данной платформы расположены шестерёнки, благодаря которым осуществляется перемещение платформы второго уровня вдоль оси X. Движение осуществляется за счет сервомоторов, которые вращают валы с шестерёнками. Платформы перемещаются по шестерёнкам за счет зубчатых балок, которые накладываются на шестерёнки (рис.1). На платформу третьего уровня закрепляется материал, который подлежит фрезеровке. Движение

основания ограничено датчиками движения, расположенными на критическом для каждой оси расстоянии.

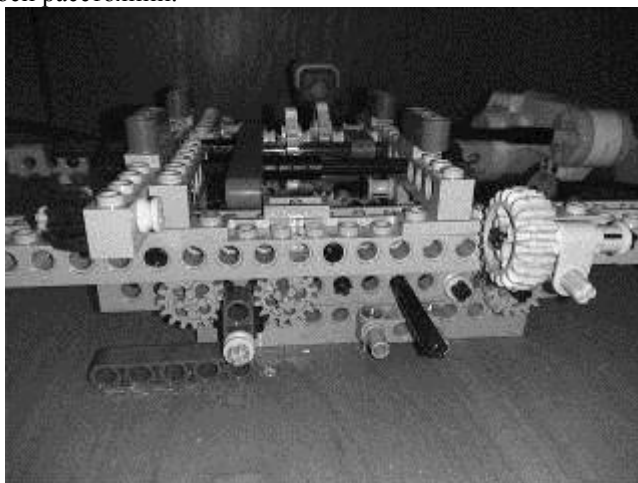


Рисунок 1 – Основание конструкции, платформы первого и второго уровня

Над основанием располагается П-образная арка, к которой закреплена фреза. В качестве фрезы используется гравер (многофункциональный инструмент) с гибким валом.

Описание программного кода. Программный код написан в среде LabView. Интерфейс программы представляет собой прямоугольную плоскость, на которой пользователь может изобразить рисунок, который в последствии будет выполнен на материале с помощью станка. До непосредственного начала работы самой фрезы идет определение начальных координат, платформа сперва двигается вдоль оси OX , пока не коснется датчика касания, затем производится откат на исходное положение и движение по оси OY до датчика. Результаты отклонения платформы от начального значения по каждой оси выводятся на приборную панель программы (рис.1).

После выставления начального положения опускается фреза и платформа перемещается по массиву точек, образованному при рисовании пользователем фигуры. С помощью расчета коэффициентов на моторы подается оптимальная для работы фрезы мощность, с которой перемещаются платформы. Кроме того, массив точек хранится внутри программы, что позволяет использовать его в случае необходимости (рис.3).

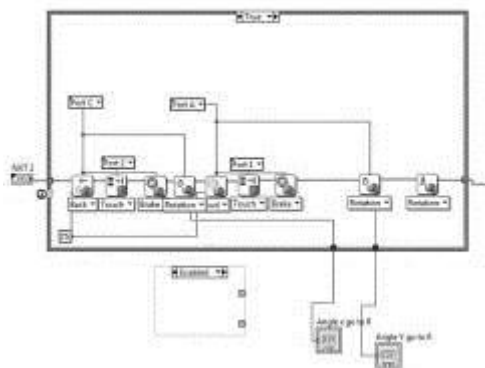


Рисунок 2– Определение начального положения

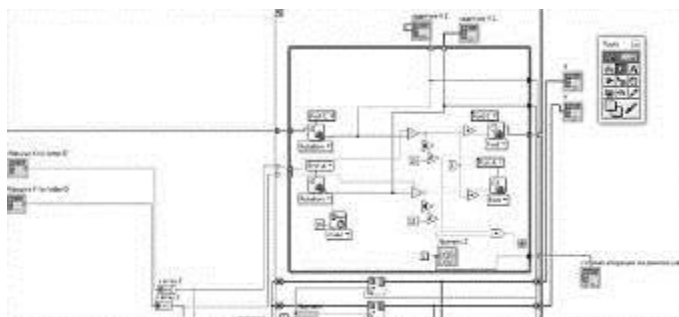


Рисунок 3 – Передвижение платформ согласно массиву точек

В программе можно выполнить любой рисунок с помощью оси координат, расположенной на лицевой панели. В качестве примера работы станка использовался выполненный нарисованный в программе логотип Донецкого Национального технического университета (рис. 4). Неточности изображения связаны непосредственно с процессом рисования самого логотипа, фрезерный станок чётко перенес изображение согласно полученным данным.

Актуальность работы. Актуальность работы заключается в том, что фрезерный станок является универсальным средством обработки любых деревянных, металлических и других видов изделий. Поскольку полноценный станок достаточно дорогой и громоздкий, преимуществами данной установки является то, что она собрана из деталей Lego, что позволяет легко найти или заменить недостающие детали. Кроме того, строение конструкции позволяет изменять насадку для гибкого вала, не разбирая ее целиком. Данная работа была представлена на выставке Малой Академии Наук ДНР в феврале 2017 года, где

заняла третье место в категории «Лучший групповой проект учащихся высших учебных заведений» (рис.5).

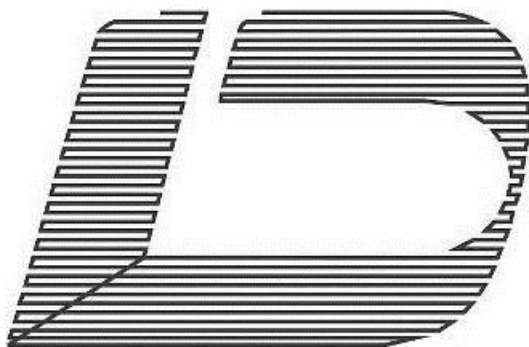


Рисунок 4 – Логотип ДонНТУ, который используется в качестве примера

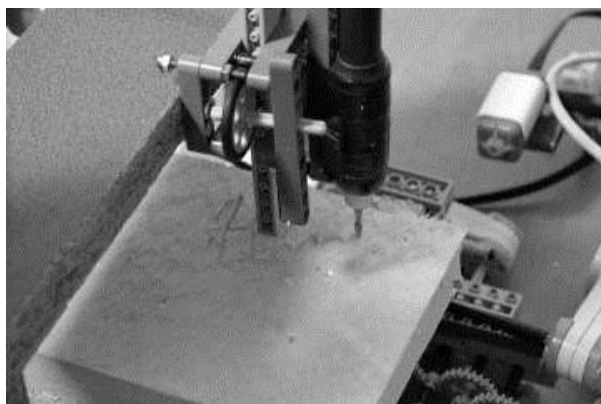


Рисунок 5 – презентация работы станка на выставке НТТМ МАН-2017

Выводы о работе. В результате выполнения данного проекта были развиты конструкторские навыки, умение моделировать объекты, аналитически рассчитывать необходимые ресурсы. Кроме того, усовершенствованы навыки написания программного кода в среде LabView.

Список литературы

1. Числовое программное управление [электронный ресурс] // Википедия – свободная энциклопедия [сайт] [2017]. URL: [www.wikipedia.org/wiki/Числовое программное управление](http://www.wikipedia.org/wiki/Числовое_программное_управление)
2. Суранов А. Я. LabVIEW 8.20: Справочник по функциям. – М.: ДМК Пресс, 2007. – 536 с.

УДК 622:658.011.56

ПРОЕКТИРОВАНИЕ ГОРНОТЕХНИЧЕСКИХ СИСТЕМ НА ОСНОВЕ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

Скаженик В.Б.

Донецкий национальный технический университет
горный факультет, кафедра управления производством
geosoft.svb@gmail.com

Аннотация

Скаженик В.Б. Проектирование горнотехнических систем на основе компьютерного моделирования. Рассмотрены вопросы моделирования месторождений полезных ископаемых и решения проектных задач на основе модели. Выделены общие функции построения модели любых горнотехнических систем. Представлены результаты моделирования золоторудного карьера и угольной шахты.

***Ключевые слова:** компьютерная модель месторождения, проектирование, горная выработка, залежь полезного ископаемого.*

Общая постановка проблемы. Динамично изменяющиеся условия функционирования горнодобывающих предприятий предопределяют необходимость пересмотра проектных решений на всех стадиях развития горнотехнической системы. С другой стороны, бурное развитие информационных технологий создает предпосылки для совершенствования методов принятия решений и повышения качества проектных решений в целом.

Горнодобывающее предприятие как объект управления представляет собой сложную динамическую систему, функционирующую в условиях существенной изменчивости факторов внешней и внутренней среды.

Для обеспечения стабильного, устойчивого функционирования горнотехнических систем при постоянном изменении характеристик среды и пространственной конфигурации собственно объекта управления необходимо решение многофакторной задачи в условиях неопределённости.

Инерционность горнотехнических систем также осложняет поиск наилучших решений и своевременную корректировку управляемых переменных.

Целесообразность и эффективность разработки месторождений определяется не только качеством принимаемых решений на этапе проектирования и функционирования предприятия. Стадии развития горных проектов включают: геологоразведочные работы; строительство горнодобывающего предприятия; выход на проектную мощность; работа на уровне проектной мощности; затухание горных работ - доработка запасов;

рекультивация и закрытие (консервация) предприятия. В процессе разработки за счёт средств предприятия может производиться доразведка запасов, а также капитальные работы, обеспечивающие доступ к оставшимся запасам. Поэтому оценка таких проектов должна учитывать всю совокупность факторов, влияющих на всех стадиях жизненного цикла горного проекта.

В силу указанных обстоятельств на практике принимаемые решения по управлению горными проектами, как правило, далеки от оптимальных. Эффективность горного производства можно существенно повысить за счёт выработки обоснованных, системно увязанных проектных решений. Однако для научного обоснования всего комплекса решений необходимо построение адекватных моделей развития горного проекта, учёта всех существенных взаимосвязей между подсистемами и звеньями горнодобывающего предприятия а также компьютерная реализация моделей принятия решения.

Специализированные компьютерные системы. Для решения задач горного производства на различных стадиях развития горнодобывающего предприятия применяются компьютерные системы различной степени интеграции функций. Как правило, известные интегрированные системы горного производства предполагают пространственное моделирование месторождения и включают совокупность стандартных задач геологии, маркшейдерии, проектирования, планирования [1]. Среди геологов и горняков широко распространены такие системы как Micromine, Geovia (Gemcom, Surpac), Datamine, Vulcan.

На постсоветском пространстве для моделирования месторождений применяется система K-MINE. Система позиционируется как геоинформационная система с возможностью решения проектных задач. Возможно проектирование элементов горных работ при открытом и подземном способах отработки [2].

Интегрированные системы для решения геологических, проектных, производственных и экономических задач разрабатываются, как правило, с возможностью применения на различных видах месторождений. В рамках такого подхода в наибольшей степени удается обобщить задачи построения геологической модели месторождения. Многообразие условий и технологических особенностей разработки месторождений затрудняет создание типовых модулей для решения проектных и производственных задач.

В этой связи разработка программного обеспечения в горном деле должна быть тесно связана с типизацией горнотехнических систем, выделением общих и частных характеристик систем с целью соответствия структуры программного обеспечения особенностям объектов и решаемым задачам.

В таблице 1 на примере системы Surpac представлены типичные функциональные возможности модулей по созданию геологической модели

месторождения и по автоматизации проектирования в интегрированных системах для горного производства.

Несмотря на широкое распространение перечисленных систем для рудных месторождений, эти системы практически не используются на угольных шахтах. Причины – как относительно высокая стоимость, так и отсутствие специального функционала для угольных шахт.

Таблица 1. Функциональные возможности интегрированных систем для горного производства

Виды функциональных модулей	Функциональные возможности
Модули по созданию геологической модели месторождения	Статистический анализ геологической информации
	Моделирование залежей полезного ископаемого
	Подсчет запасов различными методами
	Построение блочной геологической модели
Модули по автоматизации проектирования	Проектирование сетей взрывных скважин, включая оконтуривающие наклонные скважины
	Дизайн карьеров и отвалов в интерактивном режиме
	Проектирование дорог
	Оптимизатор Карьера (трехмерный алгоритма Лерча-Гроссмана)
	Проектирование подземных взрывных скважин
	Проектирование подземных горных выработок

В процессе использования интегрированных систем возникают противоречия между стандартным функционалом и особенностями конкретных месторождений.

Цель данной статьи – анализ подходов к построению моделей различных горнотехнических систем и выделение общих функций программного обеспечения при разработке систем автоматизированного проектирования.

Технология построения модели месторождения. Для условий разработки рудных и россыпных месторождений с участием сотрудников ДонНТУ разработан программный комплекс «Рудник – 3D» [3]. Модули этого программного комплекса используются также для моделирования карьеров и угольных шахт.

Рассмотрим общую технологию построения геологической модели месторождения и проектных горных выработок в программном комплексе «Рудник – 3D» на примере создания модели проектируемого золоторудного карьера и угольной шахты.

При разработке системы были выделены функции моделирования, применимые для различных горнотехнических систем: для открытой разработки россыпных и рудных месторождений; для подземной разработки золоторудных месторождений и пластовых месторождений каменного угля. Перечень таких общих функций моделирования представлен в таблице 2.

Таблица 2. Моделирование объектов горнотехнической системы

Функции моделирования	Варианты выполнения	Условное обозначение функции
Моделирование поверхностей	По изолиниям (изогипсам пласта)	F1.1
	По отдельным точкам с указанием координаты Z точек	F1.2
Моделирование залежи полезного ископаемого	По разрезам	F2.1
	По двум поверхностям (почвы и кровли – для пластовых месторождений)	F2.2
Моделирование существующих горных выработок	Создание выработки по бортам	F3.1
	Создание выработки по разрезам	F3.2
Моделирование проектных горных выработок	Выбор сечения и указание трассы с привязкой к поверхности	F4.1
	Проведение трассы выработки в заданной плоскости	F4.2

На рис. 1 и рис. 2 представлены модели карьера и угольной шахты с указанием функций моделирования объектов в соответствии с обозначением их в таблице 2.

Проведенный анализ геологических и технологических особенностей разработки различных месторождений позволил обосновать общую структуру базы данных программного комплекса.

Реализация в программном комплексе взаимосвязей между условиями разработки месторождения, принятыми технологическими вариантами, производственной мощностью и последовательностью разработки месторождения, потребностями ресурсов по вариантам, экономическими и финансовыми показателями обеспечивает оперативную оценку и выбор решений по развитию горного проекта.

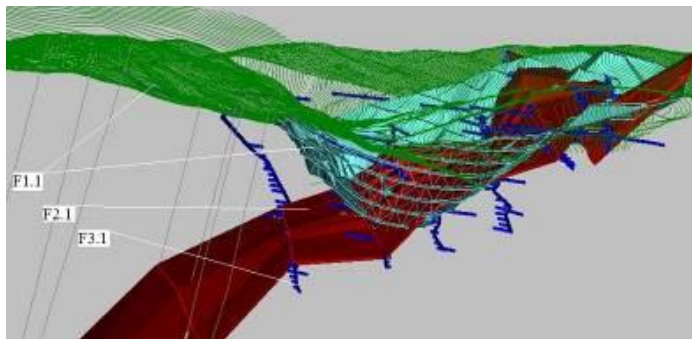


Рисунок 1. Модель золоторудного карьера

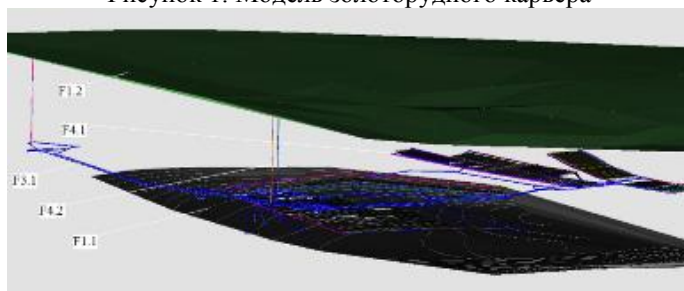


Рисунок 2. Модель угольной шахты

Выводы. Системное представление всех функций моделирования позволяет типизировать механизмы создания моделей объектов любых горнотехнических систем, выделить общие функции моделирования и на этой основе упорядочить создание моделей месторождений. Типизация моделей предопределяет возможность обобщения и систематизации функций проектирования при создании систем автоматизированного проектирования различных горнотехнических систем.

Список литературы

1. Оганесян А.Н., Агафонов В.В. Интегрированные системы и комплексы геоинформационных систем в горном деле // Горный информационно-аналитический бюллетень (ГИАБ). 2011. Т. 6. № 12. С. 623-630.
2. Назаренко В.М. Автоматизированная система моделирования, оценки запасов месторождений и управления горными работами на базе геоинформационной системы K-MINE / В.М. Назаренко, М.В. Назаренко, С.А. Хоменко // Журнал "ТИТАН". – №3. – 2009.
3. Скаженик В.Б. Система автоматизированного проектирования разработки россыпных месторождений // Горный информационно-аналитический бюллетень (научно-технический журнал). – М.: «Горная книга». – 2012. – № 3. – С. 150 – 152.

УДК 622:658.011.56

Методы создания графических баз данных в САПР

Авраимов В.С., Григорьев А.В.

Донецкий национальный технический университет

kenni909090@gmail.com

Авраимов В.С., Григорьев А.В. Методы создания графических баз данных в САПР. В данной статье был проанализированы существующие методы работы создания графических баз данных в САПР. Были выявлены преимущества и недостатки методов построения и технологий типовых графических баз данных.

Ключевые слова: графическая база данных, методы разработки ядро базы данных, пакетные файлы, САПР, AutoCAD, интерфейс.

Введение

Одним из видов баз данных являются ГБД. Они используются для обработки, хранения и автоматизированного графического построения стандартных элементов конструкций. Зачастую графическая база данных содержит всю справочную информацию об элементе (начиная с варианта установки объекта и кончая всеми необходимыми ГОСТами). Это позволяет пользователю-конструктору сэкономить свое время, не тратя его на поиск этой информации в справочниках. К тому же сокращается число возможных ошибок.

Актуальность данной работы состоит в том, что технология графических баз данных дает пользователю существенные преимущества такие как: удобность для пользователя в обращении, экономия времени, минимизирование ошибок.

При автоматизированном вычерчивании стандартных элементов с помощью ГБД реальные затраты времени пользователя-конструктора сокращаются в несколько раз. Весь процесс создания графической документации значительно ускоряется, при доле в конструкциях от 70 до 95%. Разработка ГБД является одним из главных направлений повышения эффективности САПР.

Целью данной работы является анализ технологии и методов создания графической базы данных в САПР, определение достоинств и недостатков.

1. Структура и схема функционирования типовых ГБД.

На данный момент наиболее развитыми являются ГБД, в которых изображения создаются при помощи параметрических программ. Обычно они содержат следующий функционал [1]:

- Черчение с помощью одной программы не одного типоразмера элемента, а целого размерного ряда. Это является возможным за счёт описания формы и размеров элемента в программе при помощи полного набора геометрических параметров. В отдельных пакетных файлах хранятся стандартные списки этих параметров, соответствующие каждому типоразмеру элемента.

- Создание разных вариантов изображений – нанесением штриховки, осей, с сечениями и без, в разных масштабах и т. д. Выбор конкретного варианта осуществляется либо с помощью меню или в диалоговом режиме в командной строке.

- Вставка элемента практически в любом возможном положении в любое место чертежа. Это условие обеспечивается при помощи использования в программах параметров положения, однозначно характеризующих положение элемента на чертеже. Однако осуществляется это только при условии того, что на элементе указаны оси привязки базовые точки и базовые точки. Параметры положения, такие как, точки и оси вставки указываются пользователем непосредственно при процессе построения изображения.

Использование параметрических изображений позволяет при помощи одной программы чертить полный ряд типоразмеров элемента. Прикладные ГБД обычно встроены в базовые графические системы, такие как AutoCAD и содержат ряд компонентов.

Два основных компонента:

1) набор параметрических программ, образующих необходимые изображения (обычно – в диалоговом режиме), а также 2) совокупность пакетных файлов, которые хранит всю справочную и вспомогательную информацию по вычерчиваемым элементам, в сумме составляющих ядро базы данных.

Ядро базы позволяет выполнять автоматизированное построение элементов в графической системе, вызывая соответствующие им параметрические программы из командной строки. Однако использование данного способа приносят ряд неудобств:

- а) пользователю нужно помнить названия соответствующих параметрических программ.

- б) введение имен займет много времени (по сравнению, например, с использованием мыши).

- в) при наборе имен существует возможность совершить ошибку.

- г) пользователь должен помнить ряд свойств: геометрические параметры элементов и названия типоразмеров, расположение точек и осей привязки и т.д.

Для избежания этих неудобств, в базовых системах обычно предусмотрен пользовательский интерфейс. Он упрощает создание чертежей и выполнения дополнительных действий.

1. Вызов программ из меню, которых, как правило, бывает несколько видов.

2. Наличие информационных слайдов (специальных графических файлов с изображениями элементов), которые могут:

- выполнять роль графического меню при наличии нескольких разновидностей элементов,
- пояснять смысл и назначение параметров, задающих геометрическую форму и размеры выбранных элементов, указывать положение точек и осей привязки на них.

Графическим меню называют набор изображений, который позволяет выбирать одно или несколько из них. Поэтому при встраивании ГБД в графическую систему наряду с ядром также создают пользовательский интерфейс. Сделано это для упрощения работы с базой. Такой интерфейс включает в себя:

1. модифицирование меню базовой графической системы (включение в него дополнительных пунктов, которые позволяют вызывать необходимые параметрические программы щелчком мыши)

2. создание необходимого набора информационных слайдов.

Основную часть базы данных составляют: параметрические функции, пакетные файлы и библиотека слайдов. Эти части дополняет модифицированное меню системы, в котором предусмотрен вызов параметрических программ добавляемой БД.

Схема функционирования ГБД может изменяться в зависимости от представленных к ней требований, однако, в основном, порядок действий следующий. Сначала пользователь обращается с запросом к меню системы, в которой выбирает пункт, относящийся к добавляемому типу элемента. Затем по предлагаемому ГБД графическому меню выбирается нужный вид элемента и по подсказке из информационного слайда в диалоге с помощью вспомогательных меню, а также из пакетных файлов задаются геометрические параметры вставляемого элемента. После чего пользователь указывает особенности нужного изображения и параметры его положения на чертеже. На последнем этапе параметрическая программа создаёт требуемое изображение.

2. Постановка задачи проектирования ГБД в графической системе AutoCad.

Рассмотрим на примере процесс создания и функционирования ГБД в графической системе AutoCAD на языке AutoLISP [2].

Данная база данных предназначена для черчения элементов, условно названных "четырёхугольник". Данный элемент может представлять собой прямоугольник либо равнобедренную трапецию. Она должна обеспечить следующие возможности.[3]

1. При запросе пользователя в модифицированном меню система должна предоставить ему графическое меню, при помощи которого пользователь может выбрать конкретный элемент - прямоугольник или трапецию.

2. Запрос в диалоговом режиме геометрических параметров элементов и параметров вставки. Предусмотреть альтернативный ввод стандартных геометрических параметров (из пакетных файлов) либо нестандартных – по запросам в командной строке.

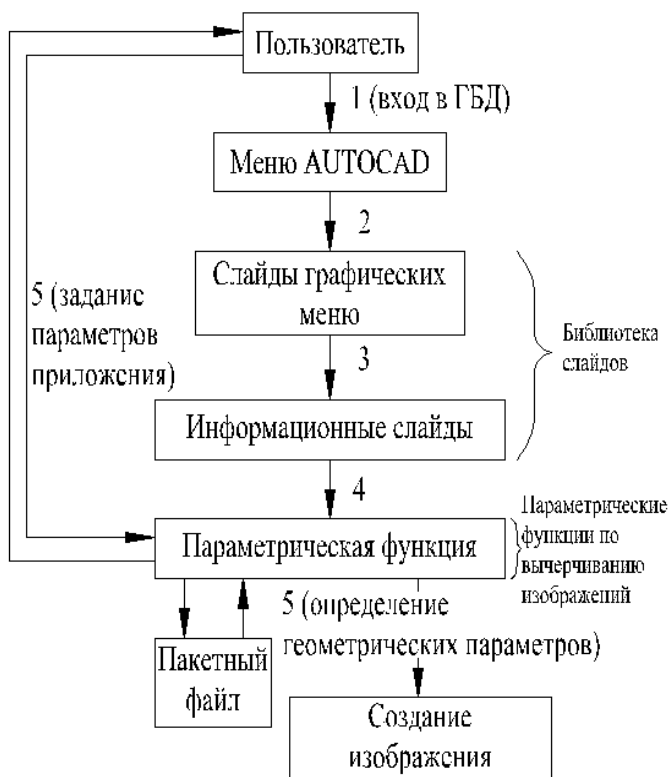


Рисунок 1 – Структура и схема функционирования ГБД, встроенной в базовую графическую систему AutoCAD.

3. Черчение элемента по заданным наборам параметров, горизонтальная и вертикальная оси должны быть показаны штриховой линией.

3.Разработка структуры ГБД.

Так как в базе должно быть предусмотрено создание двух разных изображений с соответствующими наборами стандартных параметров, то ядро её проще организовать следующим образом:[4]

1) две параметрических функции (поскольку AutoLISP – функциональный язык) – для черчение прямоугольников и по вычерчиванию трапеций.

2) два пакетных файла – содержащих стандартные типоразмеры элементов.

Ядро базы должно быть дополнено информационными слайдами, а также модифицированным меню базовой графической системы - AutoCAD.

Выводы

В ходе выполнения данной работы были проанализированы существующие методы создания графической базы данных в САПР. После их детального изучения, были выявлены присущие им преимущества и недостатки.

Также была проанализирована структура и схема функционирования графических баз данных.

К тому же был рассмотрен на примере процесс создания и функционирования ГБД в графической системе AutoCAD на языке AutoLISP.

Список литературы

1. Википедия. Схема базы данных [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/схема_базы_данных.
2. AutoCAD 2006 подробное руководство Жадаев А.Г. \ 2006 г.
3. САПР на базе AutoCAD - как это делается Зуев С. 1 января 2004 г.БХВ-Петербург.
4. Графическая база [электронный ресурс]. – Режим доступа: <http://www.ngpedia.ru/id137498p1.html>.

УДК 681.3

ОНТОЛОГИИ В САПР, АНАЛИЗ СОСТОЯНИЯ

Калинина Е.С., Григорьев А.В.

Донецкий национальный технический университет

lena00777@gmail.com, grigorievalvl@gmail.com

Аннотация

Калинина Е.С., Григорьев А.В. Онтологии в САПР, анализ состояния. В статье рассмотрены достоинства и недостатки БД и онтологий, представлена классификация онтологий и языки описания. Рассмотрен алгоритм поиска трёхмерной модели САПР на основе онтологии, выявлены основные требования необходимые для разработки онтологии в данной области.

Ключевые слова: онтология, БД, САПР, алгоритм поиска, OWL

Введение

На сегодняшний день заметно увеличился поток информации, и как следствие, возникает необходимость в новых способах её хранения, формирования, представления, обработки и систематизации. Таким образом, растёт интерес к глобальным базам знаний, которые можно использовать для практических целей. Большой интерес вызывают системы, способные самостоятельно извлекать сведения из текста. Для решения этих задач применяется онтологический подход. В чистом виде онтологии используются в качестве источника данных для многих компьютерных приложений (анализа текстов, для информационного поиска), что позволяет более эффективно обрабатывать сложную информацию.

Актуальной задачей является разработка и применение системы онтологий в САПР как инструмента представления методик проектирования. В области САПР наблюдается тенденция к упрощению процесса проектирования за счёт создания методик проектирования, их сохранения и редактирования (Грация, САПА, NIX Solutions). Подобные системы ориентированы на конкретную предметную область и предполагают создание моделей каких-то предметных областей в форме онтологий.

Современные интеллектуальные САПР реализуют общие подходы, сложившиеся в области теории искусственного интеллекта, в данном случае - в области современных онтологий, но с учётом специфики САПР. Спецификой САПР является наличие БД, структурных моделей проектирования в той или иной области, наличия БД готовых проектов и т.д.

Целью данной статьи является анализ общих требований полноценной онтологии, анализ сложившегося уровня применения онтологии в САПР и выявление тенденций дальнейшего развития онтологического подхода в САПР.

Достоинства и недостатки БД и онтологий

Прежде чем приступить к анализу и практическому применению онтологий и БД, дадим определения используемым понятиям.

База данных – это информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств. Программное обеспечение, предназначенное для работы с базами данных, называется система управления базами данных (СУБД). СУБД используются для упорядоченного хранения и обработки больших объемов информации [1].

За время применения БД нашли большое распространение в сфере моделирования реальных объектов и систем и используются в качестве хранилища данных и манипулирования этими данными. В реляционных БД вся информация представлена в виде таблиц, строки и столбцы которой называются атрибутами. Эти таблицы получили названия реляций и записи в них, как правило, не повторяются. Уникальность подобного рода записей заключается в первичном ключей который содержит набор полей определяющий запись. К основным достоинствам БД можно отнести следующее:

- в основе БД лежит математический аппарат, который позволяет основные операции над данными;
- при манипулировании и доступе к данным используются языки манипулирования данными не процедурного типа;
- манипулирование данными на уровне выходной БД и возможность динамического изменения данных;
- хранение большого объёма информации и предоставление её пользователю в простой форме [2].

Однако, несмотря на явные достоинства в БД обнаружен ряд недостатков и существенных ограничений:

- трудоёмкость разработки;
- медленный доступ к хранимой информации;
- довольно проблематично моделировать сложные связи в БД, в отличие от онтологий;
- результатом запроса БД является информация, которая хранится в самой БД. Также, часто требуется, чтобы в результате опроса был получен логический вывод на основе хранимых данных [2].

На сегодняшний день термин «онтология» достаточно популярен в обществе программистов, однако, точного определения и чёткого понимания о нём ещё нет. Впервые этот термин использовал Томас Грубер в одной из своих работ, в которой рассматривались аспекты взаимодействия человека и

интеллектуальных систем. Основная идея Грубера состояла в том, чтобы предоставить интеллектуальным системам обмениваться между собой заложенными в них знаниями. Так как сами эти знания внутри интеллектуальной системы могут быть закодированы различным образом, то для осуществления обмена знаниями с другой системой необходимо предоставить их описание. Само описание должно быть в общей степени понятным другой системе и человеку, как и язык описания. В связи с этим Грубер выделил несколько способов писания знаний:

- в канонической форме, которая представляет собой описание знаний на языке логики предикатов (например, в виде фактов языка Prolog);
- в форме онтологии, которая представляет собой множество классов, связанных между собой отношением обобщения (это обратное отношение для отношения наследования) [3].

Исходя из этого, можно сделать вывод, что онтология по Груберу представляет собой описание знаний, представленное в виде классов и иерархией между ними. Каждая интеллектуальная система, в его понимании, может предоставить сразу несколько вариантов таких описаний и предполагал, что подобного рода системы будут выступать в качестве библиотек онтологий с возможностью обмена знаниями между собой.

Однако, тот факт, что Грубер разделял спецификацию знаний (каноническую форму и онтологию) вызывало неудобство, на сегодняшний день понятие онтологии несколько изменилось. Сейчас под онтологией понимают описание декларативных знаний на формальном языке, снабжённое классификацией специфицируемых знаний, позволяющей человеку удобно воспринимать их [3].

Общая классификация онтологий

Онтологии принято классифицировать по следующим составляющим (аспектам):

- семантическая:
 - по уровню выразительности;
 - по степени формальности;
 - по уровню детальности представления.
- прагматическая:
 - по степени зависимости от конкретной задачи или прикладной области;
 - по языку представления онтологических знаний;
 - по предметной области;
 - по цели создания;
 - по наполнению (содержанию).

При этом для реализации той или иной онтологии используются специализированные языки описания онтологий:

- RDF. Язык разработан в рамках проекта Семантик-веб (Semantic Web). Основное предназначение языка – описание метаданных документов, размещаемых в Интернете;
- OWL (Web Ontology Language) – язык представления онтологий в виде последовательности аксиом, фактов и ссылок на другие онтологии;
- KIF (Knowledge Interchange Format, или формат обмена знаниями) – специальный язык, предназначенный для обмена знаниями между разными компьютерными системами;
- СусL (язык описания онтологии Сус) – это гибридный язык, в котором объединены свойства фреймов и логики предикатов [4];
- F-Logic – онтологический язык, который базируется на логиках первого порядка, однако классы и свойства в нем представлены как термины, а не как предикаты.

Для разработки онтологий существует достаточное количество редакторов онтологий, которые позволяют работать с онтологиями, реализованными при использовании различных языков и методик разработки.

К основным достоинствам применения онтологий при разработке систем можно отнести:

- возможность динамического изменения данных;
- в результате запроса исходными данными является логический вывод на основе хранимых данных;
- простое моделирование сложных связей и отношений, в отличие от БД;
- представляют естественно-языковый текст в таком виде, что он становится пригодным для автоматической обработки;
- возможность использования согласованной (разделяемой всеми) терминологии с точно определенной семантикой, позволяющего интегрировать и использовать данные (информацию) из разных источников [4].

К недостаткам онтологии можно отнести следующее:

- сложность построения онтологии (не всегда легко собрать понятия и выделить дифференциальные признаки);
- затруднения восприятия хранимых данных.

Классификация семантических онтологий

Рассмотрим подробнее классификацию семантических онтологий. По уровню выразительности такие онтологии подразделяются на тяжеловесные и легковесные.

Легковесные онтологии представляют собой простую таксономическую структуру или композиции терминов с соответствующими определениями. Они

слабо аксиоматизированы, т.е. значение термина, используемое внутри, обычно более-менее известно всем членам общества и связи между такими терминами представлены ограниченно [5].

Тяжеловесные онтологии сильно аксиоматизированы, что предоставляет возможность онтологического связывания явно. Основной целью аксиоматизации, является, прежде всего, избежание терминологической и концептуальной неоднозначности. Многие онтологии предметных областей – тяжеловесные, т.к. в их обязанность входит поддержка процессов построения сложных умозаключений. Любая тяжеловесная онтология может иметь легковесную версию [5]. Предметные онтологии, как правило, описывают словарь, заданный с определённой предметной областью, задачей или деятельностью за счёт специальных терминов. Примерами таких онтологий являются TOVE и Plinius.

Модель TOVE является многоуровневой, охватывающей концептуальный уровень, обобщённый и уровень приложений. Онтологии в системе TOVE ориентированы на представление модели корпорации, инструменты для навигации, визуализации и дедуктивных запросов. Основные цели данного проекта:

- создание такой терминологии, которая будет проста для понимания и использования;
- определение смысла каждого термина;
- ввод семантики в виде набора аксиом, которые будут автоматически выдавать ответы на поставленные вопросы о корпорации;
- определение символьных обозначений для терминов и понятий [6].

Алгоритм поиска трёхмерной модели САПР на основе онтологии

Из-за непрерывного развития MBD-технологий механизм поиска 3D-моделей теперь широко используется в CAD/CAM. На сегодняшний день многие учёные начали проводить различного рода исследования семантического поиска по 3D-модели, однако есть несколько исследований по 3D-моделям в САПР. Более того, исследования в семантическом поиске по 3D-моделям САПР могут предложить более целенаправленный подход к повторному использованию знаний о соответствующих моделях САПР. Однако исследования 3D-поиска всё ещё находятся на экспериментальном этапе.

3D CAD-модели могут быть определены на 4 уровнях:

- геометрический уровень;
- уровень структуры;
- уровень функций;
- семантический уровень.

Большинство методов поиска не в состоянии описать детальную информацию по 3D-моделям САПР на структурном, функциональном и

семантическом уровне. Они используют единичную информацию из модели и упрощают её для поиска. Далее будут описаны методы получения модели САПР на функциональном и семантическом уровнях. Модель САПР следует сегментировать в соответствующие разделы, которые будут описаны и обозначены по смысловой информации модели. На основе этого, используя семантическую информацию модели и знания пользователя, можно реализовать семантическую базу данных 3D-моделей САПР, которая полностью будет удовлетворять запросам пользователя.

В целом, для оценки сходства между двумя моделями функциональный метод опирается на характеристики геометрических форм объектов. Графический метод основывается на геометрических значениях моделей, используя график, чтобы показать, как компоненты моделей связаны между собой. Такие методы применимы к трёхмерным моделям, созданным в большинстве САПР. Наиболее распространённым методом твердотельного моделирования является метод граничного представления (B-ger) и метод конструктивной твердотельной геометрии (CSG).

B-Rep - это метод представления фигур с использованием пределов. Т.е. модель САПР может быть представлена в виде набора связанных между собой элементов. B-Rep предоставляет явные представления геометрических элементов (вершины, рёбра, грани и т.д.) и отношения между ними определяются легко. Извлекая информацию B-Rep методом из САПР-моделей, она представляется в виде атрибутов смежных графов (AAG), где узлы ссылаются на поверхности объектов и атрибуты узлов представляют геометрические свойства. Связи между узлами соответствуют связям между поверхностями модели. На рис. 1 представлено построение графа AAG на основе САПР-модели.

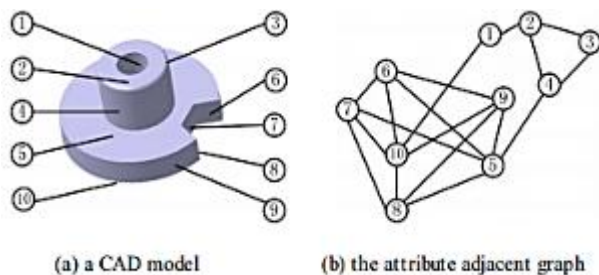


Рисунок 1 – Граф AAG на основе САПР-модели

Модель САПР должна быть сегментирована на множество значительных частей (особенности или частичные структуры) на основе перцептивной характеристики, чтобы получить характерные знаки и семантическую

информацию модели на функциональном и семантическом уровнях. Особенность формы САПР-модели состоит из двух свойств, которые отличны друг от друга: семантики и формы. Таким образом, данный алгоритм определяет сегмент САПР-модели с помощью метода распознавания в теории графов (после определения функции и библиотеки функций, в зависимости от области применения). Смежный граф можно построить на основе отношений между частями, полученными из сегментации. Он представляет отношения между составляющими частями, свойства узлов и рёбер, соответствующие информации о связях между частями модели. На рис. 2 предоставлена модель лопасти, которая была разложена на составляющие компоненты, используя алгоритм изоморфизма графов.

Для построения связи между нижним функциональным уровнем и высокоуровневой семантикой 3D САПР-модели, реализована семантическая аннотация модели, основанная на онтологии. Онтология может быть построена одним из следующих образов:

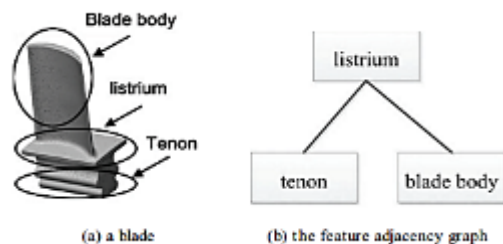


Рисунок 2 – Модель лопасти, разложенная на компоненты

- анализ различных приложений в области САПР и список основных структур. Определить иерархические отношения между структурами и сформировать концептуальное дерево онтологий;
- определить отношения между структурами (part-of, kind-of, instance-of, attribute-of). Part-of выражает наследование отношений между частями и целым. Kind-of выражает наследование между концепциями. Instance-of представляет связь между концепцией и экземпляром объекта. Attribute-of предполагает, что определённая концепция – атрибут другой концепции;
- извлечь атрибут концепции и взаимоотношений (атрибут «тело» может определять длину, ширину, точность и т.д.);
- создать экземпляры.

На рис. 3 показана классификация лопастей авиационного двигателя и их иерархические отношения.

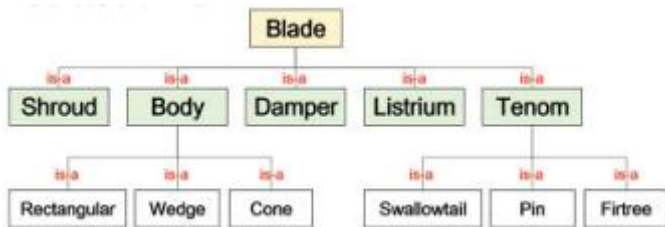


Рисунок 3 – Классификация лопастей авиационного двигателя и их иерархические отношения

Для обозначения семантической информации САПР-моделей нужно отметить семантику основной модели (имя модели, тип, приложения, материал, информация о процессе). После чего сопоставить сегментированные части из САПР-модели с семантической моделью, унаследованной из онтологии библиотек функций. В конце необходимо установить отношения между частями. На рис. 4 показано OWL представление модели.

```

<blindhole rdf:ID="blindhole_1">
  <depth rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >50.0</depth>
  <length rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >10.0</length>
  <connect rdf:resource="#cylinder_1"/>
  <radius rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >35.0</radius>
    
```

Рисунок 4 - OWL представление модели

Целью поиска по САПР-модели является возможность повторного использования модели и информации о ней. Например, существуют две различные модели САПР, которые имеют отверстия. Типы этих отверстий одинаковые, но точность измерения и материал совершенно разные. Можно сделать умозаключение, что обе модели идентичны, сравнивая форму, но производственная и термическая обработка этих деталей будет отличаться по точности размеров и материалу. Поэтому необходимо провести анализ многоточечного соответствия заданному профилю поверхностей, исходя из геометрических размеров, топологии, семантической информации и информации о процессах. Нахождение сходства двух моделей превращается в поиск наибольшего общего подграфа этих двух моделей и их свойства представляются в графе ААГ. Определение NP-полного подграфа - проблематичная задача, а уровень сложности алгоритма – высокий, т.е. для того

чтобы найти достаточно большой общий подграф нужно использовать алгоритм восхождения к вершине, который произвольно выбирает начальные координаты между узлами двух графов. Далее меняется местами отображение двух узлов, чтобы свести оценочную функцию S к минимуму (формула 1), где S - количество не совпавших краёв.

$$S_f = \frac{\min\{S_1, \dots, S_n\}}{|E_1|}, \text{ где } E_1 - \text{набор рёбер меньшего графа} \quad (1)$$

Стоит заметить, что производственный процесс является сложным и зависит от множества факторов, также нужно учитывать сходство, связанное с технологичностью изделий (формула 2).

$$S_p = \omega_m S_m + \omega_h S_h + \omega_r S_r + \omega_c S_c \quad (2)$$

где S_h обозначает сходство типов материалов двух моделей; S_r - означает подобие типа "пробелов"; S_c -сходство механической обработки требуемой точности; ω -весовые коэффициенты [7].

Вывод

В данной статье были рассмотрены основные отличия БД и онтологий, их достоинства и недостатки. Описана классификация онтологий и основные языки их описания. На сегодняшний день не существует правильного способа или методологии разработки онтологии для САПР, однако можно выделить основные требования:

- понятия в онтологии должны быть близки объектам (к физическим и логическим) и отношениям между ними;
- онтология должна содержать в себе все 4 уровня абстрагирования модели (структурную, системную, функционально-логическую и количественную);
- необходимо выявить основной набор терминов и понятий, описывающих заданную предметную область;
- основой построения онтологии по предметной области является привязка модулей реальных САПР к этапам жизненного цикла изделия посредством решения типовых задач с использованием современных интегрированных САПР (создание 3D-модели, проектирование технологического процесса).

В настоящее время проводятся исследования в области построения онтологий по предметной области «CALS-, CAD-, CAM-, CAE-технологии». Параллельно начаты работы по созданию онтологии, описывающей предметную область «Искусственный интеллект».

Список литературы

1. Система управления базами данных (СУБД), Назначение и основные функции [электронный ресурс]. – Режим доступа: <http://www.infosgs.narod.ru/31.htm>
2. Сравнительный анализ баз данных и баз знаний (онтологий) применимо к моделированию сложных процессов [электронный ресурс]. – Режим доступа: <http://web.snauka.ru/issues/2014/05/34325>
3. Онтологии в компьютерных системах [электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=33244>
4. Онтологии как системы хранения знаний [электронный ресурс]. – Режим доступа: <http://www.ict.edu.ru/ft/005706/68352e2-st08.pdf>
5. Обзор баз знаний онтологического типа [электронный ресурс]. – Режим доступа: <http://dspace.nbuu.gov.ua/bitstream/handle/123456789/8144/27-Nikonenko.pdf?sequence=1>
6. Обзор применения онтологий в моделировании и управлении [электронный ресурс]. – Режим доступа: http://bigc.ru/theory/experience/ontologies_for_modelling.php
7. Three-dimensional CAD Model Retrieval Algorithm Based on Ontology [электронный ресурс]. – Режим доступа: <http://www.sciencedirect.com/science/article/pii/S2212827116311088>

УДК 681.3

Разработка приложения при помощи совместной работы библиотеки ObjectARX и Javascript API с платформой AutoCAD на примере САПР трубопроводов

И.О. Зимонин, А.В. Григорьев

Донецкий национальный технический университет
zimoningor@gmail.com, grigorievalv1@gmail.com

Зимонин И.О., Григорьев А.В. Разработка приложения при помощи совместной работы библиотеки ObjectARX и Javascript API с платформой AutoCAD на примере САПР трубопроводов. В тексте данной статьи рассматривается принцип создания приложений на платформе AutoCAD на примере САПР трубопроводов. В статье описан алгоритм настройки данной архитектуры, её достоинства, недостатки и приведены примеры рабочего кода пошаговой разработки с иллюстрированием процесса.

***Ключевые слова:** Плагин, AutoCAD, ObjectARX, JavaScript API, C#, .NET, Javascript, трубопровод.*

Введение

На данный момент в сфере информационных технологий наблюдается тенденция стремительного развития Web-технологий. Вследствие этого, разработчики систем автоматизированного проектирования и платформ для разработки САПР вводят поддержку взаимодействия их приложений с этими технологиями. Таким образом, в силу удобства использования и простоты кода, в платформу AutoCAD[1] внедрили поддержку JavaScript API[2] (клиентского скрипт-языка, с возможностью посылать запросы на сервер). Это расширило спектр возможностей платформы AutoCAD как в сторону интерфейсных, так и в сторону функциональных решений.

С другой же стороны, уже много лет специалисты AutoCAD развивают собственную библиотеку для работы с различными объектами обширной базы данных. Называется эта библиотека ObjectARX[3].

Для создания так называемых приложений (плагинов) для AutoCAD было решено использовать JavaScript API в качестве интерфейсного решения (в том числе для возможности последующей связи с сервером), и библиотеку ObjectARX как мощный и надежный инструмент работы со средой AutoCAD.

В данной статье будет рассмотрено взаимодействие JavaScript API, ObjectARX и AutoCAD на примере преобразования макета из линий в простейший трубопровод.

1 Настройка ObjectARX и JavaScript API

Библиотека ObjectARX подходит для использования во многих объектно-ориентированных языках. Однако для наших целей был выбран C# на платформе .Net, как самый удобный с точки зрения взаимодействия и синтаксиса язык.

Средой разработки является Microsoft Visual Studio 2013 (далее MVS). Версия AutoCAD в свою очередь 2016.

Итак, прежде всего, необходимо создать проект, в котором будет собираться динамическая библиотека для AutoCAD.

Для этого в MVS выбираем File->New->Project->Visual C#-> Class Library

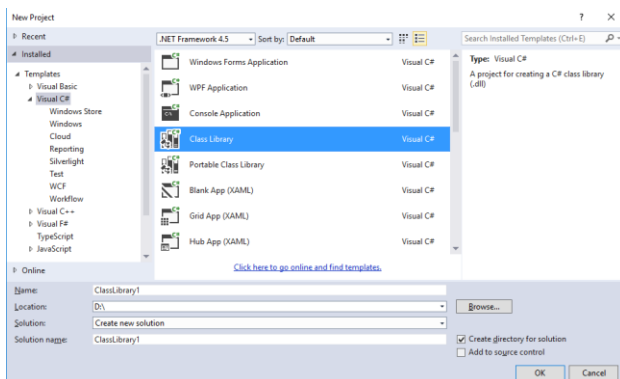


Рисунок 1 – Создание проекта для ObjectArx

После создания проекта, необходимо добавить в него ссылки на нужные файлы библиотеки ObjectARX, а также установить Newtonsoft JSON.Net.

Для добавления ссылок необходимо нажать правой кнопкой мыши по References в дереве проектов и выбрать Add Reference (см. рис), затем на вкладке Browse указать путь к таким dll AutoCad, как: acmgd.dll, acdbmngd.dll, accoremgd.dll

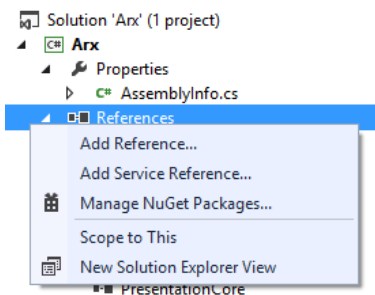


Рисунок 2 – Добавление ссылки

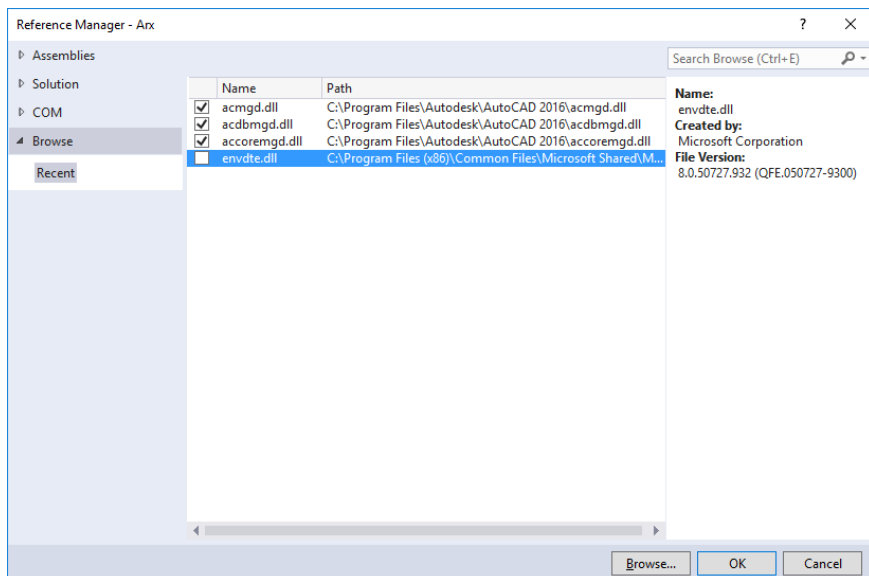


Рисунок 3 – Необходимые dll зависимости

Для установки Newtonsoft JSON .NET нужно вызвать контекстное меню на проекте и выбрать ManageNuGetPackages и на вкладке Online через поиск найти и установить данный модуль.

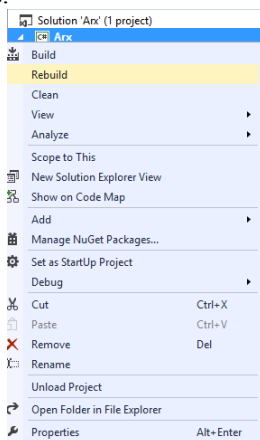


Рисунок 4 – NuGet Packages

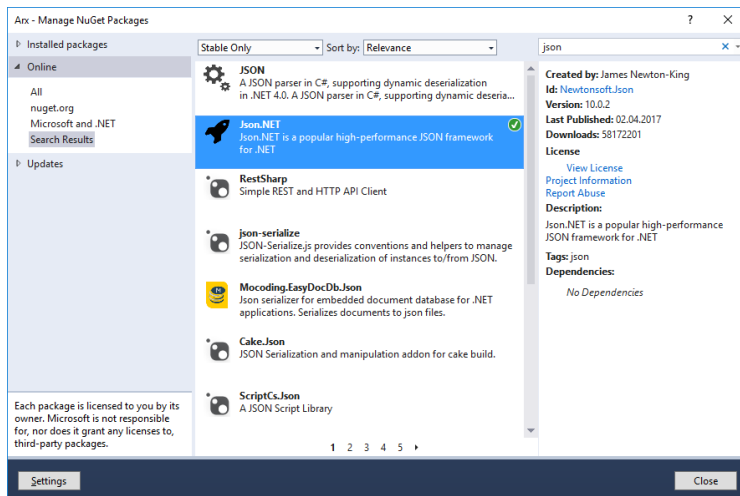


Рисунок 5 – Установка JSON.Net

На этом настройка зависимостей проекта заканчивается, о коде речь пойдёт немного дальше, а пока переходим к настройке Javascript API.

JS API настроить намного проще, чем ObjectARX. Для подключения возможностей этой библиотеки, достаточно создать html файл в удобном для пользователя месте, подключить в нем JS файл библиотеки для AutoCAD (см. рис. 6-7), а после этого все необходимые JS и css файлы.

```
<head>
<script type="text/javascript" src="http://www.autocadws.com/jsapi/v1/Autodesk.AutoCAD.js"
></script>

<link rel = "stylesheet" href = "main.css">

<script
src="https://code.jquery.com/jquery-2.2.4.min.js"
integrity="sha256-BbhdvlQF/xTY9gja0Dq3HivQF8LaCRTxZKRutelT44="
crossorigin="anonymous"></script>
<script type="text/javascript" src="ArxJs.js"></script>
```

Рисунок 6 – Подключение JavaScript API и других файлов к HTML

```
<div class = "MarginBlock">
<label>Radius:</label>
<input type = "text" id = "PipeRad" value = "10">
</div>

<input type='button' class = "Button wFill" onclick='CreatePipeline()' value='Create Pipeline' />
<br />
```

Рисунок 7 – Код кнопки и текстового поля

2 Настройка взаимодействия компонентов системы

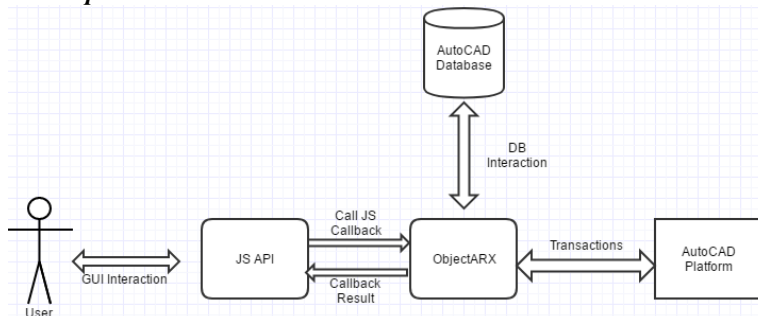


Рисунок 8 – Схема взаимодействия

Как видно на рис. 8, взаимодействие происходит следующим образом. Пользователь взаимодействует с графическим интерфейсом зашифрованной HTML странички (JS API). При срабатывании скриптов в ответ на действия пользователя вызываются JavaScript Callback функции из скомпилированной dll проекта с ObjectARX. Далее они взаимодействуют с базой данных для произведения операций над существующими сущностями, либо производят добавление новых в БД. Затем при помощи механизма транзакций изменения уходят на сцену AutoCAD, где их и видит пользователь.

Подключение ObjectARX модуля происходит при помощи команды `_NETLOAD` и выбора скомпилированной DLL проекта ObjectARX (см. рис. 9).

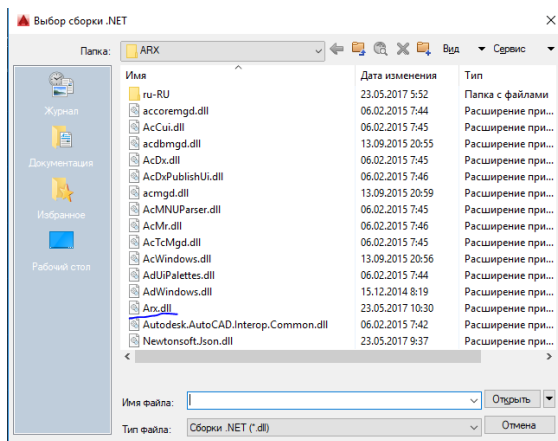


Рисунок 9 – Подключение ObjectARX модуля

Для удобной загрузки HTML странице в ObjectARX проекте необходимо объявить функцию LoadHTML (см. рис. 10), и разрешить доступ к ней из GUI AutoCAD посредством ключевого слова CommandMethod.

```
[CommandMethod("LoadHtml")]
public void LoadHtml()
{
    string filename = SelectFile(
        "Select Html File to load...",
        "Html Files (*.html)|*.html");

    if (filename == string.Empty)
        return;

    if (_ps == null)
    {
        _ps = new Autodesk.AutoCAD.Windows.PaletteSet(
            "JavaScript Demo",
            new Guid("730CF323-7D71-40A1-990E-F7CF81A84340"));
    }

    String url = "file://" + filename;

    try
    {
        String tabName = "";
        Uri uri = new Uri(url);
        if (uri.IsFile)
        {
            String[] segments = uri.Segments;

            if (segments.Length > 0)
            {
                tabName = segments[segments.Length - 1];

                String[] fileSplit = tabName.Split('.');

                if (fileSplit.Length > 0)
                    tabName = fileSplit[0];
            }
        }
        else
        {
            tabName = uri.Host;
        }

        if (_ps.Count != 0)
        {
            _ps[0].PaletteSet.Remove(0);
        }
    }
}
```

```

    Palette p = _ps.Add(tabName, uri);
    _ps.Visible = true;
}
catch (UriFormatException ex)
{
    System.Windows.Forms.MessageBox.Show(ex.Message);
}

```

Рисунок 10 – Код подключения HTML файла в палитру AutoCAD

После загрузки это DLL в AutoCAD пользователю становится доступна команда LoadHTML, и предлагается выбрать HTML файл (см. рис. 11). При подтверждении выбора HTML файл автоматически заносится в палитру AutoCAD и отображается на экране готовый к использованию (см. рис. 12).

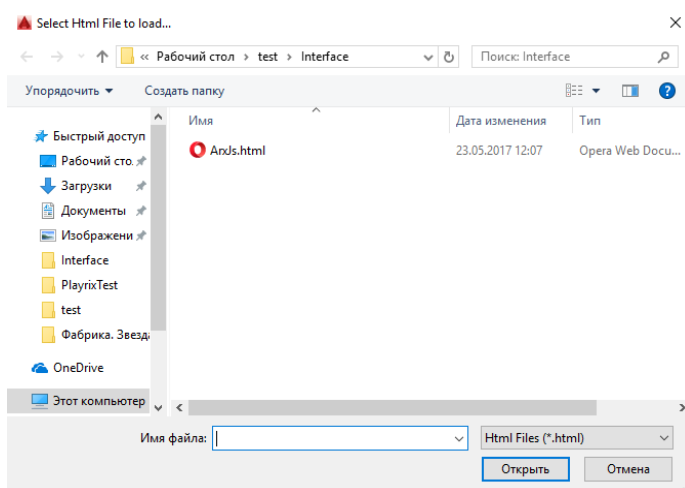


Рисунок 11 – Подключение HTML файла

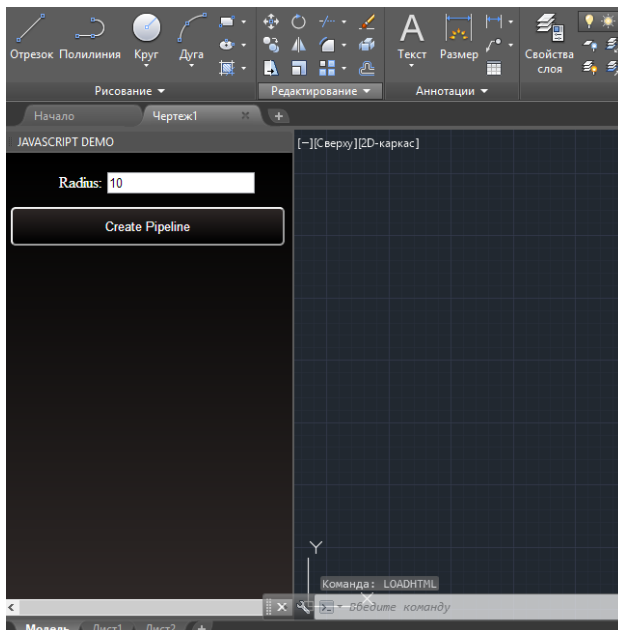


Рисунок 12 – Загруженный HTML файл

Для вызова пользовательской функции ObjectARX используется технология Javascript Callback. Суть такова, что посылается запрос на асинхронное выполнение (см. рис. 13), который приходит в соответствующую функцию ObjectARX по имени (в данном случае – “CreatePipelineJs”), выполняется и возвращает строку результата (см. рис. 14).

```

function OnArxSuccess(result) {
    write(result);
    write("\n");
}

function OnArxError(result) {
    write("\nOnParseError: " + result);
}

function CreatePipeline() {
    execAsync(JSON.stringify({
        functionName: 'CreatePipelineJs',
        invokeAsCommand: false,
        functionParams: { radius: $("#PipeRad").val() }
    )),
    OnArxSuccess,
    OnArxError);
}

```

Рисунок 13 – Вызов JavaScript Callback'а в ObjectARX

```

[JavaScriptCallback("CreatePipelineJs")]
    public string CreatePipelineJs(string jsonString)
    {
        JObject obj = JObject.Parse(jsonString);
        CreatePipeline(Single.Parse(obj.GetValue("radius").ToString()));
        return "Converted";
    }

```

Рисунок 14 – JavaScript Callback в ObjectARX

Сам алгоритм преобразования контура в трубы в данной статье предельно прост (см. рис. 15). Происходит проход по контуру (см. рис. 16) с созданием цилиндров заданного радиуса и соответствующих элементам контуров длины с последующим их позиционированием (см. рис 17). Между цилиндрическими элементами располагаются сферы (в дальнейшем это расширится до соединительных объектов БД, в данной статье используется только как пример взаимодействия с БД).

```

[CommandMethod("CreatePipeline", CommandFlags.UsePickSet |
    CommandFlags.Redraw |

```

```
CommandFlags.Modal)
]
public void CreatePipeline(float pRadius)
{
    Document doc =
Autodesk.AutoCAD.ApplicationServices.Application.DocumentManager.M
diActiveDocument;
    Editor ed = doc.Editor;
    Database db = doc.Database;
    Matrix3d curUCSMatrix =
doc.Editor.CurrentUserCoordinateSystem;
    CoordinateSystem3d curUCS = curUCSMatrix.CoordinateSystem3d;

    try
    {
        PromptSelectionResult selectionRes =
ed.SelectImplied();
        if (selectionRes.Status == PromptStatus.Error)
        {
            PromptSelectionOptions selectionOpts =
                new PromptSelectionOptions();
            selectionOpts.MessageForAdding =
                "\nSelect objects to list: ";
            selectionRes = ed.GetSelection(selectionOpts);
        }
        else
        {
            ed.SetImpliedSelection(new ObjectId[0]);
        }
        if (selectionRes.Status == PromptStatus.OK)
        {
            Transaction tr =
                doc.TransactionManager.StartTransaction();
            try
            {
                ObjectId[] objIds = selectionRes.Value.GetObjectIds();
                foreach (ObjectId objId in objIds)
                {
                    Entity obj = (Entity)tr.GetObject(objId,
OpenMode.ForRead);
```

```

Autodesk.AutoCAD.DatabaseServices.Line line = obj as
Autodesk.AutoCAD.DatabaseServices.Line;

    if (line == null) {
        obj.Dispose();
        continue;
    }

Solid3d sol = new Solid3d();
sol.CreateFrustum(line.Length, pRadius, pRadius,
pRadius);
doc.LockDocument();

BlockTable bt =
    (BlockTable)tr.GetObject(
        db.BlockTableId,
        OpenMode.ForRead
    );

BlockTableRecord ms =
    (BlockTableRecord)tr.GetObject(
        bt[BlockTableRecord.ModelSpace],
        OpenMode.ForWrite
    );

ms.AppendEntity(sol);
tr.AddNewlyCreatedDBObject(sol, true);

sol.TransformBy(Matrix3d.Displacement(line.StartPoint -
Point3d.Origin + new Vector3d(0, 0, 0.5f * line.Length));
sol.TransformBy(Matrix3d.Rotation(Math.PI * 0.5f,
curUCS.Yaxis, line.StartPoint));
sol.TransformBy(Matrix3d.Rotation(
    Math.Atan2(Cross2D(new Vector3d(1, 0, 0),
(line.EndPoint - line.StartPoint)), new Vector3d(1, 0,
0)).DotProduct(line.EndPoint - line.StartPoint)),
curUCS.Zaxis, line.StartPoint));

Solid3d sphere = new Solid3d();
sphere.CreateSphere(pRadius);

ms.AppendEntity(sphere);

```



```

        tr.AddNewlyCreatedDBObject(sphere, true);

sphere.TransformBy(Matrix3d.Displacement(line.StartPoint -
Point3d.Origin));

        sphere = new Solid3d();
        sphere.CreateSphere(pRadius);

        ms.AppendEntity(sphere);
        tr.AddNewlyCreatedDBObject(sphere, true);

sphere.TransformBy(Matrix3d.Displacement(line.EndPoint -
Point3d.Origin));

        line.Dispose();
    }
    tr.Commit();
    }
    catch (Autodesk.AutoCAD.Runtime.Exception ex)
    {
        ed.WriteMessage(ex.Message);
        tr.Abort();
    }
}
}
catch (Autodesk.AutoCAD.Runtime.Exception ex)
{
    ed.WriteMessage(ex.Message);
}
}
}

```

Рисунок 15 – Простейший алгоритм преобразования контура в трубопровод.

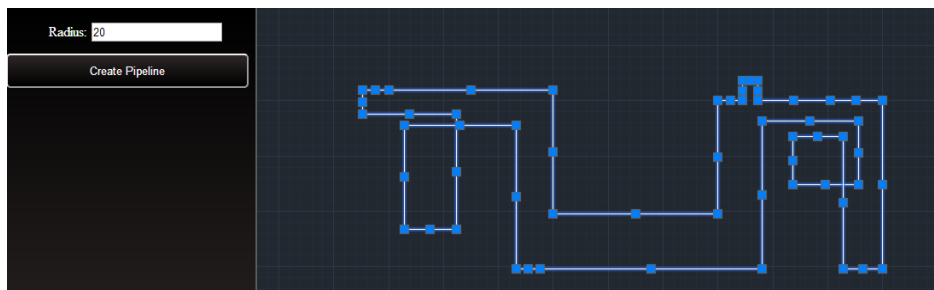


Рисунок 16 – Выделенный контур

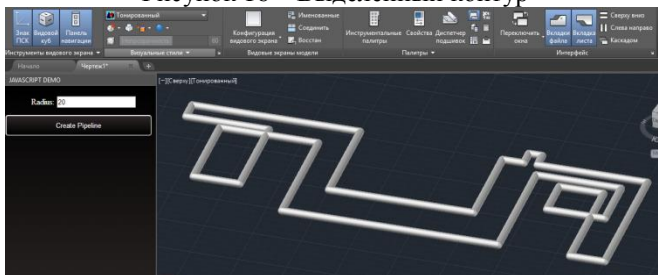


Рисунок 17 –Результат работы алгоритма

Преимущества и недостатки данной концепции

Преимуществом данной концепции является скорость, так как ObjectARX – «родная» библиотека AutoCAD, гораздо большее удобство при работе с объектами базы данных, чем при использовании AutoLISP, поддержка Web-технологий, также поддержка известных JS библиотек (например, Three.js [4]).

Из недостатков можно выделить начальную сложность освоения данной архитектуры, по причине не большого количества информации о возможности совмещения данных библиотеки и решений воедино [5].

Выводы. В ходе данной статьи были продемонстрированы возможности взаимодействия Web-технологий (JavaScript API) с библиотекой ObjectARX на платформе AutoCAD, что понижает порог вхождения Web-разработчиков в использование возможностей AutoCAD благодаря упрощению создания интерфейса и связи его с программным кодом библиотеки. Также данный подход открывает возможности связи с сервером и управления платформой с помощью приходящих инструкций. Также был рассмотрен примитивный алгоритм преобразования части чертежа-макета в объемные объекты типа «трубы». Данную архитектуру достаточно легко усовершенствовать и развивать, поэтому использование данной концепции полностью оправдано.

Список литературы

1. Технологии AutoCAD [электронный ресурс]. – Режим доступа: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=1911627>
2. Javascript API [электронный ресурс]. – Режим доступа: <https://df-prod.autocad360.com/jsapi/v3/GettingStart/index.html>
3. ObjectARX [электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/ObjectARX>
4. 3D solids in an HTML AutoCAD UI using Three.js [электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=J0CCF2j0crg>
5. Kean Walmsley Blog[электронный ресурс]. – Режим доступа: http://through-the-interface.typepad.com/through_the_interface/about-the-author.html

Секция № 4
Информационные
управляющие системы и
технологии



УДК 004.89

ПОДСИСТЕМА ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ТЕХНИКО-ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ

Бизянов Е.Е., Гутник А.А.

ГОУ ВПО ЛНР «Донбасский государственный технический университет», г. Алчевск

кафедра специализированных компьютерных систем

E-mail: hutnik.aa@gmail.com

Аннотация

Бизянов Е.Е., Гутник А.А. Подсистема интеллектуального анализа технико-экономической информации. В работе предложена структура подсистемы интеллектуального анализа технико-экономической информации. Выявлены проблемы, влияющие на качество работы подсистемы. Предложены методы и модели, необходимые для проведения интеллектуального анализа.

Ключевые слова: методы интеллектуального анализа, технико-экономическая информация, информационная система.

Постановка проблемы. Эффективное управление промышленным предприятием невозможно без полноценной информационной поддержки руководителей, которые при принятии управленческих решений используют результаты финансового анализа, анализа хозяйственной деятельности, отчеты цехов и другую информацию.

Анализ технико-экономической информации позволяет изучить взаимодействие производственных и экономических процессов на предприятии и установить их влияние на результаты деятельности последнего [1]. Для проведения анализа используют технико-экономическую информацию, хранящуюся и циркулирующую в информационной системе предприятия, а также поступающую извне. При оценке технико-экономической информации целесообразно использовать методы интеллектуального анализа, позволяющие совместить обработку данных, представленных в числовой, лингвистической и других формах.

Результаты исследования. Для проведения технико-экономического анализа на предприятии будем использовать следующие источники первичных данных: планы предприятия, отчетность предприятия, техническую и технологическую документацию, информацию с машинных носителей, результаты маркетинговых исследований, результаты анализа внешних факторов, результаты опросов работников и экспертов, другие источники данных.

В [2-5] для проведения интеллектуального анализа данных предлагается использовать следующие методы и модели:

- методы классификации данных (иерархические деревья, метод ближайших соседей);
- прогнозирующие методы;
- методы фильтрации данных (метод главных компонент, метод независимых компонент, ядровый метод главных компонент);
- искусственные нейронные сети;
- нечеткую логику и нечеткую математику.

Не каждый из перечисленных методов и моделей позволяет оперировать одновременно с данными, представленными в количественной и качественной форме. Поэтому имеет смысл использовать известные методы и модели совместно, производя их выбор и настройку в зависимости от исходных данных. Предлагаемая подсистема интеллектуального анализа технико-экономической информации (рис. 1) встраивается в существующую информационную систему предприятия.

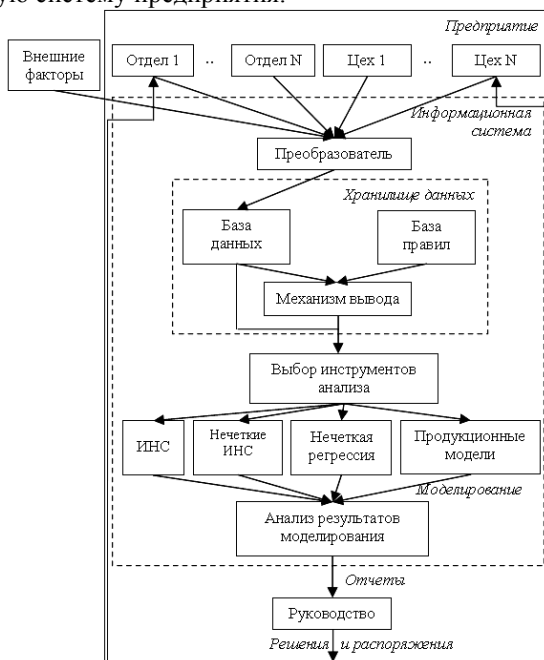


Рис. 1. Подсистема интеллектуального анализа технико-экономической информации

Исходные данные от подразделений сначала поступают в модуль преобразования данных, который конвертирует качественные показатели в

количественные и производит операцию фаззификации (при необходимости). После первичной обработки данные заносятся в базу данных, а затем поступают в модуль выбора инструментов анализа, в котором принимается решение об использовании наиболее подходящего метода анализа.

Выбор инструмента зависит от типа данных, их количества и настроек подсистемы, заданных пользователем. После этого происходит процесс настройки инструментов анализа (подбор коэффициентов, обучение и т.д.) и моделирование. Результаты моделирования поступают к руководителям различных уровней в форме отчетов: графиков, таблиц или диаграмм, с возможностью фильтрации отображаемых параметров. Руководство на основании отчетов принимает управленческие решения, которые затем в форме распоряжений поступают в отделы и цехи предприятия.

При реализации разрабатываемой подсистемы необходимо решить проблемы: недостатка данных, выбора инструментов анализа; а также учесть наличие запаздываний, соответствующих разности во времени между поступлением информации в подсистему, ее обработкой и анализом, причем величина запаздывания может варьироваться. Для уменьшения запаздывания можно увеличить частоту сбора данных, например, добавив контрольные моменты сбора информации в интервалах между моментами, заданными бизнес-правилами предприятия. Проблему недостатка данных можно решить, используя данные из предыдущего анализа, или спрогнозировав динамику изменения параметров на основе имеющихся данных.

Вывод. Проведенные исследования позволили выяснить, что для анализа технико-экономической информации необходимо использовать методы интеллектуального анализа данных. Установлено, что подсистема интеллектуального анализа должна содержать хранилище данных, модуль выбора инструментов анализа, модуль анализа результатов моделирования. Были выявлены проблемы, которые могут возникнуть на этапе реализации предлагаемой подсистемы, и предложены варианты их решения.

Список литературы

1. Савицкая Г.В. Анализ хозяйственной деятельности предприятия / Г.В. Савицкая. – Минск: ООО "Новое знание", 2001. – 668 с.
2. Рыбина Г.В. Основы построения интеллектуальных систем / Г.В. Рыбина. – М.: "Финансы и статистика", 2010. – 432 с.
3. Методы и модели анализа данных: OLAP и Data Mining / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. – СПб.: БХВ-Петербург, 2004. – 336 с.: ил.
4. Базы данных. Интеллектуальная обработка информации / В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх. – М.: Нолидж, 2000. – 352 с.
5. Силов В.Б. Принятие стратегических решений в нечеткой обстановке / Силов В.Б. – М.: ИНПРО-РЕС, 1995. – 228 с.

УДК 004.65:004.75

АРХИТЕКТУРА ИНТЕГРАЦИИ РАЗНОРОДНЫХ ИСТОЧНИКОВ ДАНЫХ

Бизянов Е.Е., Кононенко Н.Н

ГОУ ВПО ЛНР «Донбасский государственный технический
университет», г. Алчевск
кафедра "Специализированные компьютерные системы"
E-mail: nikolay.kononenko.1986@gmail.com

Аннотация

Бизянов Е.Е., Кононенко Н.Н. Архитектура интеграции разнородных источников данных. В докладе рассмотрены основные подходы интеграции разнородных источников данных, выявлены проблемы, их использования. Предложена архитектура интеграции, разнородных источников данных.

Ключевые слова: архитектура, интеграция, источники данных.

Общая постановка проблемы. Нарастание объемов информации, хранящейся в базах данных (БД) информационных систем, с одной стороны, способствует повышению эффективности принимаемых управленческих решений, однако, с другой стороны, вызывает проблемы, связанные с развитием и обслуживанием хранилищ данных. На действующих промышленных предприятиях в течение длительного времени создавались локальные БД для различных предметных областей и подразделений, а также формировались центры актуализации информации. Проблема использования указанных БД в единой информационной системе предприятия состоит в том, что центры актуализации, как правило, разнесены территориально, а системы управления базами данных (СУБД) используют различные технологии [1].

Перенос существующих БД в среду более современных СУБД связан со значительными затратами ресурсов, а также может привести к потере и искажению данных, хранящихся в БД (нарушению целостности данных). Интеграция информации представляет собой процесс, при котором данные из нескольких БД или других информационных источников объединяются вместе и представляются, для пользователей, как одна БД [2 с. 1037].

Разработка методов и моделей для интеграции разнородных БД осуществляется исследователями постоянно, что говорит об актуальности проблемы [3–5]. Так, в [4] проблему интеграции разнородных БД предложено решать в контексте разнородности структуры данных в пределах одной платформы СУБД. Ведущие разработчики коммерческих СУБД также постоянно совершенствуют свои программные продукты с целью объединения разнородных по платформам баз данных [5, 6].

Цель исследования – разработка архитектуры интеграции разнородных источников данных на промышленном предприятии.

Проведенные исследования. На сегодняшний день известно три наиболее распространенных способа объединения нескольких разнородных БД в единое информационное пространство [2, с. 1041]: федеративные базы данных (federated databases), хранилища данных (data warehouses) и медиаторы (mediators).

При использовании федеративных баз данных каждая из БД для получения информации из другой БД должна использовать заранее разработанные интерфейсные средства взаимодействия, обеспечивающие трансляцию запросов [2 с. 1042].

Преимуществом данного подхода является его простота при небольшом количестве узлов в федерации [2 с. 1042], а основным недостатком – экспоненциальный рост количества интерфейсных компонент при увеличении числа узлов в федерации [2 с. 1042].

При использовании хранилищ данных необходимо выделение дополнительного централизованного хранилища данных, выполняющего сбор данных и их предварительную обработку из разнородных источников [2 с. 1043].

Основной проблемой хранилищ данных является проблема сбора информации в хранилище, для обеспечения ее актуальности [2]. Существуют следующие варианты обновления информации в хранилище [2 с. 1043-1044]:

- все данные хранилища периодически обновляются (полная перестройка хранилища);
- данные, измененные с момента последнего обновления, периодически обновляются (сложный алгоритм инкрементального обновления данных);
- данные в хранилище обновляются в ответ на операции изменения данных в источниках данных (большая интенсивность передачи данных).

Поток информации является однонаправленным от интегрируемых БД к хранилищу, что запрещает обновление данных, находящихся в нем.

Подход с применением медиаторов представлен в статье [7]. Медиатор не является непосредственным хранилищем данных, в его задачи входит преобразование запросов, полученных от клиентов в соответствующие запросы, адресованные источникам данных [2 с. 1046]. Каждый из полученных запросов медиатор перенаправляет оболочке, отвечающей за преобразование и отправку запроса источнику данных. Помимо преобразования запросов и трансляции данных, медиаторы также могут выполнять оптимизацию запросов с учетом возможностей источника [2, с. 1056].

Проанализировав методы интеграции информации, рассмотренные выше, и проблемы, возникающие при их реализации – предлагается использовать гибридный метод интеграции разнородных источников данных, который основан на схеме интеграции медиатора, с добавлением подсистем,

отвечающих за кэширование данных и результатов выполнения запросов, координацию распределенных транзакций.

Предлагаемая архитектура интеграции разнородных источников данных приведена на рис. 1.

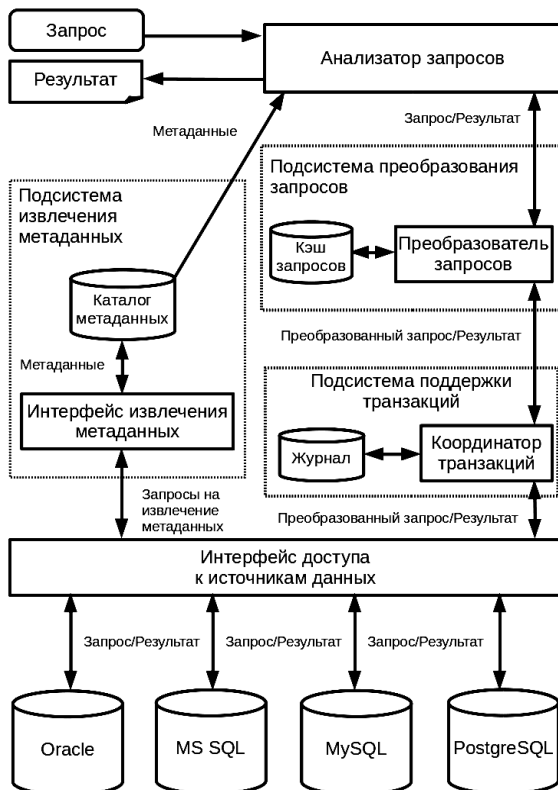


Рис 1. Архитектура интеграции разнородных источников данных.

В состав предлагаемой архитектуры входят следующие компоненты:

- анализатор запросов – производит проверку синтаксиса поступающих запросов на основе данных, имеющихся в хранилище метаданных;
- преобразователь запросов – отвечает за разделение поступившего запроса на составляющие, их преобразование в соответствии с синтаксисом, поддерживаемым источником данных, перенаправление запросов к источникам данных, объединение и конвертирование результатов выполнения составляющих запросов, кэширование результатов выполнения запросов;

– координатор транзакций – организует поддержку ведения транзакций, на основе средств, предоставляемых интерфейсом доступа к данным, если поддержка транзакций отсутствует – организует поддержку транзакций своими средствами, обеспечивает ведение журнала транзакций;

– интерфейс доступа к источникам данных – обеспечивает единый программный доступ к разнородным источникам данных. В качестве поставщиков для доступа к разнородным источникам могут быть использованы: ADO, ODBC, OLE DB, JDBC и др.

Выводы. Проведенный анализ существующих методов интеграции разнородных БД позволил предложить архитектуру интеграции разнородных источников данных на основе медиатора, с добавлением средств оптимизации выполнения запросов, что позволит упростить использование существующих разнородных БД в информационной системе промышленного предприятия, и предоставит возможность применять их в едином информационном пространстве предприятия.

Список литературы

1. Современная концепция построения информационной системы управления крупным металлургическим предприятием / Н.Н. Заблодский, Е.Е. Бизянов, И.С. Зайцев // Сборник научных трудов Донбасского государственного технического университета. Вып.27 . 2008. 490с. : ил. с. 186-191.
2. Garcia-Molina, H. Database systems the complete book. Second Edition / H. Garcia-Molina, J. D. Ullman, J. Widom. – Department of Computer Science Stanford University, 2009. - 1240 p.
3. Дейт, К. Дж. Введение в системы баз данных, 8-е издание. / К. Дж Дейт. – М.: Издательский дом "Вильямс", 2005. - 1328 с.
4. Плеханов, С.В. Интеграция разнородных баз данных на основе многомерных моделей данных: на примере интеграции геоинформационной системы с информационными системами предприятия : дис. ... канд. тех. наук : 05.13.11 : защищена 29.12.2006 / Плеханов Сергей Владимирович. Уфа 2006. – 199 с.
5. Луни, К. Oracle Database 10g Настольная книга администратора / К. Луни, Б. Брилла – М: Лори, 2008. – 729 с.
6. Виейра, Р. Программирование баз данных Microsoft SQL Server 2005 для профессионалов / Р. Виейра. – М.: ООО "И.Д. Вильямс", 2008. – 1072 с.: ил.
7. Wiederhold G. Mediators in the Architecture of Future Information Systems / G. Wiederhold // IEEE Computer, 1992. - V. 5, № 3. - Pp. 38 - 49.

УДК 004

ВЫБОР СИСТЕМЫ АВТОМАТИЗАЦИИ

Чередникова О.Ю., Капралова А.Ю.

Донецкий национальный технический университет

кафедра компьютерной инженерии

E-mail: westy35@mail.ru

Аннотация

Чередникова О.Ю., Капралова А.Ю. Выбор системы автоматизации. Рассмотрены методики MRP-, MRPII- и ERP- систем. Выделены основные параметры выбора системы автоматизации. Определены затраты и эффективность внедрения.

Введение

Сегодня с полной уверенностью можно утверждать, что во все времена эпохального шестивия производства, ИТ выступали тем двигателем, без которого предприятия не достигли бы высоких результатов. В условиях жесткой конкуренции, динамичного рынка даже самые небогатые предприятия не могут позволить себе отказаться от столь мощного и эффективного средства развития, как автоматизация.

Говоря о современном предприятии как объекте автоматизации, условно можно выделить две группы его задач: производственные и административные. Первая группа относится строго к производственной деятельности предприятия, а вторая - к административно-хозяйственной. Системы, решающие задачи управления производством называются автоматизированными системами управления технологическими процессами (АСУТП), а системы для управления административно-хозяйственной деятельностью - автоматизированными системами управления предприятием (АСУП). АСУП имеют более широкое распространение, поскольку потенциально могут применяться на предприятиях любого назначения. Значит, имеет смысл выяснить, по каким критериям следует выбирать АСУП.

MRP- и ERP- системы

MRP-, MRPII- и ERP- системы – основные программные инструменты современного планирования и принятия грамотных управленческих решений на предприятии.

Методика MRP (Manufacturing Resource Planning – планирование производственных ресурсов) декларирует, какие процессы учета и управления должны быть реализованы на предприятии, в какой последовательности они должны выполняться, и содержит рекомендации о том, как они должны выполняться.

MRPII представляет собой методологию, направленную на эффективное управление всеми ресурсами производственного предприятия (набор принципов, моделей и процедур управления и контроля, выполнение которых

должно способствовать улучшению показателей экономической деятельности предприятия).

ERP (Enterprise Resource Planning – планирование ресурсов предприятия) является дальнейшим развитием систем MRPII. Системы этого класса ориентированы на работу с финансовой информацией для решения задач управления большими корпорациями с разнесенными территориально ресурсами. Сюда включается все, что необходимо для получения ресурсов, изготовления продукции, ее транспортировки и расчетов по заказам клиентов[2].

Выбор АСУП

При автоматизации бизнес-процессов должен обеспечиваться полный охват всех циклов управления предприятием. АСУП должна иметь современную архитектуру, максимально использовать все возможности мощных систем управления базами данных (СУБД), поддерживающих язык структурированных запросов SQL и стандарт ODBC, иметь открытый справочник данных, открытую бизнес-логику.

После автоматизации предприятия каждое его подразделение становится центром финансового учета с соответствующим уровнем финансовой ответственности его руководителя. Существенное сокращение бумажной рутины ускоряет процесс и повышает качество обработки заказов, поднимает конкурентоспособность и рентабельность продукции предприятия в целом.

На мировом рынке предлагается сейчас свыше 500 систем класса MRP II - ERP. Рынок бурно растет - на 35% - 40% каждый год. В настоящее время в России присутствуют около десятка западных систем и три-четыре отечественных. При выборе системы управления необходимо обратить внимание на следующие пункты:

- Распространенность (количество успешных внедрений);
- Наличие русифицированной документации;
- Качество локализации (если система западная);
- Функциональная полнота;
- Гибкость (наличие возможности менять модули, меню, формирование отчетов и справки, менять бизнес - процессы и алгоритмы путем параметрической настройки и т. Д.);
- Архитектура. Желательна трехзвенная - сервер базы данных, сервер приложений, клиент - клиент-серверная;
 - СУБД. Обязательно должен входить Oracle, желательны SQL Server и MS SQL.

Таблица 1 – Сравнение наиболее популярных АСУП в России

АСУП	Распространенность (% от российского рынка ERP-систем)	Наличие русифицированной документации	Качество локализации	Функциональная полнота	Гибкость	Архитектура	СУБД	Стоимость внедрения (на одно рабочее место)
SAP R/3	40%	+	Удовлетворительное	+	+	2-х или 3-х звенная Возможность запуска нескольких серверов приложений	MS SQL, Oracle	около \$7000
Галактика	4%	+	Отличное	+	+	клиент-сервер	MS SQL, Oracle	\$350-1200
1С: Предприятие 8.0. УПП	22%	+	Отличное	+	+	Клиент-сервер и файл-сервер	Pervasive SQL, MS SQL, Oracle	\$150-600

Как можно заметить из таблицы 1, наиболее приемлемым вариантом в отношении цена/качество является 1С: Предприятие 8.0. УПП.

Затраты на разработку и внедрение АСУП. Эффективность

Затраты на этапах эксплуатации и сопровождения могут значительно превышать затраты при приобретении (разработке) и внедрении. На рис. 1 представлены затраты в течение жизненного цикла АСУП и эффект от её использования, в котором сосредотачиваются основные, функциональные критерии качества, отражающие назначение, область применения и качественные характеристики функций [1].

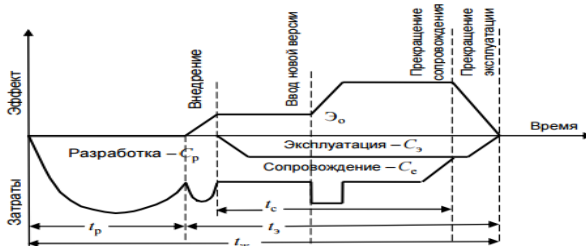


Рис. 1. Затраты в течение жизненного цикла

Принимая решение о комплексной информатизации предприятия, необходимо оценить предполагаемую эффективность АСУП, которую в первом приближении можно определить как разницу между совокупным

доходом от ее использования и затратами в течение ее жизненного цикла. Например, при внедрении АСУП, поддерживающих методологию MRP II, были получены следующие результаты:

- снижение уровня запасов (включая материалы, незавершенное производство, готовую продукцию) – в среднем на 17 %, лучший результат – 25 %;
- улучшение обслуживания клиентов (повышение доли своевременных поставок) – в среднем на 16 %, лучший результат – 28 %;
- повышение производительности – в среднем на 10 %, лучший результат – 16 %;
- снижение себестоимости закупаемых материальных ресурсов – в среднем на 7 %, лучший результат – 11 %.

Для того чтобы определить эффективность внедряемой АСУП для конкретного предприятия, необходимо определить систему показателей эффективности автоматизируемых бизнес- процессов и провести их прогнозную оценку. Система показателей разрабатывается индивидуально для каждого бизнес-процесса[3].

Вывод

Внедрение средств автоматизации может привести к корректированию самого бизнес-процесса, так как задачи выполняются быстрее. Сотрудники могут обрабатывать большие объемы информации за свое рабочее время, что можно использовать или для уменьшения затрат на персонал или для быстрого развития бизнеса при неизменности количества сотрудников, занятых обработкой информации. Как показывает практика автоматизация бизнес процессов несет в себе большой потенциал для развития и материальную выгоду с течением времени.

Проанализировав зарубежные и отечественные АСУП было принято решение отдать предпочтение 1С: Предприятию 8.0. УПП, т.к. данная система имеет широкое распространение в странах СНГ, большое количество специалистов, готовых внедрить и сопровождать систему и относительно невысокую стоимость.

Список литературы

1. ИНТУИТ [Electronic resource] / Интернет-ресурс. – Режим доступа [www/URL: http://www.intuit.ru/studies/courses/1055/271/lecture/6886?page=3](http://www.intuit.ru/studies/courses/1055/271/lecture/6886?page=3)
2. Mathnet [Electronic resource] / Интернет-ресурс. – Режим доступа [www/URL: http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=vagtu&paperid=48&option_lang=rus](http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=vagtu&paperid=48&option_lang=rus)
3. Kazedu [Electronic resource] / Интернет-ресурс. – Режим доступа [www/URL: https://www.kazedu.kz/referat/43280](https://www.kazedu.kz/referat/43280)

УДК 681.518.22

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЗВУКОАНАЛИЗИРУЮЩЕГО ПУЛЬСОМЕТРА В СРЕДЕ NI LabVIEW

Т.Г.Москалюк, А.Ю.Харитонов

Донецкий национальный технический университет, г. Донецк
кафедра компьютерных систем мониторинга
tarasusthelight@gmail.com

Аннотация

Москалюк Т.Г., Харитонов А.Ю. Разработка программного обеспечения звукоанализирующего пульсометра в среде NI LabVIEW. Выполнен анализ средств фонокардиографии. Разработан программный продукт, реализующий графическое представление фонокардиограмм, фильтрацию и анализ записанных фонокардиограмм.

Ключевые слова: *Кардиограф, звук, фонокардиограф, кардиография, звукоанализ, фильтрация, базы данных.*

Постановка проблемы. В процессе проектирования медицинских устройств прямая разработка реально существующих систем трудозатратна и ресурсоёмка. Для уменьшения затрат и трудозатрат на разработку следует создать математическую, а следом и программную модель проектируемой системы. Это позволяет провести эксперименты с программной моделью, без сложностей и затрат осуществить изменения и оптимизацию оборудования.

Введение. Фонокардиография – метод графической регистрации тонов и шумов сердца, не всегда определяемых при аускультации (прослушивания). В диагностировании пороков сердца аускультация имеет большое значение. При выслушивании необходимо знать особенности звуков сердца, при этом определённое значение имеет особенность слуха врача. Фонокардиография же даёт возможность объективно проводить качественный и количественный анализ тонов и шумов сердца [1].

Постановка задачи. Цель данного исследования – разработка программного продукта – фонокардиографа, который позволит визуализировать записанный сигнал, усиливать его, автоматизировать расчет длительности промежутков между первыми тонами, автоматизировать расчет длительности одного тона, определять длительность тона. Задача программного продукта – помочь медицинскому персоналу в диагностике различных заболеваний человека.

При движении крови по сосудам, движении клапанов, сокращении сердца, возникают различные колебательные движения. Наслоения их друг на друга создают звук. Вибрации с частотой 6-10 колебаний в секунду, т.е. 6-10

Гц, не являются источником звука, и не улавливаются слухом[1]. Если к грудной клетке приложить регистрирующее устройство — микрофон, и результат записи графически представить — получится графическое изображение звука, которое используется в диагностике. Программный продукт-же работает с цифровыми данными звукозаписи.

Безвредность и простота метода позволяют производить исследования даже у больного, находящегося в тяжелом состоянии, и с частотой, необходимой для решения диагностических задач.

В данном исследовании, при проектировании программы, будет использована среда разработки NI LabVIEW.

Также, NI LabVIEW предоставляет широкие возможности визуализации полученных и результирующих данных, вплоть до демонстрации самого процесса в схематическом виде. Эта особенность удобна для визуализации различных графиков, навигации по ним, соответственно, работа медицинского персонала облегчается посредством удобства обращения с результатами вычислений программы.

В основной пакет NI LabVIEW встроены пакеты, обеспечивающие поддержку работы с базами данных, включая БД MySQL, используя подключение компьютера к интернету. Работа с сетевыми базами данных позволяет создать легкодоступный и администрируемый, централизованный сервис для обмена данными.

Результаты.

В результате данного исследования был определен состав программных и физических компонентов системы фонокардиографа. Было определено, что нет необходимости в специализированном устройстве, и для работы системы достаточно простого звукозаписывающего оборудования, способного создавать файлы в формате .wav, и персонального компьютера, обладающего достаточными вычислительными мощностями для поддержки работы ОС, и необходимого пакета LabVIEW Runtime Engine.

Такой способ компоновки программных и аппаратных средств, задействованных в системе, обеспечивают гибкость в выборе рабочего оборудования. В качестве записывающего устройства может выступать как высококачественное звукозаписывающее оборудование, так и микрофон мобильного устройства, способного записывать звук в формате .wav, независимо от операционной системы. Простота использования может дойти вплоть до самостоятельной записи звукового файла пациентом, руководствуясь инструкциями врача, и отправки записи медицинскому персоналу используя сеть интернет, или физические носители, для дальнейшего анализа. Малый объем подобных файлов способствует накоплению информации о пациенте, с возможностью составления статистики о жизненных данных пациента, и прогрессии/регрессии заболевания.

Задача программного продукта — звуковой анализ файла с записью активности сердца, вывод графического представления звука, усиление звука, фильтрация звука, расчет длительности промежутков между первыми тонами, расчет длительности одного тона.

Такие задачи решаются с помощью оцифровки звуковой волны, и обработки массивов, оцифрованных данных, спектрального анализа звука, алгоритмов поиска пиков, звукового фильтра Чебышева.

Фильтр Чебышёва — один из типов линейных аналоговых или цифровых фильтров, отличительной особенностью которого является более крутой спад амплитудно-частотной характеристики и существенные пульсации амплитудно-частотной характеристики на частотах полос пропускания (фильтр Чебышёва I рода) и подавления (фильтр Чебышёва II рода), чем у фильтров других типов. Фильтр получил название в честь известного русского математика XIX века Пафнутия Львовича Чебышёва, так как характеристики этого фильтра основываются на многочленах Чебышёва [2].

В результате было разработано программное обеспечение, реализующее обработку звука с выводом информации на экран, и предусмотренной работой с базой данных MySQL.

Интерфейс программы представляет собой набор графиков и элементов управления, способствующих повышению качества фильтрации и усиления звуковых данных, и представлен на рис. 1

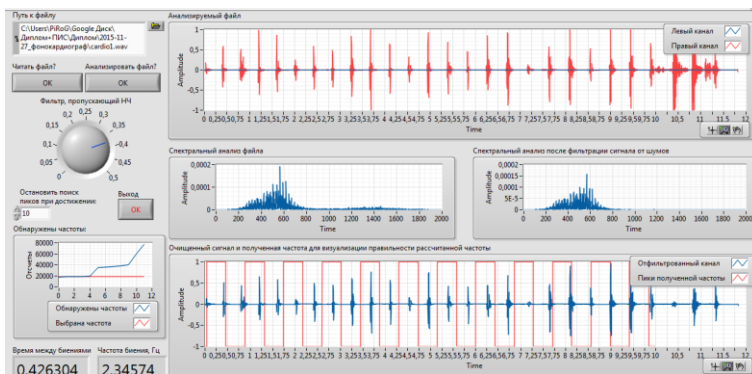


Рис. 1 – Интерфейс программы

NI LabVIEW, используя data flow архитектуру - архитектуру потоков данных, предоставляет возможность использования графического программирования. Данная особенность позволяет человеку без опыта программирования приступить к выполнению задачи в минимальный срок, а схематичность реализованных программ облегчает отладку и модификацию систем. На рис.2 изображена блок-диаграмма программного модуля, занятого

звукоанализом. На рис.3 изображена блок-диаграмма модуля, занятого визуальной частью интерфейса программного продукта.

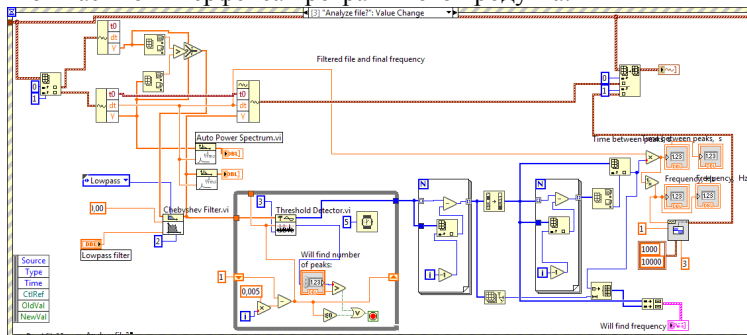


Рис.2 - Блок-диаграмма звукоанализирующего модуля

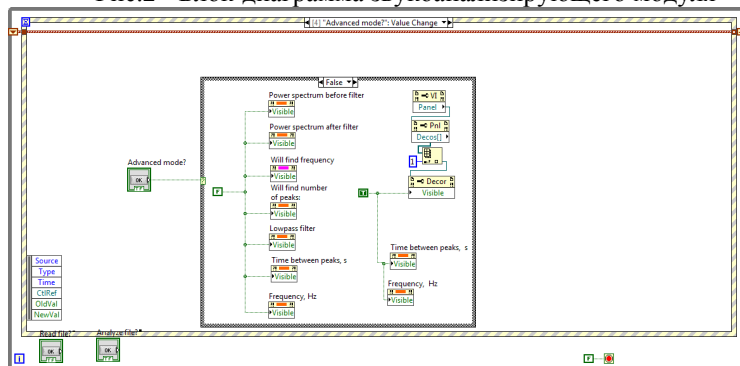


Рис.3 - Блок-диаграмма модуля визуальной части

Выводы

Данное исследование несёт практическую пользу, а также социально-экономический эффект для потребителей. Программный продукт снизит финансовые затраты, исключив необходимость приобретения дорогостоящего оборудования, и увеличит качество услуг медицинских учреждений, предоставляя дешёвый и достаточно точный инструмент диагностики заболеваний сердечно-сосудистой системы.

Список литературы

1. Юлдашев К.Ю.и др. Краткое пособие по электро- и фонокардиографии — Медицина, 1984 — 191 с., ил., табл.
2. Б.Х. Кривичкий Справочник по теоретическим основам радиоэлектроники. — М.: Энергия, 1977.

УДК 004.048+004.855

АНАЛИЗ СТАТИСТИЧЕСКИХ ДАННЫХ ДЛЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПРОДАЖ

Вудвуд Е.Ю., доц. Васяева Т.А.

Донецкий национальный технический университет, г.Донецк

кафедра АСУ

E-mail: katya-1_93@mail.ru

Аннотация

Вудвуд Е.Ю., доц. Васяева Т.А. Анализ статистических данных для системы прогнозирования продаж. *В статье рассмотрена задача прогнозирования продаж строительных материалов. Выполнен анализ и предварительная обработка входных данных. Проведены экспериментальные исследования с использованием данных, предоставленных частным предприятием. Построены и исследованы временные ряды продаж строительных материалов за 2015 – 2016 гг.*

Ключевые слова: *временной ряд, строительные материалы, корреляция, автокорреляционная функция.*

Общая постановка проблемы. Любое торговое предприятие в современных условиях представляет собой довольно сложный комплекс различного рода звеньев, от слаженной работы которых зависит успех предприятия в целом. Объектом компьютеризации является процесс обработки информации на частном предприятии, которое занимается сбытом строительных материалов. Частное предприятие состоит из головного офиса, где находятся: генеральный директор, бухгалтера и координатор по продажам, и двух торговых точек, на которых находятся управляющие, несущие непосредственную ответственность за сохранность материальных ценностей, их учет, контроль за перемещением. Координатор по продажам руководит работой управляющих, осуществляет прием заказов из регионов, контролирует правильность указания платежных и отгрузочных реквизитов в соответствующих документах, передает на торговые точки сопроводительные документы на отгрузку материалов, ведет учет отгрузок продукции и получения товара регионами. Бухгалтер предоставляет информацию про количество средств, которые пришли от клиентов, и наличие задолженостей перед поставщиками. Управляющие торговых точек ведут учет количества стройматериалов.

Для управления предприятием необходимо уметь предвидеть будущее состояние предприятия и среды, в которой оно существует, вовремя

предупредить сбои и срывы в работе, поэтому стал актуальным вопрос о применении прогнозирования в деятельности данного предприятия.

Прогнозирование – одна из основных составляющих управленческого процесса. Игнорируя прогнозирование и планирование своей работы, руководитель предприятия может столкнуться с такими ситуациями:

1. Товар находится на складе, не принося прибыль, так как нет на него спроса;
2. Отсутствие товара из-за большого спроса.

Любая из этих ситуаций отрицательно скажется на работе предприятия в целом.

Целью работы является создание компьютеризированной системы прогнозирования спроса на товар, что повлечет за собой эффективное планирование закупок товара и как следствие увеличение прибыли.

Подсистема позволит:

1. Обеспечить наличие товара в нужном количестве.
2. Минимизировать упущенные продажи.
3. Повысить эффективность контроля и выполнения процессов по управлению товарооборотом.

Исследования. В качестве исследуемых данных были рассмотрены данные, отражающие объемы продаж строительных материалов за 2015 – 2016 года (табл. 1-3). На основе этих данных были построены временные ряды (рис. 1-3). Временной ряд — это последовательность упорядоченных во времени числовых показателей, характеризующих уровень состояния и изменения изучаемого явления [1].

Таблица 1 – Продажи по месяцам за 2015 – 2016 гг строительных материалов (ДВП)

2015 год												
Месяц	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Продажи	280	265	350	360	420	657	790	761	530	400	350	330
2016 год												
Месяц	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Продажи	290	300	375	475	450	700	750	683	500	350	300	243

Таблица 2 – Продажи по месяцам за 2015 – 2016 гг строительных материалов (шлакоблок)

2015 год												
Месяц	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Продажи	356	230	310	330	510	725	1200	1520	800	720	425	480
2016 год												
Месяц	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Продажи	320	250	350	310	480	790	1010	1310	725	620	470	320

Таблица 3 – Продажи по месяцам за 2015 – 2016 гг строительных материалов (рубероид)

2015 год												
Месяц	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Продажи	31	20	28	47	83	132	147	138	102	75	57	21
2016 год												
Месяц	Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Продажи	26	27	35	43	76	181	172	171	106	52	31	15

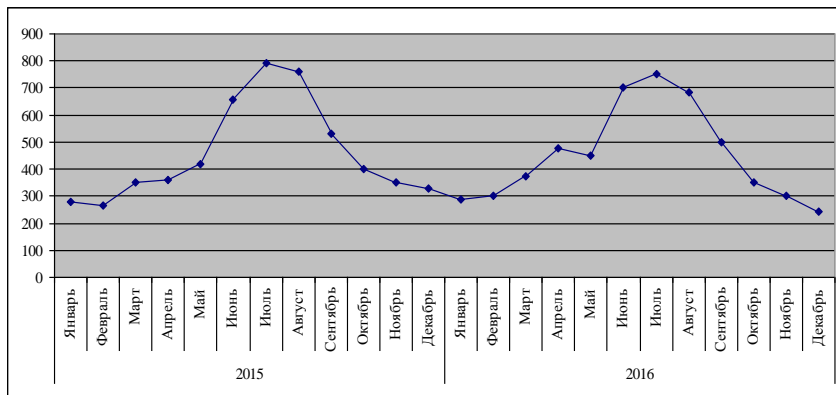


Рисунок 1 – График продаж ДВП по месяцам за 2015 – 2016 гг.

При анализе и изучении временного ряда аналитик должен на основе некоторого отрезка ряда конечной длины сделать выводы о характере и закономерностях процесса, который описывается данным рядом. Таким образом, цель анализа временного ряда – описание характерных особенностей ряда, для дальнейшего построения модели ряда для предсказания будущих значений на основе прошлых наблюдений.

Для этого будем использовать аппарат корреляционного анализа. Корреляция – взаимосвязь двух или более величин, при этом изменения значений одной из величин сопутствует изменению значений другой величины [2]. При анализе временных рядов вычисляется автокорреляция ряда и строится автокорреляционная функция.

Для вычисления автокорреляции ряда, использовалась его копия, сдвинутая в сторону запаздывания на определенное количество отсчетов (табл. 4).

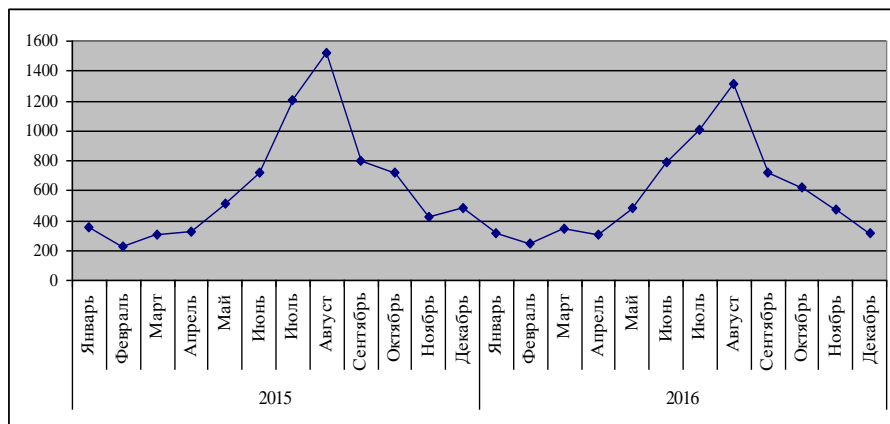


Рисунок 2 – График продаж шлакоблока по месяцам за 2015 – 2016 гг.

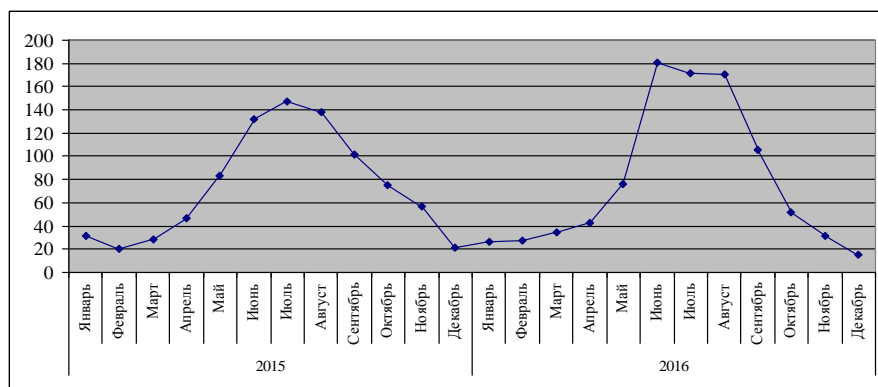


Рисунок 3 – График продаж рубероида по месяцам за 2015 – 2016 гг.

Ряд X_{i-1} представляет собой копию данного ряда, сдвинутую на один временной интервал и т.д. Для определения степени зависимости элементов временного ряда используется коэффициент автокорреляции r_k , где k – количество отсчетов сдвига временного ряда при вычислении коэффициента. Коэффициент автокорреляции вычисляем по формуле:

$$r_k = \frac{\sum_{i=k+1}^n (x_i - \bar{x})(x_{i-k} - \bar{x})}{\sum_{i=k+1}^n (x_i - \bar{x})^2} \quad (1)$$

Таблица 4 – Данные для расчета автокорреляционной функции (АКФ) на примере данных по продажам ДВП

X	280	265	350	360	420	657	790	761	530	400	350	330	290	...	350	300	243
X_{i-1}		280	265	350	360	420	657	790	761	530	400	350	330	...	500	350	300
X_{i-2}			280	265	350	360	420	657	790	761	530	400	350	...	683	500	350
X_{i-3}				280	265	350	360	420	657	790	761	530	400	...	750	683	500
X_{i-4}					280	265	350	360	420	657	790	761	530	...	700	750	683
X_{i-5}						280	265	350	360	420	657	790	761	...	450	700	750
X_{i-6}							280	265	350	360	420	657	790	...	475	450	700
X_{i-7}								280	265	350	360	420	657	...	375	475	450
X_{i-8}									280	265	350	360	420	...	300	375	475
...																	
X_{i-n}																	280

где x_i – значение i -го отсчета;

x_{i-k} – наблюдение x_i со сдвигом на k временных отсчетов;

\bar{x} – среднее значение ряда.

Коэффициент автокорреляции изменяется в диапазоне $[-1;1]$, где $r_k = 1$ означает полную корреляцию. Рассчитав коэффициенты автокорреляции для каждого сдвига, получим автокорреляционную функцию. Автокорреляционная функция – последовательность коэффициентов автокорреляции уровней первого, второго и т. д. порядков. Расчет АКФ на примере данных по продаже ДВП представлен в таблице 5, соответствующий график – на рис. 4. Также представлены расчет АКФ для шлакоблока и рубероида (табл. 6-7) и соответствующие графики (рис. 5-6).

Таблица 5 – Результаты АКФ по продажам ДВП

k	0	1	2	3	4	5	6	7	8	9	10	11
r_k	1	0,77	0,76	0,75	0,77	0,79	0,78	0,74	0,79	0,79	0,80	0,80
k	12	13	14	15	16	17	18	19	20	21	22	
r_k	0,78	0,77	0,76	0,75	0,79	0,90	0,92	0,87	0,7	0,44	-1	

Таблица 6 – Результаты АКФ по продажам шлакоблока

k	0	1	2	3	4	5	6	7	8	9	10	11
r_k	1	0,69	0,68	0,69	0,69	0,74	0,71	0,70	0,72	0,72	0,71	0,70
k	12	13	14	15	16	17	18	19	20	21	22	
r_k	0,68	0,68	0,67	0,69	0,70	0,95	0,94	0,77	0,04	-0,65	-1	

Таблица 7 – Результаты АКФ по продажам рубероида

k	0	1	2	3	4	5	6	7	8	9	10	11
r_k	1	0,78	0,77	0,77	0,8	0,82	0,79	0,74	0,84	0,85	0,85	0,85
k	12	13	14	15	16	17	18	19	20	21	22	
r_k	0,83	0,81	0,83	0,84	0,87	0,92	0,93	0,95	0,83	-0,04	-1	

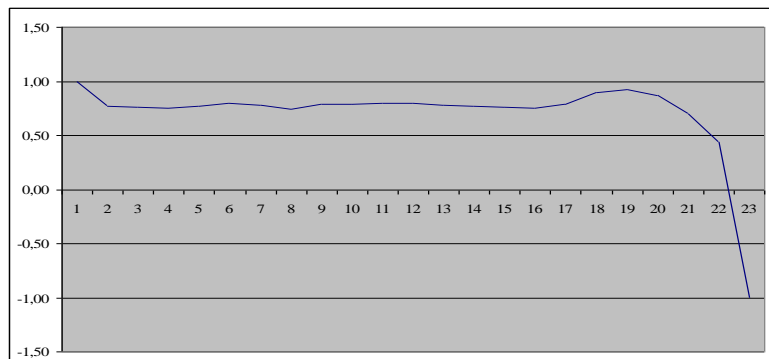


Рисунок 4 – Автокорреляционная функция временного ряда продаж ДВП

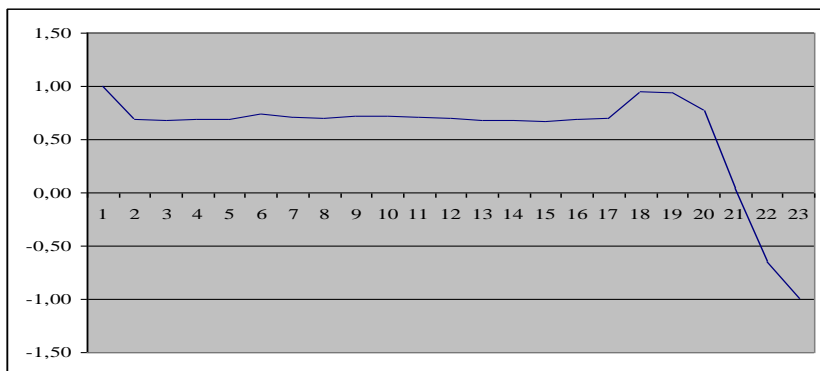


Рисунок 5 – Автокорреляционная функция временного ряда продаж
шлакоблока

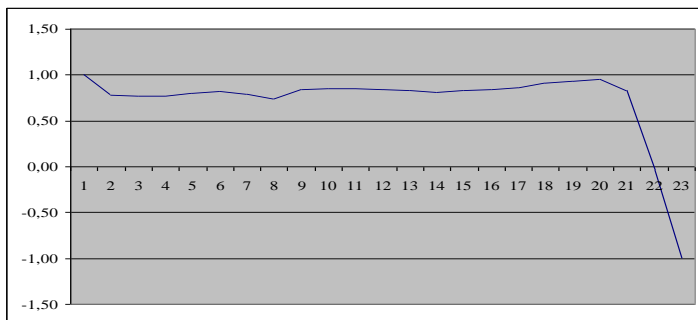


Рисунок 6 – Автокорреляционная функция временного ряда продаж рубероида

Корреляционный анализ позволяет прийти к следующим заключениям о поведении временного ряда.

1. Если ряд содержит тренд, то степень взаимной зависимости между последовательными значениями ряда и корреляция между ними очень высоки. При этом коэффициент автокорреляции значителен для первых нескольких сдвигов ряда, а в дальнейшем убывает до нуля.

2. Если действие случайной компоненты велико, то коэффициенты автокорреляции для любого значения сдвига будут близки к нулю. Высокая изменчивость ряда, являющаяся следствием воздействия случайной компоненты, приводит к уменьшению взаимной связи между последовательными значениями ряда и, соответственно, к уменьшению коэффициента автокорреляции.

3. Если ряд содержит сезонную компоненту, то коэффициент автокорреляции будет большим для значений сдвига, равных периоду сезонной составляющей или кратных ему.

Выводы. Из проделанной работы можно сказать, от каких предшествующих значений зависят будущие. При анализе автокорреляционных функций и визуальном анализе временных рядов можно сделать выводы о наличии сезонной компоненты. Также наблюдается высокая степень корреляции для некоторых сдвигов, это будет учтено при построении прогнозирующей модели.

Список литературы

1. Афанасьев В.Н., Юзбашев М.М. Анализ временных рядов и прогнозирование: Учебник. — М.: Финансы и статистика, 2001.

2. Корреляция // Википедия. [2017—2017].
URL: <http://ru.wikipedia.org/?oldid=85047438>

УДК 681.771-3:65852

ИНТЕРАКТИВНАЯ ПОДДЕРЖКА В СИСТЕМЕ АНАЛИЗА ДОСТОВЕРНОСТИ ПЕРВИЧНОЙ ИНФОРМАЦИИ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ РАСКРОЕМ ПРОКАТА НА МЕТАЛЛУРГИЧЕСКОМ ПРЕДПРИЯТИИ

Раскидкин В.В., Достлев Ю.С.

Донецкий Национальный Технический Университет
кафедра компьютерной инженерии

E-mail: vitya-raskidkin@mail.ru

Аннотация

Раскидкин В.В., Достлев Ю.С. Интерактивная поддержка в системе анализа достоверности первичной информации системы автоматического управления раскромом проката на металлургическом предприятии. Проведён анализ первичной информации системы управления раскромом на ножницах стана 500 и обводной линии на непрерывного заготовочного стана (НЗС) Блюминга. Разработана программа сортировки данных пореза плавок для детального их изучения.

Ключевые слова: системы, металлургия, прокат

Введение

Производство металлического профиля осуществляется на специализированных металлургических заводах. Получение готового профиля включает в себя преобразование слитка в готовый профиль. Последовательность преобразований выполняется на последовательности станов, каждый из которых реализует определённую операцию, и результаты работы которого являются заготовкой для последующих станов(рис.1). [1]

В ходе работы НЗС может происходить ряд проблем, нарушающих достоверность полученных данных, например засвет фотодатчика, что приводит к его ложному срабатыванию. Основой проверки достоверности срабатывания датчиков является визуализация и возможность сортировки данных, полученных из файлов первичной информации, таких как интервалы времени прохода плавок за датчики стана 500 и обводной линии НЗС. Для достижения данной цели и была разработана программа.

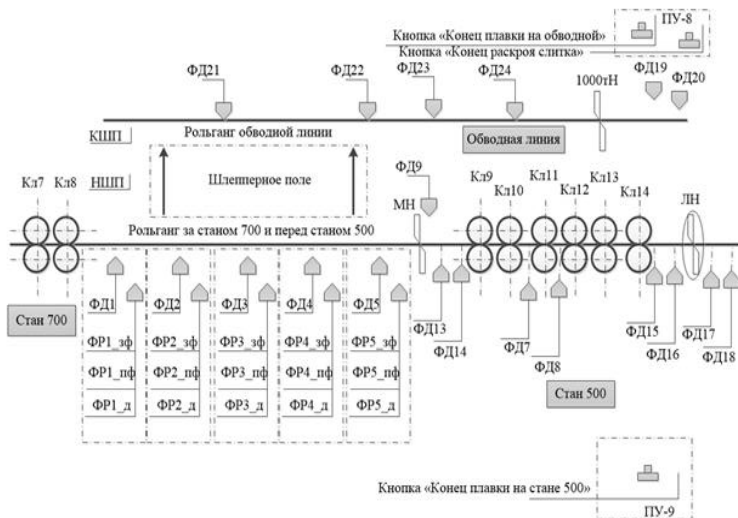


Рис. 1. Структура расположения датчиков сбора первичной информации о состояниях оборудования и параметрах раскря системы управления раскряем на ножницах НЗС

Структура файла исходной информации раскря плавок

В ходе анализа был рассмотрен файл, содержащий данные о порезе слитков плавок на штанги на стане 500 и обводной линии на НЗС Блюминга. Структура файла представляет собой набор информации о процессах раскря на рассматриваемых участках НЗС. В файле содержатся данные о полученных слитках, штангах, их количестве в одной плавке, прогнозируемой длине, средней длине, длине каждого отдельного слитка(рис. 2.1, рис. 2.2). [2]

02.11.14 07:11:40	Раскат обв.	Сл	0	Дл=	49,16	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	1,88	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:11:40	Раскат обв.	Сл	0	Дл=	54,91	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	6,83	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:12:47	Раскат обв.	Сл	0	Дл=	49,47	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	1,39	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:13:31	Раскат обв.	Сл	0	Дл=	55,71	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	7,63	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:14:26	Раскат обв.	Сл	0	Дл=	48,84	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	0,76	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:15:05	Раскат обв.	Сл	0	Дл=	55,59	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	7,51	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:15:47	Раскат обв.	Сл	0	Дл=	48,69	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	0,61	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:16:42	Раскат обв.	Сл	0	Дл=	49,47	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	1,39	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:17:18	Раскат обв.	Сл	0	Дл=	49,44	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	1,36	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:18:03	Раскат обв.	Сл	0	Дл=	54,60	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	6,52	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:18:53	Раскат обв.	Сл	0	Дл=	49,17	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	1,09	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:19:46	Раскат обв.	Сл	0	Дл=	53,89	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	5,81	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:20:30	Раскат обв.	Сл	0	Дл=	49,47	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	1,39	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:21:05	Раскат обв.	Сл	0	Дл=	54,17	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	6,09	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:22:18	Раскат обв.	Сл	0	Дл=	48,78	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	0,70	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:22:44	Раскат обв.	Сл	0	Дл=	56,27	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	8,19	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:23:28	Раскат обв.	Сл	0	Дл=	49,27	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	1,19	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:24:15	Раскат обв.	Сл	0	Дл=	55,01	Диал=	1	Дшт=	11,87	Шт=	4	Ост=	6,93	Слл=	0	В=	0	Ш=	0	Корр.=50	Автомат	Мера
02.11.14 07:24:56	Плавка обв.		02.11.2014 7:24:56	[Сл=	18	Шт=72		[Срд=	11,87	Рез1000т=0	[Кор.=8	[Ср.дл.кор=6,93										

Рис. 2.1. Фрагмент исходного файла данных раскря плавок на обводной линии

02.11.14 07:39:27	Прогноз	Черн= 43,09 Прогн= 144,91 Нпрогн= 15 Дл.шт= 11,10 Штанг= 13 Ост= 0,11 Черн. скор= 186
02.11.14 07:39:28	Прогноз	Черн= 42,54 Прогн= 145,18 Нпрогн= 0 Дл.шт= 11,13 Штанг= 13 Ост= 0,02 Черн. скор= 186
02.11.14 07:39:37	Штанга	Шт= 1 Дл.пл= 11,13 Дл.ф= 11,31 Ост.раскр.= 0,02 Дл.св= 11,95 Нпр= 0 Прок= 0 V= 636 Код=000AB
02.11.14 07:39:39	Штанга	Шт= 2 Дл.пл= 11,13 Дл.ф= 11,24 Ост.раскр.= 0,02 Дл.св= 11,95 Нпр= 0 Прок= 0 V= 641 Код=000AB
02.11.14 07:39:40	Прогноз	Черн= 42,54 Прогн= 110,91 Нпрогн= 9 Дл.шт= 11,03 Штанг= 13 Ост= 0,04 Черн. скор= 186
02.11.14 07:39:40	Прогноз	Черн= 42,54 Прогн= 111,04 Нпрогн= 4 Дл.шт= 11,04 Штанг= 13 Ост= 0,08 Черн. скор= 186
02.11.14 07:39:41	Штанга	Шт= 3 Дл.пл= 11,13 Дл.ф= 11,25 Ост.раскр.= 0,08 Дл.св= 12,00 Нпр= 0 Прок= 0 V= 642 Код=000AB
02.11.14 07:39:43	Штанга	Шт= 4 Дл.пл= 11,04 Дл.ф= 11,07 Ост.раскр.= 0,08 Дл.св= 11,95 Нпр= 4 Прок= 0 V= 642 Код=000DA
02.11.14 07:39:43	Прогноз	Черн= 42,54 Прогн= 88,93 Нпрогн= 14 Дл.шт= 11,04 Штанг= 13 Ост= 0,05 Черн. скор= 186
02.11.14 07:39:44	Штанга	Шт= 5 Дл.пл= 11,04 Дл.ф= 11,04 Ост.раскр.= 0,05 Дл.св= 11,95 Нпр= 4 Прок= 0 V= 642 Код=000DA
02.11.14 07:39:44	Прогноз	Черн= 42,54 Прогн= 77,89 Нпрогн= 14 Дл.шт= 11,04 Штанг= 13 Ост= 0,05 Черн. скор= 186
02.11.14 07:39:46	Штанга	Шт= 6 Дл.пл= 11,04 Дл.ф= 11,01 Ост.раскр.= 0,05 Дл.св= 11,95 Нпр= 14 Прок= 0 V= 641 Код=000E1
02.11.14 07:39:46	Прогноз	Черн= 42,54 Прогн= 66,88 Нпрогн= 14 Дл.шт= 11,05 Штанг= 13 Ост= 0,03 Черн. скор= 186
02.11.14 07:39:47	Прогноз	Черн= 42,54 Прогн= 66,59 Нпрогн= 11 Дл.шт= 11,04 Штанг= 13 Ост= 0,05 Черн. скор= 186
02.11.14 07:39:48	Штанга	Шт= 7 Дл.пл= 11,04 Дл.ф= 11,00 Ост.раскр.= 0,05 Дл.св= 11,88 Нпр= 14 Прок= 0 V= 641 Код=000E1
02.11.14 07:39:49	Штанга	Шт= 8 Дл.пл= 11,04 Дл.ф= 11,03 Ост.раскр.= 0,05 Дл.св= 11,95 Нпр= 11 Прок= 0 V= 641 Код=000DA
02.11.14 07:39:49	Прогноз	Черн= 42,54 Прогн= 44,52 Нпрогн= 14 Дл.шт= 11,05 Штанг= 13 Ост= 0,02 Черн. скор= 186
02.11.14 07:39:51	Штанга	Шт= 9 Дл.пл= 11,04 Дл.ф= 11,01 Ост.раскр.= 0,02 Дл.св= 11,81 Нпр= 11 Прок= 0 V= 640 Код=000DA
02.11.14 07:39:51	Прогноз	Черн= 42,54 Прогн= 33,50 Нпрогн= 14 Дл.шт= 11,06 Штанг= 13 Ост= 0,02 Черн. скор= 186
02.11.14 07:39:53	Штанга	Шт= 10 Дл.пл= 11,05 Дл.ф= 10,98 Ост.раскр.= 0,02 Дл.св= 11,95 Нпр= 14 Прок= 0 V= 638 Код=000D6
02.11.14 07:39:53	Прогноз	Черн= 42,54 Прогн= 22,51 Нпрогн= 14 Дл.шт= 11,10 Штанг= 13 Ост= 0,01 Черн. скор= 186
02.11.14 07:39:55	Штанга	Шт= 11 Дл.пл= 11,06 Дл.ф= 10,94 Ост.раскр.= 0,01 Дл.св= 11,84 Нпр= 14 Прок= 0 V= 634 Код=000D3
02.11.14 07:39:55	Прогноз	Черн= 42,54 Прогн= 11,53 Нпрогн= 14 Дл.шт= 11,23 Штанг= 13 Ост= 0,00 Черн. скор= 186
02.11.14 07:39:56	Штанга	Шт= 12 Дл.пл= 11,10 Дл.ф= 11,05 Ост.раскр.= 0,00 Дл.св= 11,95 Нпр= 14 Прок= 0 V= 635 Код=000B9
02.11.14 07:39:58	Штанга	Шт= 13 Дл.пл= 11,23 Дл.ф= 11,44 Ост.раскр.= 0,00 Дл.св= 11,87 Нпр= 14 Прок= 0 V= 638 Код=0009C
02.11.14 07:39:58	Раскат	Сл: 1 Черн= 42,54 Чист= 144,83 Шт= 13 Зач.п= 0,00 Зач.з= 0,71 Дл.п= 11,44 Выт= 3,40
02.11.14 07:39:58	Раскат доп.	Срдн= 11,10 Кор= 0 РезМн= 0 Раскат= 0 ШПр= 0 Хобор= 0,00 Корр= 25 Дл. кор= 0,00 Авт= 1 Диапазон
02.11.14 07:40:04	Плавка	K800 02.11.2014 7:40:04 Шл 0 Сл= 1 Шт=13 Срдн=11,10 РезМн=0 Корр=0 Ср.дл.кор=0,00

Рис. 2.2. Фрагмент исходного файла данных раскроя плавков на стане 500

Исходя из этих данных, была начата разработка программы, сортирующая все эти данные по определённым категориям для более детального анализа.

Разработка программы анализа первичной информации

На данном этапе разработки, программа полностью анализирует и сортирует выбранный файл пореза. Файл данных пореза для дальнейшей его обработки добавляется путём нажатия в интерфейсе кнопки "Порез" и дальнейшем его выборе в проводнике. При этом полный путь к файлу будет показан в соседнем с кнопкой текстовке. В окне программы шкала раскроя поделена на смены работы на НЗС. На самой шкале по временным промежуткам разделены данные на плавки, проходящие через стан 500 или обводную линию НЗС Блюминга. Так же на шкале отмечены временные отрезки простоя непрерывного заготовочного стана. На данной шкале можно посмотреть номер слитка на плавке рассматриваемого стана и момент времени, в который данный слиток анализируется(рис. 3). На шкале плавков используются следующие обозначения: интервал времени прохода плавки через обводную линию обозначен красным цветом(1), интервал времени прохода плавки через стан 500 - синим(2), интервал простоя НЗС – чёрным. Верхняя часть шкалы поделена на смены таким образом, что белым цветом отмечены первая и треть смены и чёрным – вторая смена работы. Ниже основной шкалы расположена шкала разбивки суток по часам. По данной шкале видно, что заполнена лишь половина основной шкалы. Это происходит

по причине того, что на данном этапе программа обрабатывает лишь один файл с исходной информацией за раз, а в одном файле хранится информация за 12 часов работы НЗС.

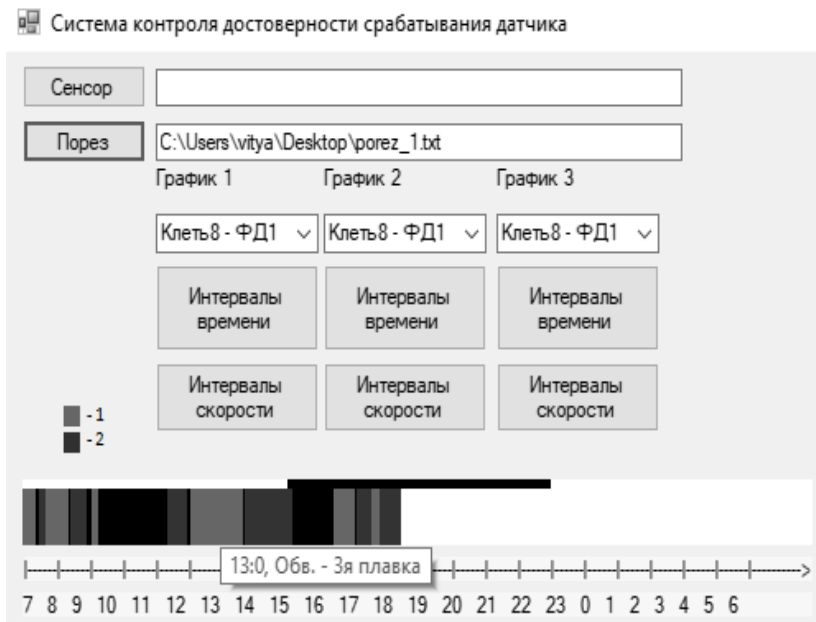


Рис. 3. Фрагмент окна программы

При нажатии на определённую плавку, в дочернем окне отображается полное её описание, включая результирующую строку с полным отчётом о прошедшей плавке(рис. 4).

Плавка

Дата, время блока	Наименование	Параметры
02.11.14 12:43:30	Раскат обв.	Сп 0 Дл= 50,65 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 2,57 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:50:15	Раскат обв.	Сп 0 Дл= 49,62 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,94 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:51:05	Раскат обв.	Сп 0 Дл= 54,71 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 6,63 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:52:28	Раскат обв.	Сп 0 Дл= 49,34 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,26 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:53:13	Раскат обв.	Сп 0 Дл= 50,00 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,92 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:53:56	Раскат обв.	Сп 0 Дл= 49,48 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,40 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:54:39	Раскат обв.	Сп 0 Дл= 49,70 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,62 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:55:18	Раскат обв.	Сп 0 Дл= 49,04 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 0,96 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:56:05	Раскат обв.	Сп 0 Дл= 49,61 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,53 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:56:45	Раскат обв.	Сп 0 Дл= 48,71 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 0,63 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:57:30	Раскат обв.	Сп 0 Дл= 49,30 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,22 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:58:14	Раскат обв.	Сп 0 Дл= 49,06 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 0,98 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:58:58	Раскат обв.	Сп 0 Дл= 49,63 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,55 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 12:59:47	Раскат обв.	Сп 0 Дл= 48,88 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 0,80 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 13:00:26	Раскат обв.	Сп 0 Дл= 49,67 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,59 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 13:01:06	Раскат обв.	Сп 0 Дл= 48,90 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 0,82 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 13:01:46	Раскат обв.	Сп 0 Дл= 49,57 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,49 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 13:02:29	Раскат обв.	Сп 0 Дл= 49,47 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,39 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 13:03:10	Раскат обв.	Сп 0 Дл= 49,31 Дкват= 1 ДлШт= 11,87 Шт= 4 Ост= 1,23 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 13:04:08	Раскат обв.	Сп 0 Дл= 63,96 Дкват= 1 ДлШт= 11,87 Шт= 5 Ост= 4,01 СлШл= 0 В= 0 Ш= 0 Корр =50 Автомат Мера
02.11.14 13:05:47	Плавка обв.	02.11.2014 13:05:47 Сл= 21 Шт=89 СрДл= 11,87 Рез1000=0 Кор =2 Ср.дл.кор=5,32

Рис. 4. Дочернее окно выбранной плавки

Вывод

Данная программа разработана для облегчения и повышения достоверности анализа информации о срабатывании датчиков, сопровождающих раскаты по линии прокатки и раскроя, с непрерывного заготовочного стана. Для этого данные визуализируются и сортируются по сменам, типам стана, плавкам и типам стана, через который плавка проходит (стан 500 или обводная линия).

На данном этапе не учитываются возможные неточности файла первичной информации, поэтому возможны погрешности, такие как отображение работы стана меньше реального промежутка времени, запись одной или нескольких плавков не в тот стан, так как между началом плавки и её окончанием на одном стане успело пройти несколько плавков на другом стане.

В дальнейшем предполагается продолжение работ по совершенствованию интерфейса и добавлению обработки дополнительных файлов для повышения уровня автоматизации анализа достоверности первичных параметров системы раскроя с выделением особых ситуаций.

Список литературы

1. Ю.С. Достлев, О.И. Демьяненко, О.Г. Шевченко и др. "Автоматизация процесса раскроя и учёта проката на НЗС 900/700/500 цеха Блюминг-2 ОАО "КГМК "Криворожсталь"" - "Теория и практика металлургии 3-4'2004, с116"
2. А.А. Коринь, А.М. Теряев, О.И. Демьяненко и др. "Автоматизация процесса раскроя проката на основных и обводных линиях непрерывно-заготовочных станов" - "Металл и литьё Украины №7-9'2001, с 55"

УДК 007.51

МОДЕЛЬ СППР ДЛЯ МАЛОГО ПРЕДПРИЯТИЯ

В.С. Лысенко, С.Ю. Землянская

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

Vadik44444444@yandex.ua, zsaa@ya.ru

Аннотация

Лысенко В.С., Землянская С.Ю. Модель СППР для малого предприятия. Выполнен анализ функционирования малого предприятия, сформулирована постановка задачи, а также цели разрабатываемой системы поддержки принятия решений, разработана модель системы поддержки принятия решений для малого предприятия по изготовлению печатных плат.

Ключевые слова: малое предприятие, СППР, анализ иерархий, ранжирование, альтернатива, ранг, модель.

Постановка проблемы. Процесс управления предприятием состоит из разработки, принятия и реализации стратегических и тактических управленческих решений. Ограниченность человеческих ресурсов и постоянное желание сократить расходы привели к созданию систем, которые могут учитывать различные аспекты, способные повлиять на выбор того или иного варианта в процессе принятия решений, а также определять, какие варианты более предпочтительны.

Облегчение руководящей функции при планировании выполнения заказов и организации их выполнения является масштабной задачей, включающей в себя несколько подзадач.

СППР - это совокупность интеллектуальных информационных приложений и инструментальных средств, которые используются для манипулирования данными, их анализа и представления результатов такого анализа конечному пользователю. Современная СППР позволяет предусматривать степень влияния принятых решений на дальнейшее развитие бизнеса. Таким образом, в СППР объединяются на общей основе подходы, характерные для таких направлений: принятие решений, получения и представления знаний, построение человеко-машинных (диалоговых) систем.

Разрабатываемая СППР необходима для облегчения процесса организации производства и будет решать две задачи:

1. Оценка целесообразности выполнения рассматриваемого заказа исходя из ситуации на предприятии, а также параметров заказа
2. Выбор оптимальной последовательности выполнения рассматриваемого ряда заказов исходя из ситуации на предприятии, а также параметров заказов

Оценка целесообразности выполнения рассматриваемого заказа

Решение данной задачи позволит упростить руководству выбор между тем, стоит ли отклонить данный заказ и не браться за его выполнение, или напротив, взяться. Таким образом, можно сделать вывод, что результатом работы данной функции разрабатываемой СППР должен быть совет руководству из двух возможных:

1. Выполнение данного заказа целесообразно
2. Выполнение данного заказа нецелесообразно

Проанализировав параметры заказа, а также текущую ситуацию на предприятии можно определить степень полезности заказа для данного предприятия. Степень полезности заказа является определяющим критерием, который будет определять целесообразность выполнения данного заказа.

Выбор оптимальной последовательности выполнения рассматриваемого ряда заказов

Вычисление оптимальной последовательности является задачей ранжирования. Решения задачи ранжирования заказов происходит в 2 этапа:

1. Оценка полезности каждого заказа
2. Ранжирование заказов

Проанализировав параметры каждого из заказов, а также текущую ситуацию на предприятии, можно определить степень его полезности для данного предприятия, что позволит определить его приоритет в рамках очереди выполнения заказов на предприятии и ранжировать заказы, сформировав оптимальную для предприятия последовательность их выполнения.

Производственные факторы

На работу предприятия влияет ряд факторов, которые необходимо проанализировать для того, чтобы учесть их влияние в разрабатываемой СППР. Были выделены следующие внутренние факторы:

1. Надежность заказчика (функция N)
2. Степень достаточности материалов для выполнения данного заказа на складе(функция M)
3. Степень загруженности работников, необходимых для выполнения данного заказа(функция Z)
4. Уход работников, необходимых для выполнения данного заказа на больничный(функция B)
5. Прибыльность (функция P)

Также были выделены следующие внешние факторы:

1. Стабильность подачи электроэнергии(функция SE)
2. Стабильность водоснабжения(функция SV)
3. Стабильность общеполитической обстановки в регионе(функция SP)
4. Стабильность поставок материалов, необходимых для производственного процесса(функция SPM)

Модель системы поддержки принятия решений

На основании всех перечисленных производственных факторов и определяется степень полезности каждого заказа для предприятия.

Под такой характеристикой как полезность заказа понимается совокупность двух вычисляемых коэффициентов для каждого из заказов:

1. Коэффициент удобства
2. Коэффициент прибыльности

Так как полезность заказа определяется перечисленными производственными факторами, то, соответственно, и описанные коэффициенты для каждого заказа зависят от приведенных производственных факторов. Коэффициент удобства ($KoefU$) зависит от степени достаточности материалов, степени загруженности работников, от фактора ухода работников на больничный, стабильность подачи электроэнергии, стабильность водоснабжения, стабильность общеполитической обстановки в регионе, стабильность поставок материалов, необходимых для производственного процесса.

$$KoefU = f(M, Z, B, SE, SV, SP, SPM) \quad (1)$$

Коэффициент прибыльности ($KoefP$) зависит от таких факторов как прибыльность и надежность заказчика:

$$KoefP = f(N, P) \quad (2)$$

Из-за неравнозначности факторов принято решение оценить их важность, чтобы учесть это при обучении нейронной сети. Для этого был выбран метод анализа иерархий (МАИ). Этот метод позволяет оценить значимость каждого фактора в рамках поставленной задачи.

СППР должна выполнить оценку целесообразности выполнения заказа и выбор оптимальной последовательности выполнения заказов. В каждой из этих функций необходимо определить удобность и прибыльность каждого заказа. Для реализации этой задачи было решено использовать нейронную сеть, которая будет обучаться на данных предыдущих заказов реально функционирующего предприятия по изготовлению печатных плат. Данные включают в себя наборы факторов и оценки удобства и полезности для каждого заказа в рамках данных факторов. На входы нейронной сети подаются производственные факторы, а на выходе получаем коэффициенты прибыльности и удобства (рис. 1).

Данные, которые необходимы для обучения нейронной сети, хранятся в базе данных. Эти данные включают значения производственных факторов и соответствующие им экспертные оценки, а также экспертные значения оценки каждого производственного фактора с точки зрения его значимости. База данных для обучения нейронной будет заполняться данными, предоставленными частным предприятием, занимающимся изготовлением печатных плат.

В свою очередь, данные, которые понадобятся при работе системы поддержки принятия решений на предприятии, а именно значения производственных факторов рассматривающихся заказов или уже рассмотренных, другими словами, все данные, участвующие в работе СППР хранятся в специальной базе данных, которая и будет обеспечивать информационную поддержку системы поддержки принятия решений.

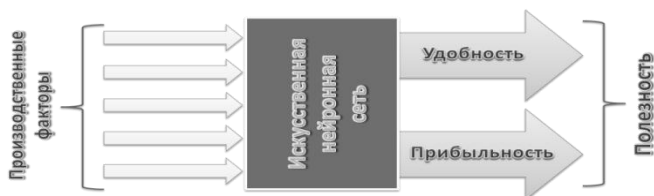


Рисунок 1 - Вычисление коэффициентов
удобности и прибыльности

Для того, чтобы выбрать наилучшую последовательность выполнения заказов, необходимо рассмотреть все альтернативы. Каждая альтернатива представляет собой одну конкретную последовательность выполнения заказов. Полезность каждой альтернативы определяется суммой полезностей всех заказов, входящих в данную последовательность. Также следует отметить, что полезность заказа не одинакова в различных альтернативах, т.к в фактор удобства входят факторы, которые зависят от очередности выполнения заказов. Это значит, что при каждом расчете полезности каждой альтернативы следует пересчитывать и полезности каждого заказа. Для того, чтобы определить оптимальную последовательность выполнения заказов, необходимо рассмотреть все возможные альтернативы, найдя в рамках каждой общий коэффициент удобства и прибыльности. После этого необходимо найти наиболее оптимальную альтернативу. Для данного процесса выбран метод минимальных расстояний [4]. Каждая альтернатива будет иметь 2 характеристики по 2 критериям – удобность и прибыльность, соответственно. В данном методе каждая альтернатива рассматривается как точка в 2-мерном пространстве, а ее характеристики по каждому критерию координата по соответствующей оси (рис. 2). Кроме точек-альтернатив (синие точки на рис. 2), на сформированной 2-мерной системе координат будет еще одна точка – Идеал (зеленая точка на рис. 2). Эта точка имеет координаты, соответствующие идеальным значениям по каждому из критериев, следовательно, является абсолютно оптимальной в рамках данного процесса.

Таким образом, расстояние между двумя точками является эквивалентом степени различности двух альтернатив одновременно по всем критериям. Это значит, что вычислив различность каждой точки-альтернативы от точки Идеал, можно определить порядок оптимальности альтернатив по следующей

логике: чем меньше альтернатива отличается от идеала, тем альтернатива оптимальнее, соответственно, чем меньше расстояние между точкой-

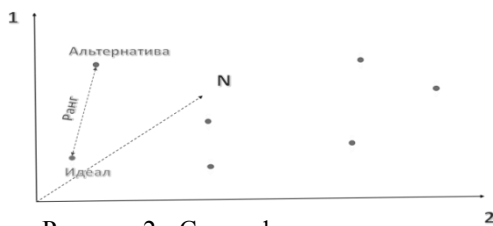


Рисунок 2 - Схема функционирования метода минимальных расстояний

альтернативой и точкой Идеал (фиолетовая пунктирная линия рис. 2), тем альтернатива оптимальнее. Следовательно, расстояние между данной точкой-альтернативой и точкой Идеал можно считать рангом данной альтернативы и рассчитывается по следующей формуле[4]:

$$r_{\text{альтернатива}} = \sqrt{(x_i^{\text{ideal}} - x_i^{\text{альтернатива}})^2 + (x_{i+1}^{\text{ideal}} - x_{i+1}^{\text{альтернатива}})^2 + \dots + (x_N^{\text{ideal}} - x_N^{\text{альтернатива}})^2} \quad (3)$$

Где, alternative- номер текущей рассматриваемой точки-альтернативы, ideal-точка Идеал, i -номер оси, $x_i^{\text{альтернатива}}$ – координата точки-альтернативы номер alternative по оси i , а $r_{\text{альтернатива}}$ - расстояние от текущей точки-альтернативы номер alternative до точки Идеал (ранг альтернативы).

Альтернатива с наименьшим рангом и будет наиболее оптимальной последовательностью выполнения заказов.

В случае функции оценки целесообразности выполнения одного заказа происходит анализ полученных коэффициентов удобства и полезности путем сравнения их с заранее определенными ограничениями, определяемыми руководством предприятия на основании личных предпочтений и корпоративных правил. Если получившиеся коэффициенты выше минимальных значений (ограничений), то выполнение данного заказа целесообразно, в противном случае – нет. Таким образом, полученные коэффициенты удобства и прибыльности являются критериями при оценке целесообразности выполнения данного заказа для данного предприятия:

- ЕСЛИ $\text{CoefU} \geq A$ И $\text{CoefP} \geq B$ ТО заказ следует выполнить;
- В остальных случаях – нет.

Здесь A – определенное руководством предприятия на основании личных предпочтений и корпоративных правил минимальное значение коэффициента удобства, а B – минимальное значение коэффициента прибыльности

В случае же функции выбора оптимальной последовательности выполнения заказов полученные для каждой альтернативной последовательности выполнения заказов пары коэффициентов будут использоваться в качестве экспертных оценок для ранжирования заказов-альтернатив методом минимальных расстояний, описанным выше. В итоге будет получена последовательность заказов, которая и будет оптимальной.

Выводы

Разработана модель системы поддержки принятия решений для малого предприятия, которая поможет улучшить организацию производства путем облегчения руководящей функции, снижения риска принятия неправильного производственного решения, что позволит уменьшить убытки и повысить прибыль предприятия. В разработанную модель СППР входят средства анализа и сбора данных, необходимых для принятия решения, которые позволят обеспечить информационную поддержку СППР, сам аппарат принятия решения, выполняющий, непосредственно, роль механизма, облегчающего руководящую функцию и дающий рекомендации относительно целесообразности выполнения заказа и построения оптимальной последовательности выполнения заказов, а также пользовательский интерфейс, который позволяет ввести параметры заказов и получить рекомендации о целесообразности выполнения отдельно взятого заказа или определить оптимальную последовательность выполнения заказов, полученных предприятием.

Список литературы

1. Сергей Гриняев. Нечеткая логика в системах управления [Текст]/ Сергей Гриняев: - Журнал «Компьютерра», Издательский дом «КОМПЬЮТЕРРА», 2002.
2. И. В. Матях, Е.О. Савкова, Корреляционный анализ методов определения весовых коэффициентов значимости для системы оценки социально-экономических показателей развития предприятия[Текст]/ И. В. Матях, Е.О. Савкова - ИНФОРМАТИКА И КИБЕРНЕТИКА №2, Донецк ДонНТУ, 2015.
3. Fundamental research [Electronic resource] / Интернет-ресурс. Режим доступа: <https://fundamental-research.ru/ru/article/view?id=39225> - Исаев И.В., Рогачев А.Ф., ФГБОУ ВО «Волгоградский государственный аграрный университет»
4. Fundamental research [Electronic resource] / Интернет-ресурс. Режим доступа: http://www.rema44.ru/resurs/study/sppr/sppr09_expert.pdf - Системы поддержки принятия решений: экспертные методы
5. Сафин Г.Г., Групповая экспертная оценка деятельности операторов одним из методов теории принятия решений - журнал [Текст] / Сафин Г.Г., Абдрахманов А.А., Великанов В.С., Усов И.Г., Савельев В.И., Мацко Е.Ю. «Современные проблемы науки и образования», 2015.

УДК 004.02

ИССЛЕДОВАНИЕ ОПТИМАЛЬНЫХ МЕТОДОВ ЗАГРУЗКИ ТРАНСПОРТА

С.Л. Хороших, Е.О. Савкова

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

E-mail: s.h.l@mail.ru

Аннотация

Хороших С.Л., Савкова Е.О. Исследование оптимальных методов загрузки транспорта. В работе рассмотрены: проблемы оптимизации загрузки транспорта, различные методы решения задачи о рюкзаке и её применение в решении данной проблемы. Проведен анализ различных видов задачи о рюкзаке и алгоритмов её решения.

Ключевые слова: транспорт, методы, система

Общая постановка проблемы

Логистика – основной элемент системы управления грузоперевозками. Основной задачей логистики является обеспечение потребителя продукцией в нужное время и определенное место при минимальных затратах на осуществление логистических операций и используемых производственных ресурсов.

Многие проблемы логистики могут быть решены с помощью задач о рюкзаке: погрузка, перевозка, контроль затрат и т.д. Рассмотрим простой пример проблемы управления. Каждому грузоперевозчику необходимо перевозить груз по оптимальному маршруту, при максимальной правильной загруженности. То есть для определения оптимального маршрута, сначала необходимо, правильно загрузить транспорт, для того что бы минимизировать время разгрузки на местах доставки, за счёт этого уменьшить общее время доставки и увеличить скорость доставки, а также оптимизировать весь маршрут. В данной статье проблема рассматривается со стороны оптимальной загрузки транспорта.

Грузы, как правило, поставляются с помощью грузовых автомобилей, самолетов или судов. Каждый груз имеет определенный вес и габариты, а грузовой транспорт имеет неизменную грузоподъемность. Поэтому перевозчики стараются максимизировать свою прибыль путем эффективной загрузки и доставки максимального веса в фиксированном объеме. Такая проблема может быть решена с помощью задачи ограничения и оптимизации, а именно с помощью задачи о рюкзаке.

Существует несколько разновидностей задачи о рюкзаке, отличия между которыми заключаются в ограничениях, наложенных на рюкзак, предметы или их выбор, для решения поставленной задачи, подходит:

1. *Неограниченный рюкзак* – каждый предмет можно брать сколько угодно раз;
2. *Непрерывный рюкзак* – возможно брать любую дробную часть от предмета;
3. *Рюкзак с мульти выбором* – есть несколько классов предметов, из которых можно брать одного представителя, причём некоторые вещи имеют больший приоритет, чем другие;
4. *Мультипликативный рюкзак* – есть несколько рюкзаков с различной вместимостью;
5. *Многомерный рюкзак* – есть более одного ограничения на рюкзак.

Методы решения

Задача о рюкзаке [3] – одна из задач комбинаторной оптимизации. Классическая задача о ранце известна очень давно. Суть задачи: имеется набор из N предметов, каждый предмет имеет массу W_i и полезность V_i , $i = (1, 2, \dots, N)$, требуется собрать набор с максимальной полезностью таким образом, чтобы он имел вес не больше W , где W – вместимость рюкзака. Традиционно полагают что W_i, V_i, W, P – целые неотрицательные числа, где P – стоимость предмета. Но встречаются и другие постановки, условия в которых могут отличаться. Возможны следующие вариации задачи:

Каждый предмет можно брать только один раз. Пусть задано конечное множество предметов $Q = \{q_1, q_2, \dots, q_n\}$, для каждого $q \in Q$, определена стоимость P_i и вес W_i , тогда нужно максимизировать $\sum_{i=1}^N p_i * x_i$, при ограничениях $\sum_{i=1}^N w_i * x_i \leq W$, где W – вместимость рюкзака, а $x_i = 1$, если предмет взят для загрузки и $x_i = 0$ если не взят. Если на размер рюкзака имеется только одно ограничение, то задача называется одномерной, в противном случае – многомерной.

Каждый предмет можно брать m раз. Формализация аналогична, разница лишь в том, что x_i принимает значения на интервале $(0 \dots m)$.

Каждый предмет можно брать неограниченное количество раз. Очевидно, что x_i лежит в диапазоне $(0 \dots \lfloor W/w_i \rfloor)$ квадратные скобки означают целую часть числа.

Если же значения весов и цен предметов не целые числа, такая задача будет называться непрерывной задачей о рюкзаке, если же числа целые, то соответственно дискретной. Например, если мы имеем дело с золотыми слитками, мы не можем их делить – это дискретная задача, а если с золотым песком, то это непрерывная задача о рюкзаке.

Учитывая сферу применения логистики грузоперевозок, можно выделить различные ограничения:

1. Необходимость загрузить груз с максимальной стоимостью, состав груза не интересует.
2. Груз ограниченного веса
3. Груз ограниченных габаритов
4. Груз с ограниченным сроком хранения
5. Некоторые предметы должны обязательно быть уложены в рюкзак (имеют приоритет).

Анализ методов решения

Существует ряд методов, позволяющих находить решение задачи о рюкзаке [1]. В данной работе рассматривались следующие методы (для всех приведенных выше постановок задачи):

1. Полный перебор
2. Метод ветвей и границ
3. Жадный алгоритм
4. Генетический алгоритм
5. Динамическое программирование

Поскольку метод полного перебора очень трудоемкий, то размер рюкзака и количество предметов при его использовании должны быть ограничены. Следовательно, этот метод неприменим для неограниченного и непрерывного рюкзаков. При решении перебором задач мультипликативного и ограниченного рюкзаков необходимо уменьшить количество предметов, чтобы решение было найдено.

Очевидным недостатком алгоритма метода ветвей и границ при решении задач большой размерности является необходимость перебрать слишком большое количество вариантов перед тем, как будет найден оптимальный. Несомненно, метод ветвей повторяет перебор. Однако, благодаря отсечению "бесполезных" решений, может самостоятельно ограничить бесконечное пространство решений. Поэтому его можно применить для неограниченного рюкзака, но нельзя применить для непрерывного рюкзака.

Жадный алгоритм является однокритериальным алгоритмом, поэтому он неприменим для многомерного рюкзака. Однако следует отметить, что для задачи непрерывного рюкзака именно этот алгоритм находит наиболее оптимальное решение.

Генетический алгоритм является одним из самых быстрых алгоритмов. Неограничением этого алгоритма является то, что его хромосомы кодируются дискретно. Соответственно, его нельзя использовать для решения непрерывных и бесконечных задач. Недостатком генетического алгоритма также является то, что он не гарантирует нахождения оптимального решения в некоторых ситуациях.

Метод динамического программирования часто помогает эффективно решить задачу. Идея этого метода состоит в сведении исходной задачи к

решению некоторых ее подзадач «меньшего размера» и использовании табличной техники для сохранения уже найденных ответов. Решение подзадач при этом происходит в порядке возрастания их размеров — от меньших к большим. Преимущество динамического программирования заключается в том, что раз уж какая-то подзадача решена, ее ответ где-то хранится и никогда не вычисляется заново.

Вывод

В настоящей статье была исследована одна из проблем логистики — оптимальная загрузка транспорта, с помощью решения задачи о рюкзаке. Разбирая область применения задачи, выясняются некоторое количество ограничений, поэтому задача является многомерной. Если требуется найти точное решение, то конечно нужно использовать точные методы, при небольшом наборе входных данных, подойдет перебор или метод ветвей и границ в силу простоты реализации, при больших, следует использовать динамическое программирование. Если же точность решения не так важна, или входные данные таковы, что ни один из точных методов не работоспособен, остается применять только приближенные алгоритмы. Но остается возможность комбинирования различных методов для ускорения, или даже применение каких-либо “уловок” для конкретного случая. Безусловно, оптимальная загрузка транспорта очень важна с точки зрения ее приложения в реальной жизни, она помогает получать большую прибыль, оптимизировать маршрут грузоперевозок, а также сокращать расходы.

Список литературы

1. Замкова Л.И. Булева двухкритериальная задача о рюкзаке / Л.И. Замкова // Известия Южного федерального университета. Технические науки. — 2009. — № 4. — С. 201 – 204.
2. Викизнание — большая универсальная гипертекстовая электронная энциклопедия / Интернет-ресурс. —режим доступа : URL: http://www.wikiznanie.ru/wikipedia/index.php/Задача_об_одномерной_оптимальной_упаковке
3. Википедия — свободная энциклопедия/ Интернет-ресурс. —режим доступа : URL: https://ru.wikipedia.org/wiki/Задача_о_ранце

УДК 004.048:622

МЕТАЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ ФОРМИРОВАНИЯ ОПТИМАЛЬНОГО ГРАФИКА ПРОХОЖДЕНИЯ ЛЕЧЕБНО- ОЗДОРОВИТЕЛЬНЫХ ПРОЦЕДУР

Задорожная Е.Г., Савкова Е.О.

Донецкий национальный технический университет, г. Донецк

кафедра автоматизированных систем управления

e-mail: ekaterina.zadorozhnaya.95@gmail.com

Аннотация

Задорожная Е.Г., Савкова Е.О. Метаэвристические алгоритмы формирования оптимального графика прохождения лечебно-оздоровительных процедур. В данной работе описаны особенности составления данного вида графика и разработана динамическая модель процесса. В качестве методов решения приведены такие метаэвристические алгоритмы, как генетический и муравьиный.

Ключевые слова: оптимизация, график, динамическая модель, сеть Петри, генетический алгоритм, муравьиный алгоритм.

Введение

Главным фактором успешной работы лечебно-оздоровительных учреждений на сегодняшний день, является повышение качества и скорости предоставления услуг. Все решения должны приниматься оперативно и быть оптимальными. Задачи оперативно-календарного планирования решались неоднократно и различными методами, однако, достаточно сложно определить качество получаемых расписаний. К тому же, в рамках данной предметной области, процесс формирования оптимального графика прохождения лечебно-оздоровительных процедур специфических особенностей, не характерных другим отраслям производства. Данный факт объясняет актуальность разработки оптимального алгоритма прохождения процедур.

Основной целью разработки алгоритма является повышение эффективности функционирования лечебного-отделения. Это позволит достигнуть минимизации времени простоя оборудования и времени ожидания обслуживания клиента, что в результате приведёт к увеличению количества обслуживаемых клиентов.

Постановка задачи

Для формирования графика прохождения лечебно-оздоровительных процедур необходимо иметь определённый набор входных данных. Санаторно-курортный комплекс рассчитан на определённое количество

пациентов (p_i ($i = 1, 2, \dots, n$)). Лечащий врач пациента, в соответствии с его диагнозом, назначает перечень обязательных процедур (исходя из всех возможных: $rgoc_i$ ($i = 1, 2, \dots, n$)) периодичность сеансов и их количество. Пациент также имеет право на посещение дополнительных процедур общего назначения, согласовав это с лечащим врачом.

Процедурные кабинеты работают по графику и имеют определенное количество оборудования (eq_i ($i = 1, 2, \dots, n$)). При этом каждое оборудование предназначено для определённой процедуры или вида процедур. Проведение каждой процедуры занимает какой-то промежуток времени. На основе данной информации перечня, составляется план посещения процедурных кабинетов и получения лечебно-оздоровительных услуг.

Для реализации поставленной задачи необходимо составить оптимальный график (расписание) с учетом ряда ограничений:

1. По срокам предоставления процедур (1):

$$T_{пл} \geq T_{ф}, \quad (1)$$

где $T_{пл}$ -скор выполнения запланированных процедур $rgoc_i$;

$T_{ф}$ -фактическое количество проведенных процедур $rgoc_i$.

2. По количеству проведения процедур (2):

$$N_{пл} = N_{ф} \quad (2)$$

где $N_{пл}$ - количество запланированных процедур $rgoc_i$;

$N_{ф}$ -фактическое количество проведенных процедур $rgoc_i$.

3. По времени непрерывной работы оборудования (3):

$$N_{пр} * T_{пр} \leq T_{пр}, \quad (3)$$

где $N_{пр}$ - количество проведенных процедур $rgoc_i$ за день, на оборудовании eq_i ;

$T_{пр}$ -время проведения (длительность) процедуры $rgoc_i$;

$T_{пр}$ – время непрерывной работы оборудования eq_i .

4. По времени простоя оборудования (4):

$$0 \leq \Delta T_{ожид об} \leq T_{прост}, \quad (4)$$

где $\Delta T_{ожид об}$ – разница во времени между окончанием процедуры $rgoc_i$ и началом следующей на оборудовании eq_i ;

$T_{прост}$ – допустимое время простоя оборудования eq_i .

5. По временному разбросу между процедурами (5):

$$0 \leq \Delta T_{ожид п} \leq T_{св}, \quad (5)$$

где $\Delta T_{ожид п}$ – разница во времени между окончанием процедуры $rgoc_i$ и началом следующей, назначенной пациенту p_i ;

$T_{св}$ – свободное время между процедурами.

Таким образом, задача сводится к тому, чтобы разработать алгоритм, который удовлетворяет всем сформулированным ранее условиям и ограничениям.

Динамическая модель

Для построения динамической модели процесса прохождения лечебно-оздоровительных процедур была выбрана сеть Петри. Это математический аппарат для моделирования динамических дискретных систем. Процесс функционирования сети Петри может быть наглядно представлен графом достижимых маркировок. Они позволяют определить общую структуру и принципы работы системы. Сети-Петри рассматриваются как вспомогательный инструмент анализа систем. Проблемы анализа указывают на изъяны в проекте [1].

Сеть Петри характеризуется входной и выходной функцией, а так множеством позиций (мест) и множеством переходов. Переход срабатывает, когда метка из входной позиции перехода перемещается в выходную позицию. Запуски могут осуществляться до тех пор, пока существует хотя бы один разрешенный переход. Когда ни одного разрешенного перехода не остается, то выполнение прекращается [2].

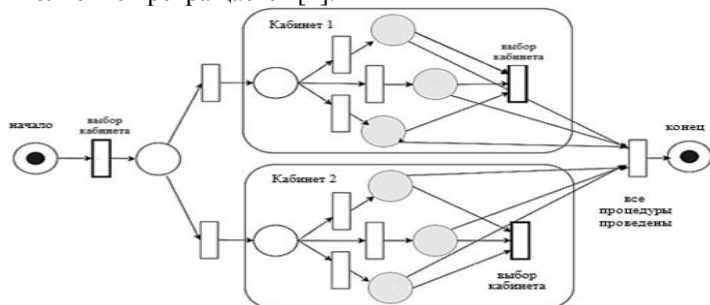


Рисунок 1 – Динамическая модель процесса

В рамках решаемой проблемы сети Петри являются удобным инструментом, для создания высокоуровневой модели работы процесса. На рис. 1 представлена динамическая модель процесса прохождения лечебно-оздоровительных процедур в виде сети Петри второго рода, с иерархической композицией объектов. Имея информацию о назначенной процедуре необходимо понимать, на каком типе оборудования её можно проводить (выбор кабинета). Далее следует выбрать свободное оборудование в данный момент времени. Для наглядности оборудование отмечено в виде закрашенного круга. Это связано с тем, что оборудованием в данный момент времени может пользоваться лишь один клиент. После предоставления процедуры клиенту, он возвращается к моменту выбора кабинета, для следующей процедуры. Это будет длиться до тех пор, пока перечень процедур не закончится. В случае, если нет свободного оборудования, клиент дожидается его освобождения.

Анализ методов решения

Основой для разработки соответствующего заданным требованиям алгоритма формирования графика прохождения лечебно-оздоровительных

процедур могут стать метаэвристики, которые хорошо себя показали при решении оптимизационных, комбинаторных, а также других видов задач. Алгоритм метаэвристик основан на случайном поиске возможных решений задачи, оптимальных или близких к оптимальным, пока не будет выполнено некое условие или достигнуто заданное число итераций[3].

Проанализируем возможность решения задачи формирования оптимального графика прохождения лечебно-оздоровительных процедур с помощью метода муравьиных колоний и генетических алгоритмов.

Муравьиный алгоритм - алгоритм для нахождения приближённых решений задачи коммивояжёра, а также решения аналогичных задач поиска маршрутов на графах[4].

В виде муравья в данном случае можно рассматривать пациента, тогда вершинами графа будет выступать оборудование. Каждая вершина содержит информацию о процедурах, которые могут проводиться на данном типе оборудования, и массив промежутков времени. В основе муравьиного алгоритма лежит поведение муравьиной колонии – маркировка более удачных путей большим количеством феромона. Когда муравей находит подходящее назначенной процедуре оборудование, то занимает интервал времени на её проведение. Затем, он возвращается в начало, оставляя за собой след из феромона.

Генетический алгоритм - это эвристический алгоритм поиска путём последовательного подбора, комбинирования и вариации искомым параметров с использованием механизмов, напоминающих биологическую эволюцию [3].

Решение задачи на основе ГП можно представить следующей последовательностью действий:

1. Установка параметров эволюции. Инициализация начальной популяции для одного пациента. Особь представлена в виде двух хромосом, где информация 1-ой - оборудование, а 2-ой – сеансы (промежутки времени) [5]. В обеих хромосомах одинаковое количество генов равно p_i , равно количеству назначенных процедур пациенту. Следовательно i гену 1-ой хромосомы соответствует оборудование (из числа допустимых), выбранное для предоставления процедуры, а i гену 2-ой хромосомы - время её проведения (рис. 2).

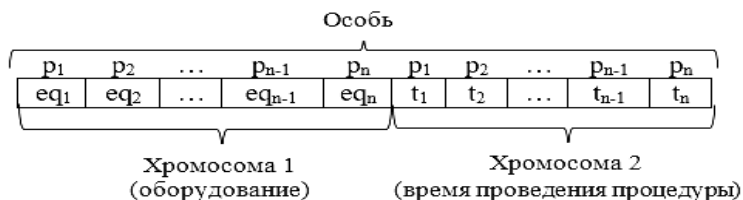


Рисунок 2 – Структура особи

2. Оценка особей, входящих в популяцию. Отбор родителей.

3. Создание потомков выбранных пар родителей – выполнение оператор кроссинговера. Для каждой пары отобранных особей случайным образом выбираются позиции гена G_1 и G_2 и производится обмен участками генетического кода между соответствующими хромосомами родительских особей[5].

4. Мутация новых особей.

5. Расширение популяции новыми порожденными особями. Сокращение расширенной популяции до исходного размера.

6. Обнаружена сходимости или достигнуто максимальное количество итераций. Если критерий останова алгоритма выполнен, то выбор лучшей особи в конечной популяции – результат работы алгоритма. Иначе переход на шаг 2.

7.Результирующую особь необходимо проверить на существование. Последовательность прохождения процедур проверяем на ранее разработанной динамической модели. Для хранения информации о загрузке оборудования необходима матрица $plan_{ij}$, где i - это перечень всего имеющегося оборудования, а j - интервалы времени. Полученные результаты заносим в матрицу.

Выводы

В рамках данной работы был проведён анализ работы процесса формирования графика прохождения процедур на основе динамической модели. Также, были проанализированы основные алгоритмы составления расписаний: муравьиный и генетический.

Муравьиный алгоритмы, не смотря на свою динамичность, использование памяти колоний и сходимости к оптимальному решению имеет ряд недостатков. Графовое представление данной задачи не является оптимальным. Результаты работы методы сильно зависят от начальных параметров поиска.

Генетический алгоритм является более гибким в процессе поиска решения, поскольку использует несколько точек поискового пространства, а не переходит от точки к точке. Для решения поставленной задачи предложено использовать модифицированный генетический алгоритм, который позволит получать оптимальные решения с учетом особенностей лечебно-оздоровительных учреждений.

Список литературы

1. Tadao Murata // *Petri Nets: Properties, Analysis and Applications* // Department of Electrical Engineering and Computer Science, University of Illinois, Chicago, IL 60680, USA

2. Branimir Sigl, Marin Golub, Vedran Mornar // *Solving Timetable Scheduling Problem by Using Genetic Algorithms* // Faculty of Electrical Engineering and Computing, University of Zagreb Unska 3, Zagreb, Croatia

3. Конькова И.С. // Использование генетического алгоритма в задаче оптимизации расписания вуза.

4. Светличная В.А., Червинская Н.В., Хаустова Д.А. // *Использование алгоритмов муравьиных колоний при решении задач оптимизации календарного планирования.* // ДонНТУ, г. Донецк

5. С. Кабальнов, Л.И. Шехтман, Г.Ф. Низамова, Н.А. Земченкова // *Композиционный генетический алгоритм составления расписания учебных занятий.* // УГАТУ, г. Уфа

УДК 004.62

АВТОМАТИЗАЦИЯ РАБОТЫ ПРИЕМНОЙ КОМИССИИ УЧЕБНОГО ЗАВЕДЕНИЯ ПРИ ПРОВЕДЕНИИ ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ

К.Д. Гимадеев (4-й курс, каф. КИ), О.Ю. Чередникова
Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
disep@mail.ru

Аннотация

Гимадеев К.Д., Чередникова О.Ю. Автоматизация работы приемной комиссии учебного заведения при проведении вступительных испытаний. *Разработано программное обеспечение, обеспечивающее ввод оценок и формирование экзаменационных ведомостей для автоматизации процесса приемной комиссии ДонНТУ.*

Ключевые слова: база данных, автоматизация, программное обеспечение.

Постановка проблемы. Приемная комиссия вуза решает широкий круг задач, таких как учет абитуриентов и результатов сдачи экзаменов, зачисление абитуриентов, формирование отчетности. Ее работа является одной из важнейших составляющих деятельности учебной части вуза. Обычно в состав приемной комиссии входит не менее десяти работников, которые принимают документы у абитуриентов и занимаются составлением отчетности. В большинстве вузов все эти задачи выполняются вручную.

Цель статьи – разработать программное обеспечение для ввода оценок и формирования экзаменационных ведомостей для автоматизации процесса приемной комиссии.

Выбор инструментальных программных средств разработки. Для написания программы используется язык программирования C# 6.0. Данный выбор обусловлен такими преимуществами данного языка программирования, как скорость разработки, большой ассортимент сторонних библиотек, простота и лаконичность синтаксиса[1]. Разработка осуществлялась в среде разработки Microsoft Visual Studio 2015. В качестве СУБД используется Microsoft SQL Server 2008[2]. Для разработки использовался Entity Framework. Entity Framework (EF) — это объектно-реляционный модуль сопоставления, позволяющий разработчикам .NET работать с реляционными данными с помощью объектов, специализированных для доменов[3]. Это устраняет необходимость в написании большей части кода для доступа к данным, который обычно требуется разработчикам.

Разработка. Для реализации поставленной задачи создадим следующие таблицы:

- rasp_sobesed – таблица с информацией о расписании собеседований;
- rasp_exam_pk – таблица с информацией о расписании экзаменов;
- rasp_exam_lang_pk – таблица с информацией о расписании экзаменов по иностранным языкам;
- ank_ved – таблица с информацией о сформированных ведомостях;
- ank_sobesed – таблица с информацией об оценках за собеседование;
- ank_exam_pk – таблица с информацией об оценках за экзамены;
- ank_exam_lang – таблица с информацией об оценках за экзамены по иностранным языкам;
- in_attr – таблица с информацией о студентах.

Программа разрабатывается с использованием паттерна MVVM. Создадим отдельные формы для каждой операции:

- Окно авторизации;
- Главное окно программы;
- Диалог открытия ведомости;
- Диалог создания новой ведомости;

В окне авторизации расположим поля ввода данных, необходимых для авторизации в базе данных. Подключение осуществляется с помощью класса MySqlConnection, который находится в пространстве имен MySql.Data.MySqlClient. При успешной авторизации откроем форму главного окна программы. Тут расположим элементы управления, необходимые для заполнения ведомости, а так же элементы управления, необходимые для работы с программой, а именно: кнопка открытия формы выбора ведомости; кнопка вызова диалога создания новой ведомости; кнопки утверждения и снятия утверждения ведомости (их сделаем доступными только при условии, если ведомость открыта); кнопка формирования ведомости в .doc файл; кнопка выхода из программы.

При вызове формы открытия ведомости будем получать список ведомостей из таблицы `ank_ved` и выводить их на форму в элемент управления `DataGrid`.

При открытии ведомости происходит получение списка абитуриентов, относящихся к данной ведомости с помощью метода `GetAbits()`, и получения списка оценок с помощью метода `GetMarkList()`. На этапе ввода оценки выполняется проверка: присутствует ли студент с заданным `id_foto` в данной ведомости. За это отвечает метод `isExist()`. В качестве входного параметра этого метода указывается `id_foto` абитуриента, возвращаемое значение – `bool`: `true` – если абитуриент присутствует, иначе – `false`.

При печати ведомости данные о ведомости, абитуриентах и оценках будут заноситься в заранее подготовленный файл шаблона документа. Работа с Word документами реализуется с помощью классов `Application` и `_Document`, расположенных в пространстве имен `Microsoft.Office.Interop.Word` библиотеки `Microsoft Word 11.0 Object Library`. После окончания формирования документ будет автоматически запущен.

Анализ работы ПО. Программа устроена следующим образом: при запуске программы перед пользователем появляется окно авторизации, где пользователь должен ввести данные для авторизации – логин и пароль. Так же пользователь может самостоятельно указать адрес сервера, на котором хранится база данных, и, собственно, указать, с какой именно базой данных предстоит работать.

В случае успешной авторизации перед пользователем появляется основное окно программы. Из него пользователь может создать новую ведомость или же открыть уже существующую. Так же в этом окне присутствует интерфейс отображения информации об открытой ведомости, а так же интерфейс ввода оценок для абитуриентов, относящихся к данной ведомости. Когда ведомость открыта, пользователь имеет возможность вводить список оценок, утверждать или снимать признак утверждения ведомости, а так же формировать ведомость в `.doc` файл.

Изображение главного экрана программы представлено на рисунке 1.

Сервер=192.168.101.80, База данных=a2016test, -Формирование экзаменационных ведомостей ...

№ ведомости: 1 Номер фото: 35

Предмет: Собеседование день 1 (11.07.2016) Оценка: 125

Тип испытания: Собеседование Шифр работы:

Абитуриентов: 28

ФИО абитуриента: Лакей1 Роман Олегович

ID Foto	ФИО	Оценка
9	Подлесный1 Дмитрий Игоревич	145
13	Пригода1 Максим Сергеевич	126
14	Харченко1 Артур Вадимович	124
35	Лакей1 Роман Олегович	125
44	Ляцук1 Максим Юрьевич	125
58	Ружило1 Надежда Петровна	124
90	Богдан1 Игнат Николаевич	136
210	Чепурко1 Олег Григорьевич	125
211	Подгорный1 Александр Сергеевич	126
309	Гаврилюк1 Вадим Петрович	135
384	Яструбенский1 Станислав Витальевич	124
436	Бубнов1 Вячеслав Сергеевич	130
439	Чевычалов1 Владимир Владимирович	150
444	Мишукова1 Кристина Александровна	128

Рисунок 1. Главное окно программы

При создании ведомости перед пользователем появляется окно, где он может выбрать тип испытания и конкретное испытание.

При открытии ведомости перед пользователем появляется окно с возможностью выбора уже созданных ведомостей с информацией о них (например, количество абитуриентов в данной ведомости и признак ее утверждения). Имеется возможность поиска и сортировки ведомости. Интерфейс окна открытия ведомости приведен на рисунке 2.

Выбор ведомости

№ ведомости: Дата экзамена: 09.09.2016 15

Тип испытания:

№	Испытание	Дата	Абитуриентов	Утверждена
1	Собеседование день 1 (11.07.2016)	7/11/2016 9:00:00 AM	28	Да
1	Собеседование день 2 (12.07.2016)	7/12/2016 9:00:00 AM	38	Да
1	Физ. Подготовка Группа 1	7/12/2016 12:00:00 AM	38	Да
1	Собеседование день 3 (13.07.2016)	7/13/2016 9:00:00 AM	35	Да
1	27.03.03 Системный анализ и управление (7/14/2016 10:00:00 AM	1	Да
1	21.03.02 Землеустройство и кадастры (13.0	7/13/2016 10:00:00 AM	2	Да
1	Собеседование день 4 (14.07.2016)	7/14/2016 9:00:00 AM	17	Да
1	Собеседование день 5 (15.07.2016)	7/15/2016 9:00:00 AM	14	Да
1	27.03.02 Управление качеством (15.07.2016	7/15/2016 10:00:00 AM	2	Да
1	23.04.02 Наземные транспортно-технологи	7/16/2016 9:00:00 AM	0	Нет
1	21.04.02 Землеустройство и кадастры (16.0	7/16/2016 1:00:00 PM	8	Да
1	Собеседование день 6 (16.07.2016)	7/16/2016 9:00:00 AM	25	Да
1	27.04.03 Системный анализ и управление (7/17/2016 1:00:00 PM	1	Да
1	11.04.02 Инфокоммуникационные технолс	7/17/2016 9:00:00 AM	25	Да
1	Собеседование день 7 (18.07.2016)	7/18/2016 9:00:00 AM	1	Да
1	05.04.03 Картография и геоинформатика (1	7/18/2016 9:00:00 AM	2	Да
1	47.04.01 Философия (18.07.2016) смена 2	7/18/2016 1:00:00 PM	1	Да
1	Французский язык (20.07.2016) смена 1	7/20/2016 9:00:00 AM	5	Да
1	15.04.06 Мехатроника и робототехника (1	7/19/2016 9:00:00 AM	3	Да
1	Английский язык (20.07.2016) смена 2	7/20/2016 1:00:00 PM	29	Да
2	Английский язык (20.07.2016) смена 2	7/20/2016 1:00:00 PM	34	Да
2	38.04.05 Бизнес-информатика (18.07.2016)	7/18/2016 9:00:00 AM	21	Да

Рисунок 2. Диалог открытия ведомости

Когда ведомость открыта и не утверждена, пользователь имеет возможность редактировать оценки и вводить новые. Оценки вводятся с помощью специального интерфейса в окне главной программы. Для ввода оценки нужно ввести ид фото абитуриента и, собственно, саму оценку. Если ведомость – экзаменационная, то так же вводится шифр работы. После ввода ид фото абитуриента справа отображается ФИО абитуриента, что помогает убедиться в правильности ввода. Так же реализован механизм защиты: при открытии ведомости из базы подгружается список ID фотографий абитуриентов. Введенный ID фото проверяется на наличие в списке. Если такого нет – выдается сообщение об ошибке.

После утверждения ведомости ввод и удаление оценок не доступен. Утвердить и снять утверждение возможно в любой момент.

При печати ведомости формируется документ, в который заносится следующая информация:

- Номер ведомости;
- Образовательный уровень (Бакалавр/Магистр);

- Предмет, по которому проводится вступительное испытание;
- Дата проведения испытания;
- Время начала и окончания испытания;
- Шифр абитуриента;
- ФИО абитуриента;
- Количество баллов, набранных абитуриентом (цифрами и прописью).

Эффективность внедрения ПО. Для начала рассмотрим что-же понимается под эффективностью. В данном случае эффективность это соотношение между полученными в результате внедрения программы преимуществами и силами, затраченными на внедрение. Очевидно, что документооборот во время приемной комиссии колоссален, а следовательно обработка информации весьма затратна по времени. Внедрение системы автоматизации позволяет значительно ускорить процесс работы сотрудников, облегчает поиск необходимой информации, а также обеспечить более высокую сохранность данных и уменьшить количество «механических» ошибок. Появляется возможность редактирования данных в случае возникновения ошибки либо появления более актуальной информации. Так как приемная комиссия работает каждый год, следовательно, средства, облегчающие работу персонала, оправдывают любые силы, затраченные на их разработку.

Выводы. В ходе проделанной работы была разработана программа, обеспечивающая реализацию модуля приемной комиссии «Ввод оценок». В ходе дальнейших исследований планируется оптимизировать структуру базы данных, разработать модуль автоматизации составления расписания экзаменов, реализовать функции зачисления.

Список литературы

1. Выбор между C++ и C# // Habrahabr [электронный ресурс]. [2015]. URL: <https://habrahabr.ru/post/262461/>
2. Обзор платформы Microsoft SQL Server 2008 [электронный ресурс] // Open Systems Publications: [сайт]. [2010]. URL: https://www.osp.ru/resources/izones/mssql/platform/platform_1.html
3. Entity Framework // MSDN [электронный ресурс]. [2013]. URL: [https://msdn.microsoft.com/ru-ru/library/gg696172\(v=vs.103\).aspx](https://msdn.microsoft.com/ru-ru/library/gg696172(v=vs.103).aspx)

УДК 004.048

РАЗРАБОТКА МЕТОДИКИ ОЦЕНКИ КОМПЕТЕНТНОСТИ СОТРУДНИКОВ ИТ-ОТДЕЛА С ПОМОЩЬЮ СИСТЕМ НЕЧЁТКОГО ВЫВОДА

А.В. Выклюк, С.Ю. Землянская

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
E-mail: vyknasty95@gmail.com

Аннотация

Выклюк А.В., Землянская С.Ю. Разработка методики оценки компетентности сотрудников ИТ-отдела с помощью систем нечёткого вывода. В статье описаны основные этапы оценки компетентности сотрудников, приведен алгоритм и рассмотрен поэтапный принцип работы систем нечёткого вывода для оценки компетентности персонала.

***Ключевые слова:** компетентность, оценка персонала, система нечёткого вывода, лингвистическая переменная, интеллектуальная система.*

Постановка проблемы. В управлении персоналом проблемы оценки остаются не до конца проработанными как в теоретическом, так и в практическом аспекте. В современных условиях актуальность проблемы оценки персонала чрезвычайно высока. Эффективно сформированная и внедренная в действие она является необходимым условием функционирования и развития предприятий любой сферы деятельности.

Для повышения конкурентоспособности персонала предприятия необходимо использование методов его оценки, что позволит получить конкурентоспособного сотрудника, который повышает свой образовательно-квалификационный уровень.

На современном рынке труда возрастает потребность в высококвалифицированных специалистах в области информационных технологий. Для проведения аттестации ИТ-персонала следует применять сложные комбинированные методики для того, чтобы составить полное представление о том, насколько специалист соответствует должности и как правильно построить его дальнейшее развитие в структуре компании. Использование подобных методов оценки персонала на предприятиях позволит руководству раскрыть трудовой потенциал самого работника и повысить его личную и профессиональную эффективность.

Учитывая вышеуказанное, разработка эффективной методики оценки компетентности сотрудников будет чрезвычайно полезна для построения интеллектуальной экспертной системы оценивания персонала.

Цель статьи– провести анализ и рассмотреть особенности процесса оценки компетентности сотрудников при помощи нечётких систем вывода. Разработать алгоритм проверки соответствия персонала занимаемым должностям.

Постановка задачи исследования. В процессе исследований предполагается решение таких задач:

- раскрытие последовательности процесса оценки персонала;
- выбор эффективного метода оценивания;
- определение соответствия сотрудников занимаемым должностям.

Процесс оценки компетентности персонала происходит следующим образом:

1. Формируется набор ключевых компетенций для данного отдела/предприятия.
2. Выбирается методика проведения тестового оценивания.
3. Составляется шкала оценивания.
4. Определяются критерии значимости каждого вида тестирования путём назначения весовых коэффициентов.
5. Проводится тестирование персонала.
6. Определяется степень соответствия сотрудника определённой должности на основании результатов оценивания.
7. Делаются соответствующие выводы и даются рекомендации.

Модели и методы в задачах управления персоналом. Современные математические методы и модели открывают новые возможности для оценивания сотрудников на предприятии и повышения эффективности методов управления персоналом компании.

Методы математического моделирования:

- оптимальное планирование процессов карьерного роста и обновления состава персонала;
 - исследование стохастических процессов карьерного роста в компании методами теории массового обслуживания;
 - разработка эффективных механизмов мотивации;
 - разработка оптимальных схем распределения ресурсов на удовлетворение разноуровневых потребностей персонала;
 - разработка оптимальных стратегий профессиональной адаптации.
- Нейросетевые модели и методы:
- прогнозирование успешности персонала в компании в процессе найма;
 - оценка результативности труда управленческого персонала;
 - выявление предпочтений агентов

Экспертные методы и технологии:

- проведение экспертизы управленческих решений в области управления персоналом;
- формирование систем деловой оценки персонала;
- выявление предпочтений агентов;
- выявление сильных и слабых сторон системы управления персоналом;
- формирование компетентностных моделей специалистов.

На основании описания методов и сферы их применения, можно сделать вывод, что для решения поставленной задачи наиболее подходят экспертные методы и технологии.

Для эффективного анализа компетентности необходимо применять подход, допускающий нечёткости и частичные истины. Таким требованиям соответствуют экспертные системы на основе нечёткого вывода (рис.1).



Рисунок 1 – Алгоритм работы системы нечёткого логического вывода

Этот подход имеет три отличительные черты:

1. В нём используются «лингвистические» переменные вместо числовых переменных или в дополнение к ним.
2. Простые отношения между переменными описываются с помощью нечетких высказываний.
3. Сложные отношения описываются нечёткими алгоритмами.

Лингвистической переменной называется набор $\langle \beta, T, X, G, M \rangle$, где:
 β — имя лингвистической переменной;

T — множество её значений (терм-множество), представляющее имена нечётких переменных;

X — универсум нечетких переменных, область определения;

G — синтаксическая процедура образования новых термов;

M — семантическая процедура, формирующая нечеткие множества для каждого термина данной лингвистической переменной.

Шкала оценивания каждой компетенции задается с помощью лингвистической переменной $K=[K_1, K_2, \dots, K_t]$, $t=5$.

β — уровень компетентности;

$T = \{ \text{“Низкий”, “Ниже среднего”, “Средний”, “Выше среднего”, “Высокий”} \}$;

$X = [0, 1]$;

Компетентность в области информационных технологий определена как показатель готовности сотрудника к профессиональной деятельности в данной сфере. Уровень сформированности данного показателя позволит судить о степени соответствия кандидата определённой должности. Шкала распределения уровней компетентности по должностям представлена в табл. 1.

Таблица 1 – Соответствие уровней компетентности должностям

Уровень компетентности	Значение общего показателя	Соответствующая должность
Высокий	0,81—1,0	Руководитель отдела/департамента.
Выше среднего	0,61—0,8	Ведущий специалист (программист, веб-программист/дизайнер).
Средний	0,41—0,6	Администратор (системный, БД).
Ниже среднего	0,21—0,4	Инженер (службы технической поддержки, сетевой).
Низкий	0—0,2	Технический писатель (копирайтер, контент-менеджер)

На рисунке 2 представлен вид треугольной функции принадлежности для уровня компетентности.

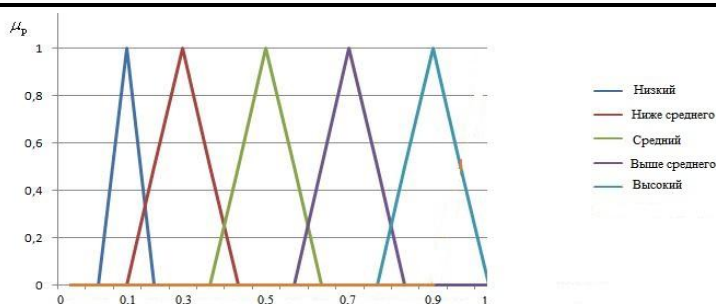


Рисунок 2 – График функции принадлежности для ЛП «Уровень компетентности»

Итоговая функция принадлежности будет иметь вид:

$$\mu_{\Sigma} = \left\{ \frac{1}{n} \sum_{l=1}^n X_{m1} \times W_{i1} / K_1, \frac{1}{n} \sum_{l=1}^n X_{m2} \times W_{i2} / K_2, \dots, \frac{1}{n} \sum_{l=1}^n X_{ml} \times W_{ip} / K_l \right\}$$

В соответствии с профессиональными стандартами сотрудникам ИТ-отдела необходимо обладать набором компетенций (качеств, навыков и умений) на определенном уровне, в соответствии с занимаемой должностью. Однако, другие качества у сотрудников могут быть менее выражены.

При составлении списка компетенций необходимо задавать значения уровня владения каждой из них, а также должна быть реализована возможность редактировать их перечень.

На основании вышеперечисленной информации строится база правил для нечеткой системы вывода. Каждому правилу можно присвоить приоритет, путём экспертного назначения весов – «критериев важности» W_i в виде числового значения от 0 до 1.

Например, $C=[C_1, C_2, \dots, C_n]$ — необходимый перечень компетенций для сотрудников ИТ-отдела, а $X=[X_{\min}, X_m, \dots, X_{\max}]$ — уровень владения каждой компетенцией. Может быть составлена такая элементарная база правил, которую в последствии можно будет расширить и дополнить:

- d1: “ЕСЛИ $C_1 = X_{\max}$ И $C_2 = X_{\max}$, И $C_3 = X_{\max}$, ТО К = Высокий”;
- d2: “ЕСЛИ $C_1 = X_{m1}$ И $C_2 = X_{m1}$, И $C_3 = X_{m1}$, ТО К = Выше среднего”;
- d3: “ЕСЛИ $C_1 = X_{m2}$ И $C_2 = X_{m2}$, И $C_3 = X_{m2}$, ТО К = Средний”;
- d4: “ЕСЛИ $C_1 = X_{m3}$ И $C_2 = X_{m3}$, И $C_3 = X_{m3}$, ТО К = Ниже среднего”;
- d5: “ЕСЛИ $C_1 = X_{\min}$ И $C_2 = X_{\min}$, И $C_3 = X_{\min}$, ТО К = Низкий”;

Выводы. Произведен анализ распространенных методов и моделей для решения задач управления персоналом. Выявлен и подробно описан наиболее подходящий метод на основе систем нечеткого вывода для решения задачи оценки компетентности персонала. Описан процесс построения базы правил

для экспертной системы. Процесс оценки компетентности сотрудников и их соответствия занимаемым должностям целесообразно проводить с использованием интеллектуальной системы на основе нечётких методов, работа которой позволит получить объективную оценку соответствия уровня компетентности персонала требованиям руководства организации.

Список литературы

1. Насейкина Л. И., Соколова И. М. Автоматизация поддержки принятия решения подбора персонала ИТ-отдела на основе оценки профессионально-значимых и личностных качеств кандидатов// Технические науки - от теории к практике: сб. ст. по матер. XXXIII междунар. науч.-практ. конф. № 4(29). – Новосибирск: СибАК, 2014.

2. Азарнова Т.В, Степин В. В, Щепина И. Н. Повышение эффективности методов управления развитием персонала на основе нейросетевых моделей и нечетких экспертных технологий// Вестник ВГУ. Серия: Экономика и Управление, 2014. – №3. – С. 121–130.

3. Звездова А.Б. Компетентностный подход в высшем профессиональном образовании. / А.Б. Звездова, В.Г. Орешкин. [Электронный ресурс] — Режим доступа. —

URL: http://www.miep.edu.ru/uploaded/zvezdova_oreshkin.pdf

4. Фот Ю.Д. Модели и алгоритмы системы отбора персонала на основе повышения достоверности данных при принятии решений. Автореф. д. исс. на соик. уч. степ. канд. тех. наук по спец. 05.13.10. Пенза, 2013. — 22 с.

5. Берестнева О.Г., Марухина О.В., Абунаваз Х.А. Алгоритмическое и программное обеспечение информационной системы оценки компетентности студентов технического вуза// Известия Томского политехнического университета. – 2006. – Т.309. – № 7. – С. 240–245.

УДК 004.42

АРХИТЕКТУРА ВЕБ-ОРИЕНТИРОВАННОЙ СИСТЕМЫ РАСПРЕДЕЛЁННОЙ ОБРАБОТКИ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Двойкин А. А., Привалов М. В., Воронова А. И.

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

94dvoikin@gmail.com

Аннотация

Двойкин А. А., Привалов М. В., Воронова А. И. Архитектура веб-ориентированной системы распределённой обработки медицинских изображений. Выполнен анализ и описаны особенности систем обработки медицинских изображений, описана необходимость распределения нагрузки между клиентским приложением и сервером (центром обработки данных).

***Ключевые слова:** системы обработки медицинских изображений, распределённая обработка, распределение нагрузки.*

Цель работы: повысить быстродействие веб-ориентированной системы хранения и обработки медицинских изображений за счёт разработки архитектуры, предусматривающей распределённую обработку данных.

Введение. При установлении диагноза и проведении лечения врачи всё больше полагаются на медицинские изображения, к которым относятся рентгенограмма, УЗИ, магнитно-резонансная томография, компьютерная томография, томография на позитивном излучении и т.д. Использование медицинских изображений непрерывно растёт по мере того, как во многих больницах (сначала в базовых больницах медицинских институтов) устанавливаются системы с высокой пропускной способностью для архивирования изображений. Медицинские изображения дают основной объём информации о пациенте, однако, самих по себе их недостаточно, поскольку требуется анализ и интерпретация в контексте истории болезни пациента (т.е. нужны метаданные, связанные с изображениями). Существует ряд факторов, которые значительно усложняют лечение пациента, основанное на медицинских изображениях: медицинские данные о пациентах собираются в разных лечебных учреждениях; врачи не всегда имеют доступ ко всем историям болезней всех своих пациентов; медицинские изображения представляются очень большими объёмами данных со сложной структурой. Объёмы обусловлены наличием 3D-изображений, показаний, полученных в последовательные моменты времени, многочисленных протоколов описания изображений, а сложность структур обусловлена клинически и эпидемиологически значимыми показателями, такими как возраст, питание,

образ жизни и история болезни пациента, параметры получения изображений и анатомические и физиологические изменения. Часто изображений одной модальности оказывается недостаточно, поскольку на получение изображения влияет много параметров, и дополнительная информация накапливается различными системами сбора физических данных (physical acquisition systems) [1].

Следовательно, информационные системы, которые предоставят возможность доступа к изображению с любого устройства и любого места при наличии интернет-соединения, являются и будут оставаться актуальными ещё долгое время, ведь компьютерная обработка медицинских изображений постоянно развивается, появляются новые физические методы трёхмерного сканирования и визуализации внутренних органов человека [2].

Обзор существующих систем.

В данный момент существует достаточное количество систем распределённой обработки медицинских изображений. Они решают следующую проблему: более широкое применение цифровой патологии в лаборатории означает, что размер изображений и метаданных, подлежащих хранению, продолжает увеличиваться, что приводит к увеличению спроса на места хранения. Наряду с увеличением требований к пространству для хранения данных скорость, с которой эти изображения необходимо анализировать и извлекать, также должна соответствовать растущим требованиям этого динамического диагностического поля [3]. Система должна быть масштабируемой и поддаваться регулярным обновлениям, учитывая быстрые изменения, которые разработчики делают с существующими решениями для анализа и хранения [4].

Для решения данной проблемы они предлагают такие возможности: веб-ориентированность системы, не требующая больших затрат времени и громоздких задач по установке и настройке, простое и удобное добавление и модификация новых алгоритмов разработчиками, доступ в любое время в любом месте из-за веб-архитектуры [5].

Преимуществами таких систем является: наличие функционального веб-ориентированного клиентского приложения, возможность изменения и добавления новых алгоритмов в серверное приложение, возможность интеграции в системы анализа и управления изображениями и т.д.

Но основными и главными для данного исследования недостатками являются: отсутствие возможности распределения нагрузки между клиентскими терминальными устройствами и центром обработки данных (ЦОД) медицинского учреждения, а также добавления сервисов и расширения клиентского приложения. Так, например, открытие клиентского приложения на мобильном устройстве, которое не сможет произвести фрагментацию огромного изображения целиком, будет проблематичным, но, тем не менее, обработка и отображение на экране какого-то фрагмента вполне возможны.

Или если мы используем достаточно мощный для обработки ноутбук, но при этом кластеры ЦОД медицинского учреждения заняты, возможна ситуация, когда пользователь либо не получит изображение, либо будет ждать своей очереди.

Решением данной проблемы является разработка архитектуры, которая будет обладать высокой гибкостью и масштабируемостью, а также будет способна определять сценарий распределённой обработки изображений в зависимости от текущей ситуации.

Разработка архитектуры подсистемы хранения и обработки медицинских изображений.

На рис. 1 изображена общая архитектура подобной подсистемы.

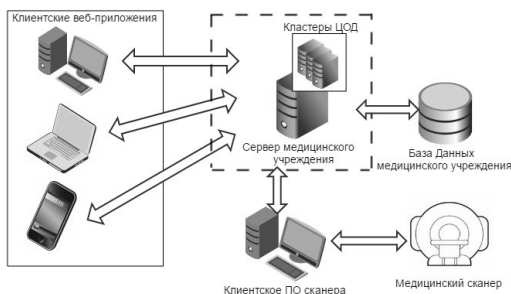


Рисунок 1 – Общая архитектура подсистемы хранения и обработки медицинских изображений

Из рис. 1 видно, что подсистема предоставит доступ к данным, которые хранятся в БД медицинского учреждения, с клиентских устройств благодаря взаимодействию клиентского веб-приложения с подсистемой, являющейся частью сервера медицинского учреждения. Данный сервер имеет прямой доступ к БД медицинского учреждения или удалённой БД, расположенной на кластере ЦОД. Более подробно архитектура подсистемы показана на рис. 2.

Взаимодействие между подсистемами основано на принципах сервис-ориентирования. Такие связи присутствуют между сервисами серверных подсистем и клиентским приложением, аналогично построено взаимодействие и с DICOM-сервером.

Все сервисы подсистемы можно разделить на 5 основных групп:

- сервисы подготовки данных к передаче клиентскому приложению;
- сервисы обработки запросов клиентского приложения;
- сервисы динамического определения сценария обработки;
- API-сервисы;
- сервисы распределённого анализа изображения.

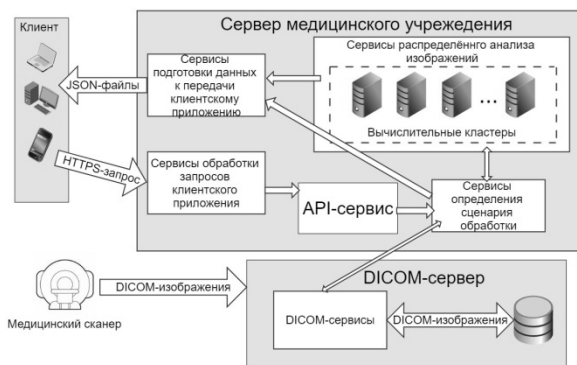


Рисунок 2 – Подробная архитектура подсистемы

Принцип работы подсистемы: с клиентского приложения по защищённому протоколу HTTPS поступает HTTP-запрос, который обрабатывается в сервисах обработки запросов. Сервисы обращаются к API-сервисам, которые подают запрос на определение сценария обработки соответственно клиентскому запросу, вследствие чего сервисы определения сценария получают от DICOM-сервера изображение и данные о нём, а также данные о загруженности ЦОД медицинского учреждения и вычислительной мощности клиентского устройства. После определения сценария задача обработки изображения выполняется в ЦОД, и клиентскому приложению приходит результат обработки, либо изображение в полном виде отправляется клиентскому приложению.

Определение сценария.

Главным критерием при выборе сценария является время обработки t_o :

$$t_o = t_{mc} + t_{ck}, \quad (1)$$

где t_{ck} – время передачи данных с сервера на клиентское приложение;

t_{mc} – время обработки изображения.

Время обработки определяется по формуле:

$$t_{mc} = t_c + t_n + t_\phi, \quad (2)$$

где t_c – время считывания изображения;

t_n – время преобразования изображения определёнными методами;

t_ϕ – время формирования данных для отображения.

Время выполнения каждого этапа обработки зависит от следующих факторов:

- производительности (M) устройства, на котором обрабатывается изображение;
- размера (S) изображения;

– сложности (D) алгоритма каждого этапа обработки.

Следовательно, формула (2) имеет вид:

$$t_{mc} = \frac{M}{S} \cdot \left(\frac{1}{D_c} + \frac{1}{D_n} + \frac{1}{D_\phi} \right), \quad (3)$$

где D_c – сложность алгоритма считывания изображения;

D_n – сложность алгоритма преобразования изображения определёнными методами;

D_ϕ – сложность алгоритма формирования данных для отображения.

Главный параметр $t_{ок}$, который учитывается при принятии решения об обработке на клиентском оборудовании, определён формулой:

$$t_{ок} = t_{скк} + t_{мск}, \quad (4)$$

где $t_{скк}$ – время передачи необработанных данных с сервера клиенту;

$t_{мск}$ – время обработки на клиентском терминальном оборудовании.

Время обработки изображения на клиентском терминальном оборудовании ($t_{мск}$) зависит от:

– производительности (M_k) клиентского терминального оборудования;

– возможности (B) выполнения обработки ($B = 0$, если возможности нет, иначе $B = 1$);

– размера (S) изображения;

– сложности (D) алгоритма каждого этапа обработки.

Таким образом, время обработки изображения на клиентском терминальном оборудовании может быть определено по формуле:

$$t_{мск} = \frac{B \cdot M_k}{S} \cdot \left(\frac{1}{D_c} + \frac{1}{D_n} + \frac{1}{D_\phi} \right) \quad (5)$$

Главный параметр $t_{оц}$, который определяет условие обработки изображения в ЦОД медицинского учреждения, определяется как:

$$t_{оц} = t_{скц} + t_{мсц}, \quad (6)$$

где $t_{скц}$ – время передачи результатов обработки с сервера клиенту;

$t_{мсц}$ – время обработки изображения в ЦОД медицинского учреждения.

В свою очередь, $t_{мсц}$ определено как:

$$t_{мсц} = \frac{M_{кл} \cdot K}{S} \cdot \left(\frac{1}{D_c} + \frac{1}{D_n} + \frac{1}{D_\phi} \right), \quad (7)$$

где K – количество кластеров, между которыми можно распараллелить обработку изображения;

$M_{кл}$ – производительность одного кластера.

Критерий, на основании которого принимается решение об обработке в ЦОД или на клиентском оборудовании, может быть сформулирован следующим образом:

– при $tok > toq$ обработку следует выполнять в ЦОД с учётом ограничений: загрузка кластеров ЦОД на момент запроса, возможность обработки изображения на клиентском терминальном оборудовании;

– при $tok < toq$ обработку следует выполнять на клиентском терминальном оборудовании.

Выводы. Развитие методов диагностики, основанных на получении и анализе медицинских изображений, а также темпы накопления этих изображений позволяют сделать вывод, что повышение быстродействия, гибкости и масштабируемости систем хранения и обработки медицинских изображений – актуальная задача. Анализ архитектур современных информационных систем показал, что масштабируемость и гибкость могут быть повышены за счёт применения принципов сервис-ориентирования. Показано, что оперативность работы и доступность системы возможно повысить за счёт автоматического определения сценария обработки на основании объёма задачи, характеристик клиентского оборудования и текущей доступности ЦОД. Сформулирован критерий выбора узла для обработки. Направлением дальнейших исследований является экспериментальное исследование применяемых методов обработки изображений, их реализаций и типов медицинских изображений с целью определения характеристик производительности оборудования, сложности алгоритма, а также уровня загрузки сети и ЦОД, которые позволят эффективно выполнять выбор сценария обработки.

Список литературы

1. Назаренко Г. И., Гулиев Я. И., Ермаков Д. Е. Медицинские информационные системы: теория и практика / под ред. Г. И. Назаренко, Г. С. Осипова. – М.: ФИЗМАТЛИТ, 2005. – 320 с.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений / М.: Техносфера, 2005. – 1072 с.
3. Попов С. Б. Концепция распределенного хранения и параллельной обработки крупноформатных изображений / С. Б. Попов // Компьютерная оптика – 2007. – т. 31 – №4 – С. 77-85.
4. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван Стеен. – СПб.: Питер, 2003. – 877 с.
5. Симоненко А. А., Привалов М. В. Разработка веб-ориентированной подсистемы доступа к медицинским изображениям // Информатика, управляющие системы, математическое и компьютерное моделирование в рамках II форума «Инновационные перспективы Донбасса» (ИУСМКМ-2016): VII Международная научно-техническая конференция, 26 мая 2016, г. Донецк – Донецк: ДонНТУ, 2016. – 624 с. – С. 274-279.

УДК 004.02

АНАЛИЗ И ОПРЕДЕЛЕНИЕ ЗАДАЧ ЛОГИСТИКИ В ИНТЕРНЕТ-МАГАЗИНЕ

Чаленко К.Ю., Светличная В.А.

Донецкий национальный технический университет г. Донецк
кафедра автоматизированных систем управления
e-mail: prostowebseite@gmail.com

Аннотация

Чаленко К.Ю., Светличная В.А. Анализ и определение задач логистики в интернет-магазине. В данной статье описываются основные виды и задачи логистики, обосновывается необходимость разработки логистической подсистемы интернет-магазина, определяются основные задачи логистической системы интернет-магазина. Проводится обзор аналитических инструментов сбора статистики и описываются основные критерии выбора закупаемой продукции и возникающие ограничения.

Ключевые слова: интернет-магазин, логистика, закупки, прогнозирования, товар.

Постановка проблемы.

Профессионально выстроенная логистика магазина способствует получению конечным потребителем высокого качества обслуживания и доставки. Она является основой построения долгосрочных доверительных взаимоотношений между продавцом и покупателем. [1] Как следствие, возникает необходимость организовать логистику интернет-магазина таким образом, чтобы с одной стороны клиенты оставались довольны во всех смыслах, а с другой стороны магазин приносил максимально возможную прибыль.

Согласно мнению экспертов интернет-магазину, у которого масштаб отправлений заказов в месяц составляет от 1 до 500, нет необходимости выстраивать логистические процессы — подготовку отправлений можно делегировать одному из сотрудников компании. Если же месячный объем отправлений достигает от 500 до 3 тысяч, тогда фулфилмент выгодней организовать самостоятельно, поручив задачи отделу продаж. [2]

В данный момент существует рассматриваемый рабочий интернет-магазин без корректно организованной логистической системы, но присутствует возможность собирать статистику продаж при помощи аналитических инструментов. Учитывая мнение экспертов, т.к. у магазина больше 500 отправлений в месяц, и многое не настроено и не

автоматизировано, возникла необходимость организация собственного логистического центра.

Цель статьи.

Основная цель статьи - определить основные направления развития логистики в интернет-магазине. Проанализировать имеющуюся модель работы предприятия и ее ограничения. Обосновать необходимость и важность собственного логистического центра для онлайн-магазина средних и больших размеров.

Постановка задачи исследования.

Для реализации логистической системы необходимо проанализировать существующие процессы функционирования интернет-магазина, включающие как запросы, продажи, отказы, так и обеспечение интернет-магазина товаром. Результаты такого анализа позволят спроектировать корректную логистическую систему, которая позволит прогнозировать и закупать товар в требуемых количествах в зависимости от спроса, учитывая специфику поставщиков и предприятия. Также позволят определить основные критерии выбора закупаемой продукции, а в дальнейшем рассмотреть правильную организацию логистики.

Решение задач и результаты исследований.

Целью любой логистической системы является организация действий каждого из участников логистической цепи с целью эффективного выполнения задач, которые поставлены перед участниками. [2]. Исходя из составляющих полного логистического процесса, выделяют три основных вида логистики: закупочная, складская и транспортная.

Рассмотрим закупочную логистику. Ее предметом является собственно покупка необходимого продукта, и основными целями в данном случае выступают максимальная экономическая выгода, минимальные затраты трудовых ресурсов, а также оптимальные сроки, в которые укладывается закупка. Соответственно, задачи, которые охватывает закупочная логистика, выглядят следующим образом:

- определение потребностей покупателя;
- выделение и ранжирование по важности основных критериев выбора закупаемой продукции;
- анализ рынка;
- выбор поставщика;
- создание и ведение разнообразных баз данных;
- проведение операций по закупке;
- контроль над своевременной поставкой товара.

Для успешной деятельности интернет-магазина качественная организация закупочной логистики крайне важна.

Складская логистика заключается в оптимальной организации полного цикла складских работ, а именно: приема, обработки, хранения, и отгрузки продукции со складов. В качестве основных задач выделяют:

- перемещение товаров;
- инвентаризация продукции;
- списание, а также оприходование материальных средств;
- правильный расчет себестоимости продукции;
- анализ и учет остатков товара на складе;
- обороты за определенный отчетный период и др.

Транспортная логистика определяется как грамотная организация процесса перемещения товара из одной точки в другую при минимальных сроках, трудозатратах и материальном обеспечении. Перед ней стоят такие задачи:

- выбор подходящего вида транспорта;
- выбор способа транспортировки;
- выбор перевозчика и других логистических партнеров;
- определение рациональных маршрутов доставки;
- обеспечение технологического единства транспортно-складского процесса;
- оптимизация параметров транспортного процесса (увеличение скорости транспортировки, уменьшение расходов топлива и пр.).

Своевременная и быстрая доставка товара покупателю наряду с качеством самого товара является краеугольным камнем успешной работы и дальнейшего продвижения в интернет-бизнесе.

Если проанализировать задачи различных видов логистических систем, то можно сделать вывод, что главная задача логистики заключается в минимизации затрат, которые включены в себестоимость конечного продукта. [2]

Перед созданием новой логистики необходимо проанализировать готовые компоненты, которые уже внедрены в онлайн магазин.

На предприятии в полной мере организована складская логистика. Присутствует складское помещение, которое вмещает определенное количество продукции на продажу. Свободное место освобождается по мере продажи. За инвентаризацию отвечают складские работники. Списание, оприходование, расчет себестоимости, анализ и учет остатков товара осуществляет программа складского учета, которая связана с интернет-магазином.

Что касается закупочной логистики, то присутствует только список поставщиков-фабрик, их ассортимент, возможности поставок, сроки и стоимость продукции. При этом прайс-листы необходимо заново запрашивать перед каждой закупкой, т.к. стоимость отдельных позиций может меняться.

Каждая фабрика производит по меньшей мере одну из пяти основных категорий товара магазина. В нижеприведённой таблице представлены соответствующие данные по категориям товара и предоставляющим их предприятиям (см. Таблица 1).

Таблица 1 – Сопоставление категорий товара с производящими их фабриками

Категория 1	Категория 2	Категория 3	Категория 4	Категория 5
Фабрика 1	Фабрика 2	Фабрика 3	Фабрика 4	Фабрика 5
Фабрика 6	Фабрика 6	Фабрика 7	Фабрика 7	Фабрика 8
Фабрика 9	Фабрика 10	Фабрика 10	Фабрика 9	Фабрика 10
Фабрика 10			Фабрика 10	

Фабрики 1-5 – это основные поставщики, которые поставляют исключительно большие партии продукции по низкой цене, и отличаются также довольно долгой доставкой. Фабрики 6-9 – имеют возможность поставлять продукцию нескольких категорий, к тому же менее масштабными партиями, чем Фабрики 1-5. Они также отличаются более быстрой доставкой, однако и стоимость работы с ними больше, чем при первом варианте. Фабрика 10 может доставить любой товар в любых количествах за короткие сроки, но себестоимость такого товара будет сильно завышена.

Определение потребностей клиентов является основой при выборе закупаемого товара. Для составления реалистичного списка потребностей необходимо использовать веб-аналитику работающего онлайн-магазина. Аналитика представляет собой непрерывный сбор, анализ и обработку сведений о действиях клиентов.

В существующем интернет-магазине есть возможность автоматически определять востребованность каждой позиции товара. Сайт связан с двумя основными аналитическими системами – Google Analytics и Яндекс.Метрика. Они позволяют определить количество покупок, конверсию и, конечно же, популярность конкретных товаров или продуктов. Эти параметры дают возможность спрогнозировать время, когда необходимо пополнить запасы определенного товара, а также помогают определиться с объемами закупок.

На складе хранится приблизительно 3000 позиций. К сожалению, докупать единицы товара по мере их покупки потребителями – решение нерентабельное из-за издержек при доставке заказа на склад. Поэтому важно понимать, когда нужно провести закупку. Таким образом, для каждого товара, зная среднее время его реализации, можно вычислить порог количества, при котором необходимо делать заказ (см. рис. 1).

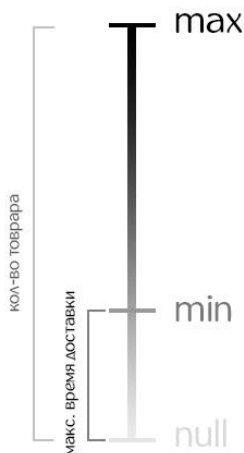


Рисунок 1 – График наличия товара на складе

При этом порог для нового заказа \min высчитывается по следующей формуле:

$$\min = t_{d \max} * \bar{t}_s, \quad (1)$$

где $t_{d \max}$ – максимальное время доставки товара, дней;

\bar{t}_s – среднее время покупки товара покупателями, шт./день.

В момент, когда количество позиций какого-либо товара опускается до числа “ \min ”, необходимо проводить закупку. Время между “ \min ” и “ null ” – максимальное время, за которое возможно доставить новую партию товара на склад.

Например, мы знаем, что максимальное количество товара «А» на складе – 1000 ед. По имеющейся статистике можно понять, что каждый день через интернет-магазин покупается в среднем 3 единицы данного товара. Самый выгодный по цене вариант доставки занимает 2-3 месяца по при минимальном размере партии в 800 ед. Из этого следует, что критический порог, при котором необходимо проводить закупку:

$$\min = (3 \text{ мес.} * 30 \text{ дней}) * 3 \text{ ед./день} = 270 \text{ ед.} \quad (2)$$

Округляем до сотен в качестве страховки на случай непредвиденной задержки, и получаем, что при самых худших условиях времени доставки нам

необходимо делать закупку при падении количества товара «А» до порога в 300 единиц. Далее данный товар заносится в базу данных как нуждающийся в закупке. По достижению определенного количества позиций, которые необходимо заказать, формируется запрос для подходящей фабрики.

Также немаловажный критерий, который можно вычислить при помощи аналитики – рентабельность совершённых закупок и интернет-магазина в целом. Основываясь на этих данных, можно скорректировать будущие заказы – повысить приоритет какого-то товара, отказаться от ненужных позиций, выбрать иной способ доставки.

Если есть возможность определить, когда и какой товар нужно закупать, то появляется вопрос – какие же основные критерии выбора закупаемой продукции? В случае с интернет-магазином существует несколько ограничений.

Это размер складского помещения, стоимость товара, количество оборотных средств и поставщики. Если рассматривать эти пункты отдельно, то получается, что размер склада не составит больших проблем, если вычислить порог каждого товара и производить заказы только при достижения этого самого порога. Также стоит учитывать, что складская логистика уже успешно работает на предприятии. Стоимость товара будет варьироваться в зависимости от типа доставки.

Поставщики и доставка же требуют более тщательного рассмотрения.

После выбора подходящей фабрики с ней необходимо связаться. Далее весты переговоры, в которых есть много важных нюансов, кроме окончательной цены. Требуется обязательно обсудить качество товара, условия оплаты, сроки поставок и ваши дальнейшие отношения с поставщиком продукции. Кроме того, у любой компании есть нижняя планка цены – минимальное рентабельное значение.

Выгоднее всего привозить легкий по весу товар, который можно заказывать большими объемами. Т.к. товар магазина достаточно легкий, можно выделить 4 наиболее подходящих способа доставки.

Курьерская доставка можно использовать, если товар уже оплачен покупателем и можно быть уверенным, что нет никаких рисков. Среди подобных служб можно выделить DHL, FedEx и EMS. Стоимость вычисляется из расчета за вес. Работая с курьерскими службами, можно не беспокоиться о решении бумажных и таможенных вопросов. [3] Эти службы позволяют получать товары очень быстро, но стоимость редко оправдывает этот способ.

Международную почту обычно используют для доставки образцов товара, правда по срокам это занимает до пары месяцев. Нет никакой возможности отследить местонахождение посылки и есть вероятность, что посылку могут потерять.

Третий вариант - воздушный фрахт. Обычно доставка занимает 5-10 дней, стоимость ее ниже, чем в вышеуказанных способах, однако придется самостоятельно решать все таможенные вопросы.

Самый популярный способ доставки – морской фрахт. Он требует довольно много времени. К тому же, придется самостоятельно разбираться с бумагами, или воспользоваться услугами экспедирующей компании. Стоимость конечного продукта будет минимальная, но заказ придется делать за несколько месяцев. Также необходимо заключить договор с экспедирующей компанией/компаниями для экономии времени прохождения таможни.

Таким образом, логистическая система должна заниматься решением следующего ряда вопросов:

- организация качественного аналитического аппарата для мониторинга ситуации на рынке – как с потребителями, так и с поставщиками;
- построение рациональной системы товарооборота для точного определения объёмов закупок и их сроков;
- составление наиболее оптимального списка поставщиков;
- подбор подходящих средств и методов доставки.

Следует отметить, что при подходе к вышеперечисленным задачам необходимо учитывать ограничения, продиктованные индивидуальными особенностями предприятия, для которого разрабатывается система, а именно:

- объем имеющихся складских помещений и возможность оперативного расширения рабочей площади в случае непредвиденных обстоятельств;
- запас доступных для использования финансовых средств и наличие кризисного фонда;
- необходимость учёта особенностей работы различных поставщиков (ценовая политика компании и её стабильность, разнообразие и стоимость доставки товара).

Планирование закупок и формирование заказов поставщикам (Закупочная логистика) будет выглядеть следующим образом (см. рис. 2).

В результате проведенных исследований о необходимости разработки логистической системы интернет-магазина выделены задачи такой системы и ограничения, накладываемые на принятия окончательных решений.

Разработка собственного логистического центра для интернет-магазина уменьшит затраты, которые включены в себестоимость конечного продукта и поможет автоматизировать отдельные элементы работы компании и увеличить товарооборот.

Выводы. Проведен анализ имеющейся модели работы компании и интернет-магазина. Выполнен обзор основных видов логистики. Рассмотрены способы прогнозирования закупок товара. Обоснована необходимость организации собственного логистического центра. Представлена диаграмма деятельности работы будущей закупочной логистической системы.

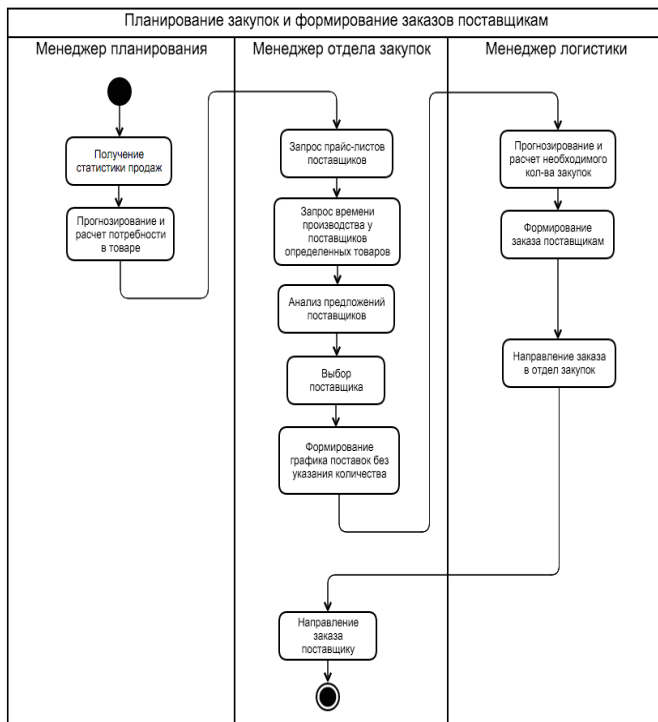


Рисунок 2 – Диаграмма деятельности. Планирование закупок и формирование заказов для интернет-магазина

Следующими шагами работы должна стать детальная разработка логистической системы с выбором и обоснованием методов определения наиболее оптимальных вариантов закупок товара.

Список литературы

1. Оптимизация издержек на логистику для интернет-магазина [Электронный ресурс] – Режим доступа: <http://www.logistics.ru/eshops/news/optimizaciya-izderzhek-na-logistiku-dlya-internet-magazina> (дата обращения: 10.05.2017).
2. Как организовать логистику в интернет-магазине [Электронный ресурс] – Режим доступа: <https://www.shopolog.ru/metodichka/logistics/kak-organizovat-logistiku-v-internet-magazine/> (дата обращения: 08.05.2017).
3. Четыре способа доставки в Россию [Электронный ресурс] – Режим доступа: <http://emagnat.ru/organizaciya-postavkok-iz-kitaya.html> (дата обращения: 11.05.2017).

УДК 004.415.2

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ КОМПЬЮТЕРИЗИРОВАННОЙ ПОДСИСТЕМЫ УПРАВЛЕНИЯ РАБОТОЙ СЛУЖБЫ ТАКСИ

В. О. Коношенко, М. В. Привалов

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
konoshenko.vl@gmail.com

Аннотация

Коношенко В. О., Привалов М. В. Проектирование архитектуры компьютеризированной подсистемы управления работой службы такси. *Спроектирована архитектура компьютеризированной подсистемы управления работой службы такси, основанная на принципах сервис-ориентирования. Предложено решение на основе WebSocket, позволяющее предоставить постоянное асинхронное соединение между компонентами подсистемы.*

Ключевые слова: *данные, сервер, архитектура, REST, COA, WebSocket.*

Постановка проблемы. Ценность различных Интернет-сервисов в том, что они постоянно предоставляют доступ к актуальной информации. Отличным примером являются различного рода социальные сети, мессенджеры и т.д. Такие приложения имеют сложную сервис-ориентированную или распределённую Интернет-архитектуру, и характеризуются наличием следующих компонентов:

- клиентское приложение;
- серверная часть (так называемый backend либо слой сервисов);
- коммуникации между компонентами;
- система управления базами данных.

Качество работы такой системы в значительной мере зависит от архитектуры каждого компонента, а также от методов соединения клиентов с серверной частью. Зачастую, для поддержания информации в актуальном состоянии, коммуникации стараются сделать постоянными, асинхронными и двухсторонними, обеспечивая организацию взаимодействия по инициативе любого из компонентов. Асинхронность особенно важна для мобильных приложений, так как позволяет экономить трафик. Компьютеризированная подсистема управления работой службы такси предполагает доступ к ней различных пользователей с различных устройств. При этом уведомления о событиях, происходящих в системе, должны направляться как от клиентов к серверу, так и в обратном направлении, по инициативе сервера. Исходя из этого, задача проектирования архитектуры данной подсистемы с учётом

обеспечения постоянного асинхронного соединения между её компонентами является актуальной.

Цель работы – снижение латентности компьютеризированной подсистемы управления работой службы такси и снижение потребления трафика за счёт разработки компонентной архитектуры подсистемы с предоставлением постоянных асинхронных коммуникаций между её частями.

Сравнительный анализ различных типов архитектур. Для достижения поставленной цели необходимо рассмотреть основные типы архитектур – сервис-ориентированную (СОА) и распределённую Интернет-архитектуру.

В распределённой архитектуре обработка информации распределена между несколькими независимыми узлами. В совокупности они представляют единую инфраструктуру. Основные требования к распределённой архитектуре заключаются в следующем. Каждый уровень распределённого приложения должен взаимодействовать со смежными уровнями. Вся сложная бизнес-логика должна быть реализована в отдельном слое. Количество передач информации между узлами следует минимизировать. Преимущества распределённой архитектуры:

- открытость (гибкое использование ПО различных производителей);
- параллелизм;
- масштабируемость;
- отказоустойчивость.

Недостатки: распределённая архитектура восприимчива к внешним атакам, ниже управляемость, работа зависит от организации системы и сетевых нагрузок.

СОА предполагает модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных между собой, заменяемых компонентах, которые оснащены стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам. Главное отличие СОА от распределённой Интернет-архитектуры – это использование независимых сервисов, которые имеют чётко определённые интерфейсы. Особенность такой архитектуры – возможность собрать новые процессы из существующих служб, функциональная совместимость [1]. Рассмотрев данный тип архитектуры, можно выделить следующие её преимущества:

- каждый компонент независим от других компонентов;
- слабое связывание сервисной ориентации обеспечивает большую гибкость;
- клиент или любая услуга могут получить доступ к другим услугам независимо от их реализации, платформы, технологии, поставщика или языка.

Сравнительный анализ использования способов коммуникаций. REST является архитектурным стилем или набором принципов [2, 3]. В свою очередь, SOAP – это семейство протоколов. SOAP требует очень много сервисных данных для отправки и получения своих сообщений, так как использует XML для обеспечения независимости от языка программирования на котором написан наш конечный продукт.

Проблемы использования SOAP:

- очень большой размер данных из-за использования XML и лишние данные, связанные с метаданными, как правило, представленными в виде документа WSDL;
- в случае изменения кода без обновления WSDL, клиенты не смогут работать, так как изменения не описаны в метаданных;
- для передачи данных можно использовать только XML формат в текстовой или двоичной форме.

Положительные стороны:

- безопасность: невозможно использовать функции, не описанные в WSDL документе;
- предоставляет сразу описание всего функционала веб-сервиса.

REST, в отличие от COA, проще в использовании на практике. Этот стиль предполагает легковесные коммуникации посредством передачи данных о состоянии по протоколу JSON, передаваемому поверх HTTP. Это позволяет снизить накладные расходы коммуникаций между компонентами систем. Также снижается риск изменения наборы методов на сервисах: достаточно того чтобы клиент знал, какие объекты необходимо передавать методам сервиса, при этом не будет никаких конфигурационных файлов и XML.

Таким образом, на данный момент использование модифицированной COA, где коммуникации построены аналогично архитектурному стилю REST, является наилучшим решением, так как это гарантирует масштабируемость системы, независимость каждого узла, легкость в разработке.

Проектирование архитектуры подсистемы. Проектируемая система управления работой службы такси включает в себя веб-клиент, мобильное приложение, сервер и сервер базы данных, а также предполагает взаимодействие со сторонними сервисами, например, со службой Google Maps. Для реализации такой системы необходимо предусмотреть сервисы для авторизации, передачи данных, кэширования данных, обработки локальной логики работы приложений. Также они имеют различную реализацию слоя представления. Сервер должен иметь методы для передачи данных и слой сохраняемости для работы непосредственно с БД. Предлагаемая архитектура данной системы представлена на рис. 1.

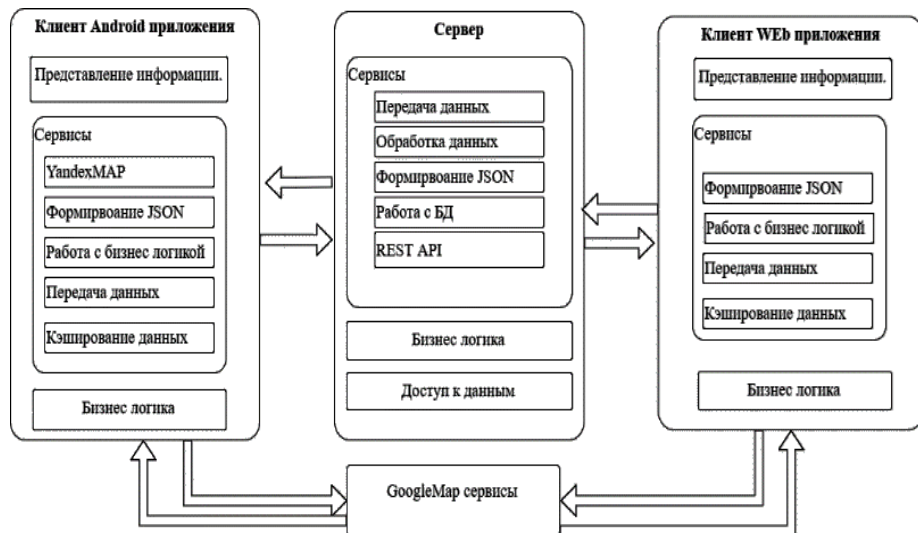


Рисунок 1 – Предполагаемое архитектурное решение

На практике для обеспечения качественного соединения узлов подсистемы немаловажную роль играют способы соединения между ними.

Анализ технологий передачи данных между компонентами подсистемы. WebSocket – протокол полнодуплексной связи (может передавать и принимать одновременно) поверх TCP-соединения, предназначенный для обмена сообщениями между веб-клиентом и веб-сервером в режиме реального времени [4]. Это крайне полезная технология для разработчиков, которые создают приложения, связанные с интенсивным обменом данными, требовательные к скорости обмена и стабильности канала связи. WebSocket позволяет передавать данные по TCP-соединению в обоих направлениях. Такой функционал позволяет снизить задержки и нагрузки на сеть. Он основан на протоколе HTTP и занимает его место после процедуры установление соединения. Схема работы WebSocket'a представлена на рис. 2. Такой метод передачи данных необходим для более оперативного взаимодействия в реальном времени и потоковой передачи.



Рисунок 2 – Схема работы WebSocket

Сегодня REST является самым стандартизированным способом структурирования API для запросов. Для большинства приложений информация требуется только передать, когда пользователь предпринимает действие. Например, при просмотре новостного сайта, когда браузер запросил статью, пользователь занят ее чтением и не предпринимает никаких действий.

Решение описанных проблем. Для достижения поставленной цели необходимо спроектировать архитектуру на основе COA и REST, а также предоставить технологии, которые позволят создавать постоянные асинхронные соединения между клиентом системы и серверной частью. При проектировании архитектуры предполагается предоставить сервисы для работы сразу двух технологий передачи (REST API и WebSocket). Распределив технологии по их назначению, мы получаем гибкую систему. Для постоянного отслеживания местонахождения автомобиля службы такси, движущегося по маршруту, и предоставления актуальной информации, как водителя, так и клиента, будет использоваться технология WebSocket. А при необходимости отредактировать какие-либо данные, потребуется просто отправить запрос, реализованный при помощи REST API (рис. 3).

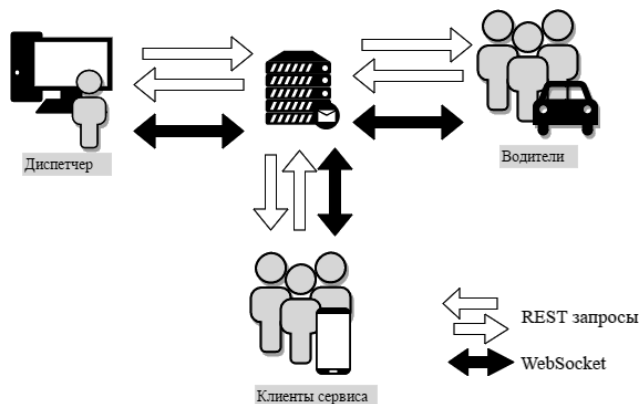


Рисунок 3 – Поток передачи данных

Выводы. Для достижения поставленной цели были проанализированы различные типы архитектур информационных систем. Выяснено, что сервис-ориентированная архитектура с применением отдельных свойств REST является наилучшим решением. Спроектирована архитектура подсистемы управления работой службы такси, в которой обеспечена возможность постоянных асинхронных коммуникаций наряду с типовыми HTTP/REST. Для предоставления асинхронного постоянного соединения клиента с сервером был выбран метод связи, основанный на технологии WebSocket, что предоставляет возможность обмениваться данными как по запросу клиента, так и по инициативе сервера, тем самым избегая периодического обновления. Такой подход позволит сэкономить трафик, что весьма критично для доступа к системе с мобильных устройств водителями такси, а также повысить отзывчивость подсистемы в целом.

Список литературы

1. Сервис-ориентированная архитектура [Электронный ресурс] // Википедия – свободная энциклопедия. (дата обращения 15.05.2017) URL: http://ru.wikipedia.org/wiki/Сервис-ориентированная_архитектура.
2. REST [Электронный ресурс] // Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/REST>. (дата обращения 16.05.2017).
3. Рой Филдинг. Архитектурные стили и проектирование сетевых программных архитектур.[Литература] // Диссертация Роя Филдинга. URL: www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (дата обращения 16.05.2017).
4. WebSocket [Электронный ресурс] // Википедия – свободная энциклопедия. URL: <http://ru.wikipedia.org/wiki/WebSocket>. (дата обращения 15.05.2017).

УДК 004.622

ПРОГНОЗИРОВАНИЕ ТЕХНИЧЕСКОГО СОСТОЯНИЯ ОБОРУДОВАНИЯ СИСТЕМЫ КОНТРОЛЯ И УПРАВЛЕНИЯ ДОСТУПОМ

Д.В. Слюнкин, Т.А. Васяева

Донецкий национальный технический университет, г. Донецк

кафедра компьютерных систем мониторинга

dimas2592@gmail.com, vasyaeva@gmail.com

Аннотация

Слюнкин Д.В., Васяева Т.А. Прогнозирование технического состояния оборудования системы контроля и управления доступом. Рассмотрена актуальная задача прогнозирования технического состояния оборудования различных систем. Разработан метод прогнозирования технического состояния на основании генетического алгоритма и множественной регрессии.

***Ключевые слова:** прогнозирование отказов, хромосома, множественная регрессия, фитнес-функция.*

Актуальность. Прогнозирование технического состояния входит в список наиболее эффективных методов повышения эксплуатационной надежности любого оборудования. Прогноз позволяет предупреждать внештатные отказы и аварии, возникающие при эксплуатации оборудования. Финансовые потери от ошибочных решений о прекращении эксплуатации промышленного оборудования или, наоборот, о необоснованном продлении его ресурса могут измеряться миллионами рублей. Оборудование современной системы контроля и управления доступом (СКУД) снабжено различными датчиками и приборами контроля, что делает задачу оценки данного оборудования и прогнозирования его отказов чрезвычайно актуальной для обеспечения безопасной эксплуатации и планирования его ремонтов. Основной путь повышения эксплуатационной надежности состоит в том, чтобы прогнозировать неисправности устройств. Наличие достаточно точного прогноза отказов сможет обеспечить высокую надежность, безопасность и экономическую эффективность эксплуатации объекта.

Прогнозирование отказов. Современное оборудование снабжено необходимым количеством датчиков и приборов контроля, позволяющих отслеживать большое количество параметров его работы. При прогнозировании отказов необходимо выбрать наиболее информативные из них. Для решения этой задачи могут быть использованы методы машинного обучения (МО). Прогнозирование отказов оборудования на основе методов МО в последнее время становится все более актуальной задачей, решаемой в

таких областях, как: машиностроительное производство, транспортные системы, промышленность, поскольку.

Однако предложенные подходы к построению прогнозных моделей тоже не лишены недостатков, так как ориентированы на обучение на прецедентах, что в ряде случаев не может быть применено на практике. Например, в ситуациях, когда поломки происходят достаточно редко или нет достаточно большой статистики. Применение методов МО для повышения качества прогнозов отказов оборудования с целью снижения аварийности, издержек является актуальным.

Процедура МО требует достаточного количества данных, соответствующих описываемому режиму работы, так как сбор данных, соответствующих тем или иным отказам требует наблюдения в течение длительного времени эксплуатации, и этих данных может быть критически мало, например, для нового оборудования. Поэтому обучение осуществляется на данных, которые соответствуют нормальному режиму функционирования объекта исследования (т.е. в то время, когда не было поломок и других аномалий). Тем самым модель учится прогнозировать, каким должен быть сигнал при нормальном режиме работы. В том случае, если в определенный момент времени фактическое значение сигнала отличается от прогнозируемого «нормального» значения сигнала, фиксируется аномальное поведение и сигнализируется о возможной поломке.

Входными данными для нашей прогностической модели будут различные параметры оборудования (номинальное входное напряжение, допустимое отклонение входного напряжения, собственный ток потребления, ток потребления при максимальной нагрузке, диапазон допустимых значений выходного тока и т.д.).

На выходе модели получим нормальное или аномальное поведение системы.

Эксплуатация СКУД предполагает контроль некоторого количества параметров его работы, некоторые из них вызывают изменения в техническом состоянии, что может привести к отказу. Существенной особенностью некоторых факторов является их случайный (стохастический) характер. Случайный характер рассмотренных факторов приводит к непредсказуемому изменению технического состояния устройств, их узлов, механизмов, следовательно, и времени работы до отказа.

Построим математическую модель классификации. Как правило на первом этапе определяется перечень всех параметров. К таким параметрам например можно отнести: напряжение питания, ток потребления, коммутируемое напряжение на различных клеммах, сопротивление линии связи с контроллерами доступа и т.д.

Положительный эффект от избыточности собираемых параметров заключается в том, что модель прогнозирования сможет учитывать такие

факторы, которые могут показаться незначительными с точки зрения влияния на отказ. При этом важным является то, что обучающая выборка состоит только из данных, в которых нет отказов и аномалий. На втором этапе разрабатывается непосредственно модель прогнозирования.

В нашей работе мы объединим эти два этапа. Очевидно, что выбирая тот или иной набор параметров можно получать различные модели для прогнозирования. Будем решать две задачи и одновременно: выбор факторов которые будут переменными множественной регрессии (нашей прогностической модели); и расчет коэффициентов уравнения регрессии. Будем решать эту задачу с помощью ГА. Стандартный ГА предполагает следующую последовательность (рис. 1).

На первом этапе выполняется генерация начальной популяции, где случайным образом генерируется некоторое количество особей. Важным этапом является разработка способа кодирования хромосомы, так как каждая особь – это возможное решение задачи (уравнение множественной регрессии). Предлагается следующая структура хромосомы. Каждая особь представляется последовательностью определенного количества битов (определяется максимально возможным количеством факторов риска). Значение каждого бита может быть равно «1», если фактор с соответствующим номером присутствует в данном уравнении регрессии, и «0», если этот фактор отсутствует. Таким образом, каждой хромосоме соответствует регрессионная модель (рис. 2), которая будет строиться и решаться методом наименьших квадратов (МНК) по соответствующему ее структуре набору входных параметров.

Второй этап подразумевает расчет фитнес - функции. В качестве фитнес - функции выступает ошибка регрессионной модели, которая может рассчитываться по формуле:

$$E = \frac{1}{N} \cdot \sum_{i=1}^N (y - y')^2,$$

где N – количество примеров в обучении, y – полученный результат регрессионной модели, y' – действительный результат.

Рассчитанная ошибка сопоставляется каждой хромосоме. Лучшая особь будет иметь минимальную ошибку, то есть минимальную фитнес - функцию.

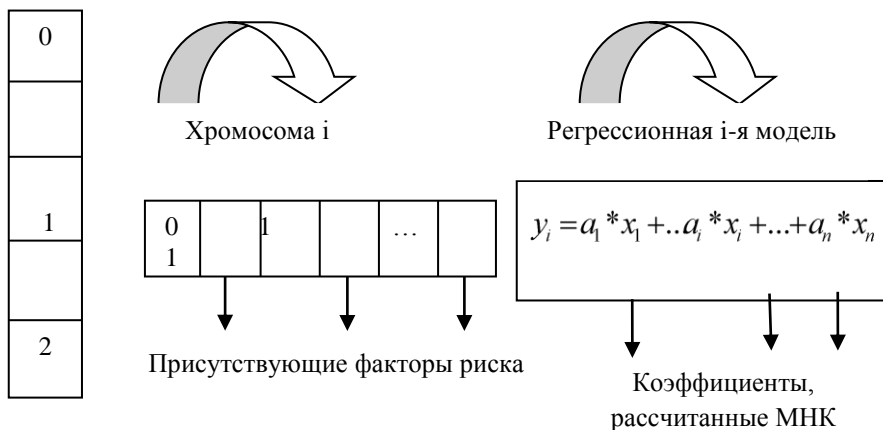
Третий этап является реализацией генетического алгоритма. Генетические операторы выполняются стандартные. Самый простой метод реализации оператор репродукции (ОР) – построение колеса рулетки, в которой каждая хромосома имеет сектор, пропорциональный ее значению ЦФ. Также применяются простые операторы кроссинговера (ОК) и мутации (ОМ).

Следующий этап алгоритма предполагает расширение популяции за счет добавления новых, только что созданных особей.



Рисунок 1. Простой генетический алгоритм

Факторы риска



Затем полученную популяцию сокращаем до исходных размеров (работает оператор редукции), путем отбора лучших особей.

Критериев останова работы алгоритма будет два: максимально заданное количество шагов; незначительное изменение значения фитнес-функции. После выполнения критерия останова выполняется поиск лучшей особи в конечной популяции, что является результатом работы алгоритма (уравнение множественной регрессии). В случае, не выполнения критерия останова, переходим ко второму этапу.

Выводы. Рассмотрена актуальная задача для прогнозирования технического состояния оборудования системы контроля и управления доступом. В дальнейшем планируется реализовать рассмотренный математический аппарат и провести эксперименты на реальных показателях оборудования, предоставленными сотрудниками Старобешевской ТЭС. Планируется разработка и внедрение подобных решений прогнозирования для энергоблока.

Список литературы

1. Д. Рутковская Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская М. Пилиньский, Л. Рутковский. – М.: 2004. – 452 с.
2. Система прогнозирования отказов оборудования промышленных предприятий URL: <http://cyberleninka.ru/article/n/sistema-prognozirovaniya-otkazov-oborudovaniya-promyshlennyh-predpriyatiy>
3. Прогнозирование технического состояния технологического оборудования URL: <http://cyberleninka.ru/article/n/prognozirovanie-tehnicheskogo-sostoyaniya-tehnologicheskogo-oborudovaniya>

Секция № 5
Математическое
и компьютерное
моделирование



УДК 519.725, 625.7, 681.3

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ КОМПАКТНОГО ТЕСТИРОВАНИЯ НА ОСНОВЕ МЕТОДОВ ЦИКЛИЧЕСКОГО КОДИРОВАНИЯ

О.Н. Дяченко, Ю.Е. Зинченко, В.О. Дяченко

Донецкий национальный технический университет, г. Донецк

кафедра компьютерной инженерии

do@donntu.org

Аннотация

Дяченко О.Н., Зинченко Ю.Е., Дяченко В.О. Исследование эффективности компактного тестирования на основе методов циклического кодирования. Выполнен анализ эффективности компактного тестирования комбинационных схем, учитывающего структуру генератора тестовых последовательностей, анализатора тестовых реакций, построенных на основе методов циклического кодирования, и характер распределения ошибок в тестовой реакции.

***Ключевые слова:** генератор тестовых последовательностей, анализатор тестовых реакций, порождающий полином, циклические коды.*

Введение. Всемирная паутина разрастается стремительными темпами: если в 2000 году было 359 млн. пользователей сети Интернет, то в марте 2017 года количество пользователей уже 3732 млн. – это половина населения земного шара. Наша планета постепенно оказывается в своеобразном информационном “коконе”. Увеличение количества информации, которая передается, хранится и обрабатывается, приводит к требованиям обеспечения ее достоверности и надежности используемых аппаратных и программных средств. От успешного решения этих задач, с одной стороны, зависит процветание нынешней цивилизации, или, с другой стороны – ее саморазрушение, например, из-за случайного или намеренного сбоя в военных приложениях. Поэтому для устранения последствий возможных ошибок или дефектов аппаратных средств используется весь арсенал методов и средств помехоустойчивого кодирования, встроенного самотестирования [1-5].

Одним из способов повышения тестопригодности СБИС микропроцессоров, устройств на ПЛИС является применение встроенных средств контроля, реализующих методы компактного тестирования. Метод сквозного сдвигового регистра (LSSD - level sensitive scan design) - другой широко известный способ снижения трудоемкости тестирования дискретных устройств. Метод LSSD сводит задачу тестирования к проверке нескольких регистров сдвига и комбинационных схем. Наиболее совместимым с методом LSSD из широкого ряда методов компактного тестирования является

сигнатурный анализ, поскольку основой анализатора тестовых реакций (АТР) в этом случае является регистр сдвига с линейными обратными связями (РСЛОС). С помощью незначительных аппаратных затрат сдвиговые регистры преобразуются в РСЛОС, которые выполняют роль генераторов тестовых последовательностей (ГТП) и АТР для тестирования комбинационных схем (КС). Реализация методов компактного тестирования ставит задачу определения достоверности результатов контроля. В работе [3] рассматриваются вопросы комплексной оценки достоверности тестирования КС при применении ГТП и АТР в виде РСЛОС, которая учитывает не только обнаруживающие способности АТР, но также структуру ГТП и характер тестовых реакций объекта диагностики. В частности, получен вывод о значительной зависимости эффективности сигнатурного анализа от выбора того или иного сочетания порождающих полиномов РСЛОС ГТП и АТР. Данная работа представляет собой продолжение исследований в этом направлении.

Цель статьи – провести анализ эффективности компактного тестирования цифровых схем при предположении, что ГТП - РСЛОС с примитивным образующим полиномом, а АТР реализованы на основе циклических кодов. Особенностью таких анализаторов является их способность локализовать ошибки, кратность которых не превышает корректирующих способностей кодов, на основе которых они построены.

Практическая реализация самотестирования СБИС. Анализ диагностического обеспечения микропроцессорных СБИС ведущих зарубежных фирм: IBM (S/390, метод LSSD); Hewlett Packard (сигнатурный анализ); альянс компаний Apple, IBM и Motorola (Power PC); Motorola (MC 202-206); Intel Corporation (микропроцессоры 80386, Pentium Pro); Advanced Micro Devices (AMD-K6), показывает, что 5-8 % площади кристалла СБИС занимают встроенные схемы тестирования, которые позволяют обнаружить практически 100% дефектов. Например, диагностическое обеспечение микропроцессора S/390 включает: ОЗУ, кэш, память, схемы их управления со встроенными схемами самотестирования; триггеры, регистровые сети, образующие в режиме тестирования сканируемый путь по методу LSSD; встроенные ГТП; встроенный АТР - многоканальный сигнатурный анализатор; порт JTAG в соответствии со стандартом IEEE 1149.1.

Методы исчерпывающего тестирования КС и сканирования позволяют вместе обнаруживать 95% неисправностей. Применение разных псевдослучайных последовательностей, обеспечивает 99,9 % покрытия всех неисправностей СБИС.

Аналитический метод вычислений компактных оценок. Прежде всего, рассмотрим метод аналитического расчета сигнатур, альтернативный методу, предложенному в [3].

Предположим, что ГТП и АТР реализованы в виде РСЛОС с внутренними сумматорами в цепях обратной связи с порождающими полиномами соответственно $h(X)$ и $g(X)$, причем оба полинома примитивные, а их корни связаны равенством $\beta = \alpha^k$, $m = \deg h(X) = \deg g(X)$.

Тестовые наборы, которые поступают на входы исследуемой КС, представляют собой ненулевые элементы поля $GF(2^m)$, являющегося расширением поля $GF(2)$ над полиномом $h(X)$. Эти элементы поля могут быть представлены в двоичном, полиномиальном и степенном обозначениях. Каждому ненулевому элементу α^k поля $GF(2^m)$ соответствует минимальный полином, причем, если минимальный полином примитивный, то его степень равна m . Если в качестве порождающего полинома РСЛОС АТР выбрать минимальный полином, соответствующий элементу α^k , то между корнями полиномов $h(X)$ и $g(X)$ будет выполнено равенство $\beta = \alpha^k$. Анализ таблицы минимальных полиномов показывает, что для любой степени $m < 5$ существует только два примитивных полинома, причем $\beta = \alpha^{-1}$, т. е. эти полиномы являются двойственными (взаимобратными). Поэтому для примеров, иллюстрирующих метод аналитического расчета сигнатур, будем рассматривать $h(X)$ и $g(X)$ степени $m=5$.

Основное отличие предлагаемого метода расчета сигнатур от известного [3] заключается в выборе степенного обозначения тестовых наборов вместо двоичного. В этом случае значение сигнатуры для конъюнкции с рангом m может быть вычислено согласно следующему выражению: $S = M_k X^{-Ak}$, где X^A - степенное обозначение тестового набора, M - матрица для перехода от значений РСЛОС ГТП к значениям РСЛОС АТР.

Рассмотрим порядок расчета сигнатур на примерах.

Пример. Пусть $F_1 = \bar{x}_5 \bar{x}_4 x_3 x_2 x_1$, $g(X) = X^5 + X^3 + 1$, т.е. $k=-1$.

Элемент поля $GF(2^5)$ над полиномом $h(X) = X^5 + X^2 + 1$, соответствующий тестовому набору 00111, может быть представлен в полиномиальном обозначении $X^2 + X + 1$. Для перехода к степенному обозначению воспользуемся таблицей логарифмов Зеча (табл. 1). Согласно этой таблице $X^j = X^{i+1}$. Учитывая свойства $X^{2i} = (X^i + 1)^2$ и $X^i = (X^{p-i} + 1)$, где $p = 2^m - 1$, определим таблицу для нечетных значений i от 1 до 15. Таким образом, $X^2 + X + 1 = X \cdot (X + 1) + 1 = X \cdot X^{18} + 1 = X^{19} + 1 = X^{11}$. Значение сигнатуры $S(F_1) = M_{-1} X^{11}$.

Таблица 1 – Логарифмы Зеча для $h(X) = X^5 + X^2 + 1$

i	1	3	5	7	9	11	13	15
j	18	29	2	22	16	19	14	24

Несколько замечаний по поводу построения матрицы M_k . Прежде всего, следует отметить, что вид матрицы M_k зависит не только от k , но также от начального значения РСЛОС ГТП, которое может быть выбрано любым

ненулевым. Каждому элементу поля $\alpha^0, \alpha^1, \dots, \alpha^{m-1}$ ставится в соответствие строка матрицы M_n , которая определяется следующим образом. Для элемента α^j отыскивается значение степени j эквивалентного элемента $\alpha^i = \alpha^j$, которое нацело делится на $-k$. Значение j определяется на основании равенства $\alpha^i = \alpha^{j+pd}$, где d - любое целое число. Строка матрицы M_k представляет собой остаток от деления полинома $X^{p+j/k+(s-1)}$ на полином $g(X)$ (α^s - начальное состояние РСЛОС ГТП).

Для рассматриваемого примера при начальном состоянии РСЛОС ГТП, равном 00001, для строки, соответствующей α^0 , значение равно остатку от деления полинома $X^{31-0-1} = X^{30}$ на полином $g(X) = X^5 + X^3 + 1$.

Для начального состояния РСЛОС ГТП равного α^0 и $k=-1$ матрица M_{-1} представляет собой последние m состояний РСЛОС АТР.

Для начального состояния РСЛОС ГТП равного α^m и $k=-1$ матрица M_{-1} представляет собой единичную матрицу, например, при $m=5$ значение сигнатуры конъюнкции с рангом m будет равно двоичному обозначению соответствующего элемента поля, записанному в обратном порядке.

Два варианта матрицы M_{-1} будут иметь вид:

$$M_{-1} = \begin{pmatrix} 01011 \\ 10110 \\ 00101 \\ 01010 \\ 10100 \end{pmatrix}, \quad M_{-1} = \begin{pmatrix} 00001 \\ 00010 \\ 00100 \\ 01000 \\ 10000 \end{pmatrix} \text{ - единичная матрица.}$$

Итак, $S(F_1) = M_{-1}X^{11}$.

Для умножения на матрицу необходимо перейти от степенного обозначения тестового набора к двоичному (или полиномиальному). Такое преобразование можно упростить, используя заранее вычисленные значения $X^1, X^2, X^4, X^8, X^{16}$ и т.д. по модулю $h(X)$ (эффективность такого упрощения увеличивается с ростом значения m): $X^1 \bmod h(X) = X^1, X^2 \bmod h(X) = X^2, X^4 \bmod h(X) = X^4, X^8 \bmod h(X) = X^3 + X^2 + 1, X^{16} \bmod h(X) = X^4 + X^3 + X + 1$.

Произвольный элемент X^j можно представить в виде произведения полиномов со степенями степени 2: $X^{11} = X^8 X^2 X^1$. Остаток от деления X^{11} на $h(X)$ будет равен остатку от деления на $h(X)$ произведения остатков сомножителей:

$(X^3 + X^2 + 1)X^2X^1 \bmod h(X) = (X^6 + X^5 + X^3) \bmod (X^5 + X^2 + 1) = X^2 + X + 1$, или в двоичном обозначении 00111.

При начальном значении РСЛОС ГТП α^5 значение сигнатуры $S(F) = 11100$.

Пример. Пусть $F_2 = \bar{x}_5 \bar{x}_4 x_3 x_2$ при тех же значениях $h(X), g(X)$ и том же начальном состоянии РСЛОС ГТП.

$$S(F_2)=M_{-1}(X^{19})+M_{-1}(X^{19}+1)=M_{-1}(X^{19}+X^{19}+1)=M_{-1}(00001)=10000.$$

$$\text{Пусть } F_3 = \bar{x}_5 \bar{x}_4 x_3 x_1 \quad F_4 = \bar{x}_4 x_3 x_2 x_1 \quad F_5 = \bar{x}_5 \bar{x}_4 x_3.$$

$$S(F_3)=M_{-1}(X^5 + X^5 + X)=M_{-1}(X)=M_{-1}(00010)=01000.$$

$$S(F_4)=M_{-1}(X^{11} + X^{11} + X^4)=M_{-1}(X^4)=M_{-1}(10000)=00001.$$

$$S(F_5)=M_{-1}(X^2 + X^2 + 1 + X^2 + X + X^2 + X + 1)=M_{-1}(0)=0.$$

Таким образом, в общем случае для $k=-1$ сигнатура конъюнкции с рангом $g=m-1$ равна произведению матрицы M_{-1} и X , где i - индекс отсутствующей переменной, уменьшенный на единицу; сигнатура конъюнкции с $g < m-1$ равна нулю.

Аналогично, для произвольных примитивных полиномов $h(X)$ и $g(X)$ степени m , корни которых связаны равенством $\beta = \alpha^{-3}$, для конъюнкции с рангом $g < m-2$ $S=M_{-3}(0)=0$. Сигнатура равна нулю в следующих случаях: $k=-5, g < m-3$; $k=-7, g < m-4$; $k=-9, g < m-3$; $k=-11, g < m-4$; $k=-13, g < m-4$; для произвольного $k < m-1-w$, где w - вес двоичной записи $-k$.

Если рассматривать полученный результат при конкретных значениях m , условие равенства сигнатуры нулю можно сформулировать иначе: $g < w[(k)_0]$, где $w[(k)_0]$ - вес двоичной записи k в обратном коде.

Таким образом, если неисправность в КС, описываемой функцией F , приводит к тестовой реакции F_{Π} , и в представлении $F+F_{\Pi}$ в виде полинома Жегалкина присутствуют только конъюнкции с рангом $g < w[(k)_0]$, то $S(F+F_{\Pi})=S(F)+S(F_{\Pi})=0$, или $S(F)=S(F_{\Pi})$, т.е. неисправность будет необнаруженной.

Итак, число w (вес двоичной записи $-k$) представляет собой параметр, с помощью которого можно оценить эффективность сигнатурного анализа при применении в качестве ГТП и АТР РСЛОС с порождающими примитивными полиномами одинаковой степени. Параметр w принимает минимальное значение 1 при $k=-1$, и максимальное значение $m-1$ при $k=1$.

Этот вывод также справедлив для многовходовых АТР и РСЛОС ГТП и АТР с альтернативной реализацией (с внешними сумматорами в цепях обратной связи).

Учитывая, что структура РСЛОС с приводимым образующим полиномом, который представляет собой произведение различных неприводимых полиномов, эквивалентна совокупности структур РСЛОС с неприводимыми полиномами - сомножителями, получаются следующие выводы. Максимальная эффективность компактного тестирования КС достигается для следующих вариантов:

1) АТР на основе кодов Хэмминга - одинаковые образующие полиномы РСЛОС ГТП и АТР;

2) АТР на основе кодов БЧХ - первый сомножитель образующего полинома РСЛОС АТР (минимальный полином, соответствующий элементу поля Галуа с наименьшей степенью j) должен быть равен

примитивному образующему полиному РСЛОС ГТП. Остальные сомножители - минимальные полиномы, соответствующие элементам поля Галуа со степенями $j+2$, $j+4$ и т.д.

3) многоходовый АТР на основе кодов Бартона - сомножитель образующего полинома РСЛОС АТР, соответствующий неприводимому полиному, должен быть равен примитивному образующему полиному РСЛОС ГТП;

4) многоходовый АТР на основе кодов Рида-Соломона, исправляющего одиночные ошибки - корень первого сомножителя образующего полинома РСЛОС АТР должен быть равен корню примитивного образующего полинома РСЛОС ГТП;

5) многоходовый АТР на основе кодов Рида-Соломона, исправляющего ошибки кратности t - корень первого сомножителя образующего полинома РСЛОС АТР (элемент поля Галуа со степенью j) должен быть равен корню примитивного образующего полинома РСЛОС ГТП; корни остальных сомножителей - элементы поля Галуа со степенями $j+1$, $j+2$, $j+3$ и т.д.

Выводы. Выполнен анализ эффективности компактного тестирования с локализацией ошибок в выходной тестовой реакции проверяемой комбинационной схемы. Предложен метод аналитического расчета сигнатур для исчерпывающего тестирования комбинационных схем при реализации генератора тестовых последовательностей и анализатора тестовых реакций на основе регистров сдвига с линейными обратными связями. Рассматриваются анализаторы тестовых реакций, построенные на основе циклических кодов Хэмминга, Боуза-Чоудхури-Хоквингема, Бартона, Рида-Соломона, позволяющие локализовать ошибки в тестовой реакции. Получена оценка меры эффективности компактного тестирования комбинационных схем, на основе которой предложены рекомендации по выбору порождающих полиномов регистров сдвига с линейными обратными связями для различных вариантов компактного тестирования с локализацией ошибок. Полученные результаты могут найти применение для встроенного самотестирования или внешнего тестового оборудования средств вычислительной техники

Список литературы

1. Richard E.Blahut. Algebraic Codes for Data Transmission/ Cambridge University Press, 2012. – 498 p.
2. Дяченко О.Н., Дяченко В.О. Альтернативный метод укорачивания циклических кодов // Электронные информационные системы. 2017. № 1 (12). С. 94–100.
3. Ярмолик В.Н., Калоша Е.П. Эффективность сигнатурного анализа в само-тестирующихся СБИС // Электрон. моделирование.- 1992.- 14,№3. - С.51-56.

4. Ершов А.Н., Петров С. В., Пятошин Ю.П., Коханько Д. В., Зяблов В.В. и др. Улучшение радиационной стойкости памяти с помощью помехоустойчивых кодов // Ракетно-космическое приборостроение и информационные системы. 2014, том 1, выпуск 4, с. 42–49.

5. Дяченко В.О., Дяченко О.Н. Циклическое кодирование цифровой информации на основе двойственных полиномов // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике: материалы II Международной научно-практической конференции (Азов, 19 мая 2015 г.) – Ростов н/Д, ДГТУ, 2015. – С. 71-76.

УДК 004.021

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДА РАСПОЗНАВАНИЯ ОБЪЕКТА НА ИЗОБРАЖЕНИИ С ПОМОЩЬЮ АЛГОРИТМА МУРАВЬИНЫХ КОЛОНИЙ

А.Ю.Плитка, К.А.Ручкин

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
dartin1@mail.ru

Аннотация

Плитка А.Ю. Ручкин К.А. Исследование и разработка метода распознавания объекта на изображении с помощью алгоритма муравьиных колоний. С развитием науки и техники, обилием графической информации и ее активным обменом, все более актуальным становится автоматическое распознавание объектов на изображениях с решением задачи выделения их контуров. Их практическое применение предполагает более высокие требования к методикам распознавания объектов на изображении. Основная сложность при решении данной задачи – учет различных дополнительных факторов на изображении: разнообразие фона, вариабельность различных частей изображения, зашумленность. В данной статье рассматривается возможность применения алгоритма муравьиных контуров для распознавания объектов на изображении, приводятся преимущества перед генетическим алгоритмом.

Ключевые слова: *распознавание объектов, муравьиные колонии, роевой алгоритм.*

Введение. Задача распознавания границ объектов на изображении является одной из основных для областей, занимающихся обработкой и

анализом изображений. Границы объектов содержат большую часть информации об изображении и исследования зрительной системы человека показывают, что края объектов изображения имеют особое значение, поскольку объекты могут быть определены всего лишь несколькими грубыми контурами.

Обычно технологии распознавания и классификация изображений включают в себя следующие этапы: предобработка изображения, определение границ изображения, извлечение признаков и автоматическая классификация. На практике цвет фона, размер объектов, зашумление, могут прервать определение границ; поэтому, большой проблемой является поиск методов, с высокой скоростью и устойчивостью к помехам.

Постановка задачи. Задачи определения границ изображений легко сводятся к задачам оптимизации графов. Поскольку в основе муравьиного алгоритма лежит моделирование передвижения муравьёв по некоторым путям графа, то такой подход является эффективным способом поиска рациональных решений для задач оптимизации, в первую очередь, на графах [1].

Генетический алгоритма для распознавания объектов на изображении. Генетические алгоритмы (ГА) являются одним из самостоятельных разделов теории искусственного интеллекта – эволюционных вычислений, которые основаны на математическом моделировании процессов биологической эволюции. В основе ГА лежат принципы естественной эволюции, сформулированные Дарвином, – естественный отбор, изменчивость, наследственность. С их помощью ГА моделируют генетические процессы, происходящие в биологических сообществах. Обычно генетические алгоритмы применяются для решения оптимизационных проблем. Предметная область ГА включает в себя проблемы комбинаторики, биоинформатики, теории игр и др. ГА применяются также для обработки и распознавания образов, в частности изображений.

В рамках данной проблемы ГА может быть использован как при подготовке изображения к распознаванию, в частности при фильтрации, сегментации, разметке сцены, так и непосредственно для его распознавания. В [4] рассматривается подход автоматической разметки сцены с помощью ГА. Комбинация использования семантических сетей для представления ограничений области и нечеткой логики для достижения соответствия меток этим ограничениям породили новую стратегию вычисления фитнес-функции для работы ГА. Показана возможность применения данного подхода для идентификации снимков облаков на мультиспектральных спутниковых снимках. Возможность применения адаптивного генетического алгоритма для решения проблемы сегментации цветного изображения, усложненной необходимостью принятия решения об оптимальном количестве сегментов и точного определения текстурных областей исследуется в [5]. Так как во

многих случаях при сегментации топологическим областям могут быть поставлены в соответствие области признаков, данную проблему можно сформулировать как оптимизационную и применить ГА для кластеризации небольших районов пространства признаков.

Применение алгоритма муравьиных колоний для распознавания объекта на изображении. Для того чтоб построить соответствующий муравьиный алгоритм для решения какой-либо задачи, необходимо:

1. Представить задачу в виде набора компонент и переходов или набором неориентированных взвешенных графов, на которых муравьи могут строить решения;

2. определить значение следа феромона;

3. определить эвристику поведения муравья, когда строим решение;

4. если возможно, то реализовать эффективный локальный поиск;

5. выбрать специфический муравьиный алгоритм и применить для решения задачи;

6. настроить параметры алгоритма.

Рассмотрим алгоритм кластеризации изображения, основанный на алгоритме муравьиных колоний. В этом алгоритме муравьи представляются простыми агентами, которые случайным образом перемещаются по дискретному массиву. Пиксели, которые рассеяны в элементах массива могут быть перемещены из одного элемента массива в другой, формируя кластеры.

Перемещение пикселей косвенным образом определяется поведением муравьев. Каждый муравей может присоединять или исключать пиксел согласно функции, которая вычисляет сходство пиксела с другими пикселями в кластере. Таким образом, муравьи динамически кластеризируют изображение на независимые кластеры, которые включают только сходные между собой пиксели. Кроме этого агент представляет новые вероятностные правила для присоединения или исключения пикселей, а так же стратегию локального перемещения для ускорения сходимости алгоритма.

Предположим, размерность массива равна N . Каждый элемент этого массива связан с гнездом муравьев в определенном порядке, что позволяет муравьям легко переходить от одного элемента к другому. В процессе алгоритма муравьи могут изменять, строить или убирать существующие кластеры пикселей. Кластер представляется соединением двух или более пикселей и пространственно расположен в отдельной ячейке массива, что упрощает его идентификацию.

Инициализируем N пикселей $\{P_1, \dots, P_n\}$ для кластеризации, расположенные в массиве таким образом, что каждый элемент массива связан только с одним пикселем. Каждый муравей a_i из колонии с K муравьями $\{a_1, \dots, a_k\}$ присоединяет случайно выбранный пиксел из элементов массива и возвращается в гнездо.

После начального этапа инициализации происходит этап кластеризации. Выбор муравья производится случайным образом. Он перемещается от своего гнезда к элементам массива и определяет с помощью вероятностного правила, присоединять ли этот пиксел к кластеру. Каждый муравей знает список расположения пикселов, которые не были присоединены другими муравьями. Случайным образом муравей определяет следующий пиксел из списка свободных. Этот алгоритм повторяется для каждого муравья. Алгоритм останавливается при прохождении заданного количества итераций.

Сравнение алгоритмов. Эксперименты, описанные в литературе, показывают, что генетические алгоритмы очень эффективны в поиске глобальных минимумов адаптивных рельефов, так как в них исследуются большие области допустимых значений параметров нейронных сетей. (Градиентные алгоритмы дают возможность находить только локальные минимумы.) Другая причина того, что генетические алгоритмы не застревают в локальных минимумах — случайные мутации, которые аналогичны температурным флуктуациям метода имитации отжига. Естественно генетические алгоритмы не свободны от недостатков. К ним можно отнести прежде всего следующие. Конфигурация для решения сложных реальных задач не очевидна. Для решения конкретной задачи необходимо выбрать или разработать представление (кодирование) потенциального решения. Существует также проблема определения фитнес-функции. Есть проблема выбора параметров, таких как мощность популяции, вероятности генетических операторов и т.д. Нет эффективных критериев окончания работы алгоритма.

В сравнении с генетическими алгоритмами муравьиные алгоритмы имеют некоторые преимущества: базируются на памяти обо всей колонии вместо памяти только о предыдущем поколении; меньше подвержены неоптимальным начальным решениям (из-за случайного выбора пути и памяти колонии) [6,7]. К недостаткам относятся следующие свойства муравьиных алгоритмов: сходимость гарантируется, но время сходимости неизвестно; обычно необходимо применение дополнительных методов, таких как локальный поиск; сильно зависят от настроечных параметров, которые подбираются только исходя из экспериментов.

Выводы. Алгоритм муравьиных колоний имеет огромный потенциал для решения задач, связанных с обработкой изображений, включая определение границ, извлечение признаков, сегментацию, сжатие изображения. Вариации алгоритма для решения задачи определения объекта на изображении были рассмотрены. Так же в данной статье был произведен общий анализ алгоритма муравьиных колоний, используемого для обработки изображений, были выявлены его преимущества и недостатки по сравнению с генетическим алгоритмом. Все это дает направление для более подробного исследования данной задачи, и попытке применить данный алгоритм на практике.

Список литературы

1. Dorigo M, Vittorio M, Alberto C. Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 1996; 26(1):29–41.
2. Dorigo M, Thomas S. Ant Colony Optimization: Overview and Recent Advances. Techreport, IRIDIA, Universite Libre de Bruxelles, 2009.
3. Tabakhi S, Parham M, Fardin A. An Unsupervised Feature Selection Algorithm based on Ant Colony Optimization, Engineering Applications of Artificial Intelligence. 2014; 32:112-23.
4. Khanna K, Navin R. Reconstruction of Curves From Point Clouds using Fuzzy Logic and Ant Colony Optimization, Neurocomputing. 2015; 161:72–80.
5. Aghdam Mehdi H, Nasser GA, Mohammad EB. Text Feature Selection using Ant Colony Optimization, Expert Systems with Applications. 2009; 36(3):6843–53.
6. Jun S, Dong H. Statistical Theory of Edge Detection, Computer Vision, Graphics, and Image Processing. 1998; 43(3):337–46.
7. Barbosa H. Ant Colony Optimization: Techniques and Applications. In Tech, 2014.

УДК 519.217.2+575.1

АНАЛИЗ ПРИМЕНЕНИЯ МАРКОВСКИХ ЦЕПЕЙ ПРИ МОДЕЛИРОВАНИИ ПРОГНОЗА ГЕНОТИПА ПОТОМСТВА

Ганенко А.И.

Донецкий национальный технический университет,
кафедра компьютерного моделирования и дизайна
E-mail: kb-ganenko.ua@yandex.ua

Аннотация

Ганенко А.И. Анализ применения Марковских цепей при моделировании прогноза генотипа потомства. В статье рассмотрено применение Марковских цепей при анализе вероятности проявления доминантных или рецессивных признаков у потомства на основе теории наследственности Г. Менделя, а также представлены конкретные примеры прогнозирования в генетике.

Ключевые слова: марковские цепи, генотип, наследственность, матрица переходов.

Проблематика исследования. Человечество всегда пугала неизвестность, в силу этого во все времена люди пытались предугадать

возможность наступления каких-либо событий и в случае необходимости поспособствовать их предотвращению. С развитием глубины знаний, человечество добилось колоссальных успехов в различных областях науки, и ученые достигли высоких результатов в сфере прогнозирования. Были разработаны алгоритмы, позволяющие смоделировать различные процессы. В частности, в биологии нашли применения информационные технологии, выделившиеся в отдельное направление – биоинформатику (совокупность математических методов компьютерного анализа, разработку алгоритмов и программ для сложных биологических систем). Инструменты биоинформатики помогают в сравнении генетических и геномных данных и, в целом, в понимании эволюционных аспектов молекулярной биологии [1 - 4].

Понятие Марковской цепи. Цепь Маркова – последовательность случайных событий с конечным или счётным числом исходов, характеризующаяся тем свойством, что, говоря нестрого, при фиксированном настоящем будущее независимо от прошлого. Поскольку система может пребывать в одном из n состояний, то для каждого момента времени t необходимо задать список вероятностей перехода P_{ij} , который удобно представить в виде матрицы [1 - 3].

Использование Марковских цепей в генетике. Марковские цепи получили широкое применение в прогнозировании, в частности, в области генетики [3, С.47]. При рассмотрении данного вопроса будем базироваться на теории наследственности Грегора Менделя, который проводил эксперименты с растениями гороха. Результаты его работы привели к открытию двух типов генов – доминантных (d) и рецессивных (r). Таким образом при наличии двух доминантных генов появляется доминантный тип признака, двух рецессивных – рецессивный тип признака. Мендель рассматривал цвет стручка гороха, так как этот признак является управляемым двумя генами: ген, дающий зеленый цвет стручку – доминантный, а желтый – рецессивный. Так, dd – растение имеет зеленые стручки, а rr – желтые, однако зеленый цвет имеют и растения с одним доминантным и одним рецессивным геном, т.е. dr и rd – растения. В этом случае растения имеют так называемый одинаковый фенотип (внешний вид), но различные генотипы (генетический набор). Согласно Г. Менделю при скрещивании двух особей потомство получает по одному гену от каждого родителя, и сам выбор проходит случайным образом, у каждого родителя независимо. Данную теорию можно рассматривать, как математическую модель. Она строится на основе реальных данных, на основе экспериментов Менделя. Необходимой составляющей здесь является анализ результата, вытекающего из модели. Если скрещивать большое количество чистых доминантов и чистых рецессивов, то примерно $3/4$ потомков проявляют

доминантные признаки. Однако, в случае одноразового повторения опыта, возможно, что ни один из потомков не проявит доминантный признак. Таким образом можно говорить о том, что в естественных явлениях всегда существуют колебания, хотя в настоящее время нет достаточно глубоких теорий и моделей для понимания протекающих не детерминированных единичных процессов.

Для получения существенных выводов на основе модели Менделя рассмотрим процесс непрерывного скрещивания с гибридом. Предположим, что мы располагаем особью неизвестного генетического происхождения и скрестим ее с известным гибридом, например, для наглядности можно взять признак «цвет глаз» человека. В данном случае карие глаза – это доминантный признак, а голубые – рецессивный, и необходимо выяснить: “Какова вероятность, что потомок, взятый из t -го поколения будет иметь карий цвет глаз?” Ответ кроется в применении теории цепей Маркова.

Формирование Марковской модели сводится здесь к построению цепи, состояниями которой являются генотипы для особи, выбранной из t -го поколения. Таким образом, имеем три состояния: D (для чистых доминантов), H (для гибридов), R (для чистых рецессивов). Переходные вероятности для данной цепи задаются матрицей \mathbf{P} , каждый элемент которой определяет вероятность того, что, если человек генетического типа i скрещивается с известным гибридом, то потомок, выбранный случайным образом, будет иметь генетический тип j . Например, элемент DD матрицы равен $\frac{1}{2}$, так как при скрещивании чистого доминанта (человека имеющего гены, отвечающие только за карий цвет глаз) с гибридом вероятность получения потомка – чистого доминанта равна $\frac{1}{2}$, и так далее, по аналогии.

$$\begin{array}{c}
 \begin{array}{ccc}
 D & H & R
 \end{array} \\
 \begin{pmatrix}
 \frac{1}{2} & \frac{1}{2} & 0 \\
 \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
 0 & \frac{1}{2} & \frac{1}{2}
 \end{pmatrix}
 \end{array}$$

$\mathbf{P} =$

Далее на рисунке 1 приведен оргграф соответствующий матрице \mathbf{P} .

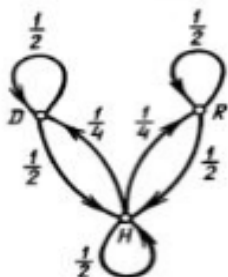


Рис.1 – Ориентированный граф матрицы P.

Из рисунка видно, что граф является сильно связным. Это означает, что в такой системе возможен переход из любого состояния (D_i, H_i, R_i) в любое состояние (D_j, H_j, R_j) за конечное число шагов. Такая цепь называется эргодической. А поскольку данная цепь состоит из одного циклического класса, то она является регулярной. Выполнив дополнительные вычисления находим стационарный вероятностный вектор матрицы P. Он имеет

следующий вид: $\mathbf{W} = \left(\frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{4} \right)$. Это дает основание считать, что вероятность того, что в t -м поколении определенная особь будет чисто доминантной равна $\frac{1}{4}$, чисто рецессивной также $\frac{1}{4}$, и вероятность появления гибридной особи, – $\frac{1}{2}$. Следовательно, выбранный случайным образом человек с вероятностью $\frac{3}{4}$ будет обладать доминантным признаком, – иметь карий цвет глаз.

При помощи аналогичного построения Марковской цепи можно спрогнозировать и ответ на вопрос: «За какое число поколений произойдет поглощение тех или иных признаков? Рассмотрим процесс непрерывного скрещивания с доминантом. Снова имеем цепь Маркова с тремя состояниями (D, H, R), в таком случае матрица переходов имеет следующий вид:

$$\mathbf{P} = \begin{matrix} & \begin{matrix} D & H & R \end{matrix} \\ \begin{matrix} D \\ H \\ R \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

Это переходная матрица для поглощающей цепи Маркова, то есть она содержит невозвратное состояние (D) [2, С.390]. Это означает, что попав в данное состояние движение останавливается. Следовательно, для него практически все потомки будут чистыми доминантами, и их внешние признаки также будут доминантными.

Матрица \mathbf{P} имеет канонический вид и существует такая матрица \mathbf{Q} , которая указывает переходные вероятности, определяющие переходы из непоглощающих состояний в непоглощающие [4, С.239]:

$$\mathbf{Q} =$$

Далее вычисляется матрица \mathbf{N} – фундаментальная матрица поглощающей цепи Маркова, такая что $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$ [4, С.241]:

$$\mathbf{N} = \begin{matrix} H & R \\ H & R \end{matrix} \begin{pmatrix} 2 & 0 \\ 2 & 1 \end{pmatrix}$$

Сумма элементов в соответствующей строке матрицы \mathbf{N} показывает среднее число поколений до поглощения, и можно прогнозировать, что если исходная особь обладала гибридным генотипом, то поглощение произойдет за два поколения, а если у неё рецессивный генотип, – за три.

Выводы. В данном исследовании было рассмотрено понятие Марковской модели. Основное внимание было уделено построению и анализу Марковских цепей. Данные модели, позволяют проводить прогнозирование проявления тех или иных признаков у человека взяв за основу данные о доминантном или рецессивном типе генов, отвечающих за выбранный признак.

Перспективы дальнейших исследований в данном направлении заключаются в использовании математического аппарата теории цепей Маркова для детального прогнозирования наследственности человека из выводов теории Г. Менделя.

Список литературы

1. Википедия свободная энциклопедия / Интернет-ресурс. – Режим доступа: www/ URL: <https://ru.wikipedia.org>.
2. В. Феллер. Введение в теорию вероятностей и её приложения, т. 1. М.: "Мир", 1964, - 499 с.
3. С. Карлин. Основы теории случайных процессов. М.: "Мир", 1971, – 537 с.
4. Ф.С. Робертс Дискретные математические модели с приложениями к социальным биологическим и экологическим задачам. М.: "Наука", 1986, — 498с.

УДК 621.311.25

ВЛИЯНИЕ ЗАТЕНЕНИЯ НА СОЛНЕЧНУЮ ЭЛЕКТРОСТАНЦИЮ

Н.В. Должиков, А.В. Левшов

Донецкий национальный технический университет, г. Донецк,
кафедра электроснабжения промышленных предприятий и городов
nktd1116@gmail.com

Аннотация.

Должиков Н.В., Левшов А.В. Влияние затенения на солнечную электростанцию. В работе рассмотрен вопрос моделирования солнечной панели в среде Matlab/Simulink. Модель разработана с использованием основных уравнений фотоэлектрических солнечных элементов, которые учитывают воздействия солнечной инсоляции и перепада температур. Рассмотрена так же проблема затенения фотоэлементов и его влияние на работу СЭС.

***Ключевые слова.** Фотоэлемент, модель солнечной батареи, вольт - амперная характеристика, затенение, шунтирующий диод.*

***Цель статьи.** На основании разработанной математической модели солнечной панели в среде Matlab/Simulink рассмотреть проблему затенения фотоэлементов и его влияния на работу СЭС.*

***Введение.** В последние годы заметно усилилось внимание к поиску и освоению нетрадиционных источников энергии, которые отличаются от ископаемых органических ресурсов своими громадными запасами, т.е. они практически неисчерпаемы или периодически возобновляются. Загрязнение окружающей среды, рост цен на энергоносители и уменьшение их запасов обосновывают использование экологически чистых и, возможно, более дешевых источников энергии. Таковыми являются, так называемые, возобновляемые источники энергии (ВИЭ). Одним из наиболее перспективных видов ВИЭ является энергия солнца. Солнце, по человеческим меркам, – неисчерпаемый источник энергии. Существуют устройства, преобразующие солнечную энергию в электрическую. Это солнечные элементы. Для создания солнечных электростанций (СЭС) солнечные элементы объединяют в батареи и крепят на основании для удобства их конечного монтажа и увеличения прочности [1]. Такая конструкция называется солнечной панелью, или солнечным модулем. В нынешнее время производство таких солнечных панелей достигло значительных масштабов. Относительно доступные солнечные модули имеют коэффициент преобразования солнечного света в электричество 7-18 %. Максимальный же коэффициент преобразования на сегодняшний день около 40 %. Но дороговизна высокопроизводительных*

элементов не позволяет использовать их для создания электростанций широкого потребления, поскольку это экономически нецелесообразно.

Описание структуры СЭС.

Основой солнечной электростанции является фотоэлектрический преобразователь. Принцип работы фотоэлектрического преобразователя достаточно прост и заключается в следующем.

При освещении солнечным светом единственный кремниевый фотоэлектрический преобразователь генерирует электрическое напряжение величиной около 0,5 В. Независимо от типа и схемы включения все (большие и малые) кремниевые солнечные элементы генерируют напряжение 0,5 В, при отсутствии нагрузки и номинальном значении солнечной инсоляции.

Для увеличения выработки энергии отдельные ФЭП собираются вместе и образуют фотоэлектрические модули, которые затем соединяются в массивы (рис. 1) для увеличения выходного напряжения фотоэлементы соединяют последовательно, а для увеличения мощности - параллельно. Модульная структура ФЭП позволяет строить различные фотоэлектрические системы в зависимости от выработки энергии для различного применения. Таким образом, фотоэлектрический элемент является составным элементом фотоэлектрических систем.

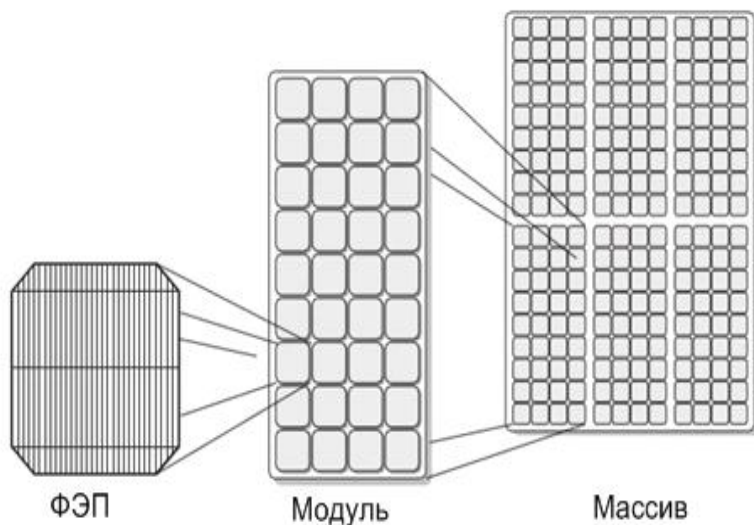


Рисунок 1 – Фотоэлектрический элемент, модуль и массив ФЭП [2]

Последовательные модули ведут себя следующим образом: они нагреваются при протекании тока и могут выйти из строя. Для защиты используются обратные диоды (как показано на рис. 2).

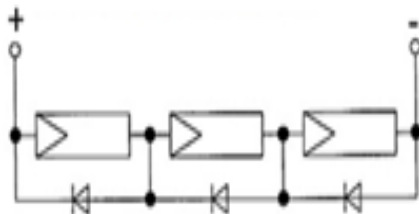


Рисунок 2 – Последовательное соединение солнечных модулей с обратными диодами [4]

При параллельном соединении и наличии затенения отдельного элемента он запирается и становится нагрузкой для остальных элементов, в результате этого затененный элемент перегревается и может выйти из строя. Для защиты от этого режима в каждой ветви устанавливают запирающие диоды (рис. 3).

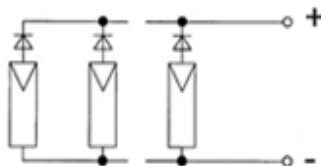


Рисунок 3 – Расположение запирающих диодов при параллельном соединении [4]

На рис. 4 показана схема комбинированного фотоэлектрического модуля с запирающими и обратными диодами.

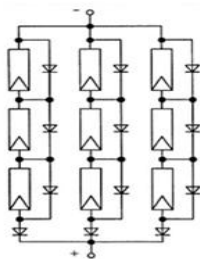


Рисунок 4 – Фотоэлектрический генератор с обратными и запирающими диодами [4]

Эффективным способом решения проблемы затенения является параллельное подключение шунтирующих диодов ко всем элементам, как это показано на рис. 5. Диоды подключены так, что при работе солнечного элемента они обратно смещены напряжением самого элемента. Поэтому через диод ток не протекает, и батарея функционирует нормально.

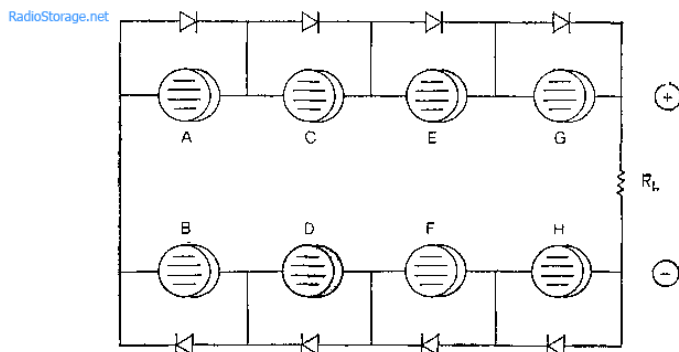


Рисунок 5 - Параллельное подключение шунтирующих диодов [4]

Схема работает следующим образом. Предположим, что один из элементов затеняется. При этом диод оказывается прямо смещенным, и через него в нагрузку протекает ток в обход неисправного элемента. Конечно, выходное напряжение всей цепочки уменьшится на 0,5 В, но устранится источник разрушительной силы для затененного фотоэлемента.

Дополнительное преимущество такой схемы состоит в том, что батарея продолжает нормально функционировать. Без шунтирующих диодов она бы полностью вышла из строя.

На практике нецелесообразно шунтировать каждый элемент батареи. Необходимо руководствоваться соображениями экономии и использовать шунтирующие диоды, исходя из разумного компромисса между надежностью и стоимостью.

Как правило, один диод используют для защиты 1/4 батареи. Таким образом, на всю батарею требуется всего 4 диода. В этом случае эффект затенения будет приводить к 25%-ному (вполне допустимому) снижению выходной мощности.

Моделирование солнечной батареи.

Электрическая схема (рис. 6), получившая название “модель одного диода”, отображает элемент солнечной батареи. Данная модель состоит из генератора тока, параллельно которому подключены диод и шунтирующий резистор R_{sh} (параллельное сопротивление). Помимо них к одному из выводов генератора тока подключено последовательное сопротивление R_s [3].

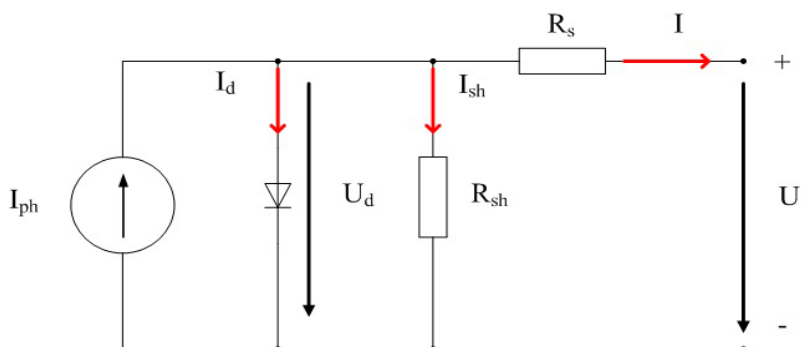


Рисунок 6 – Электрическая модель (схема замещения) солнечной батареи

Элементы схемы замещения (рис. 6) обозначены следующим образом:

- I_{ph} – фотогенерируемый ток [A],
- I_d – диодный ток [A],
- U_d – диодное напряжение [B],
- I – выходной ток [A],
- U – напряжение на клеммах [B],
- I_{sh} – шунтирующий ток [A],
- R_{sh} – параллельное сопротивление [Ом],
- R_s – последовательное сопротивление [A].

Классическая модель полупроводникового фотоэлемента состоит из соединенных параллельно источника фототока и шунтирующего диода. Моделирование солнечной панели производится с использованием элементов программы MATLAB/Simulink, которая представлена на рис. 7

На основе данной модели были выполнены расчеты вольтамперных характеристик солнечной батареи типа для условий различной освещенности, при температуре окружающей среды 25 градусов Цельсия.

На рис. 8 представлены расчетные характеристики тока и мощности солнечной панели для уровня освещенности солнечного элемента $E = 1000$ Вт/м² (при стандартных условиях). Из рисунка видно, что наибольшая эффективность солнечной панели возникает при нагрузке соответствующей фиксированному положению рабочей точки максимальной мощности.

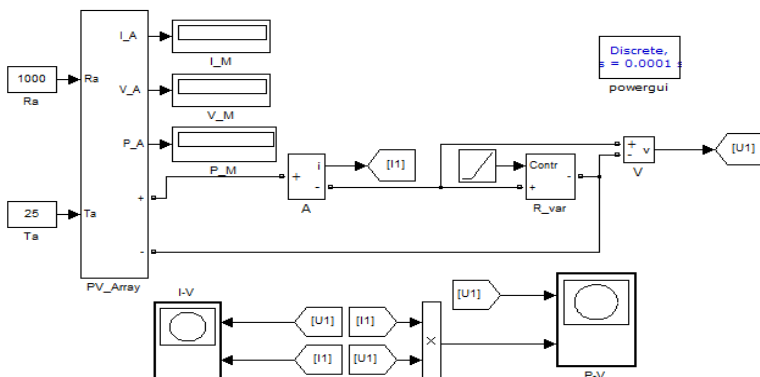


Рисунок 7 - Модель солнечной батареи в MATLAB/Simulink

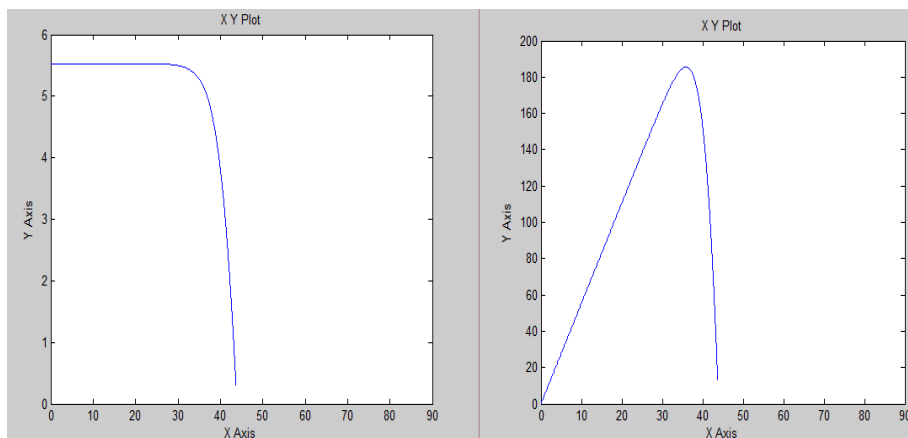


Рисунок 8 – ВАХ и ВВХ в нормальном режиме

Затенение моделировалось при снижении уровня инсоляции. В затененном состоянии модуль также преобразовывает энергию и может отдавать ее в общую цепь, но вольт - амперная характеристика солнечного модуля (рис.9) не позволяет использовать его с максимальным КПД, поскольку рабочая точка затененного элемента находится в районе нулевой мощности, и вся преобразованная энергия рассеивается в модуле.

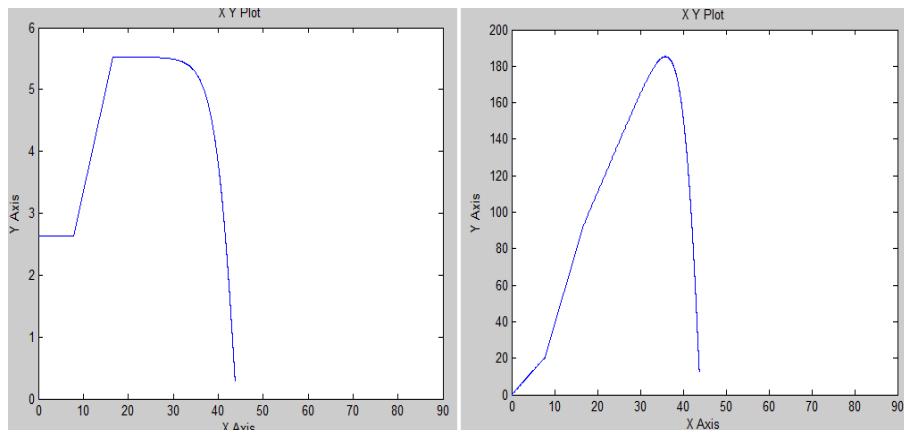


Рисунок 9 – ВАХ и ВВХ характеристики при затененном фотоэлементе

Выводы. В статье рассмотрена проблема затенения фотоэлементов и его влияние на работу СЭС, описаны меры предотвращения не эффективной работы СЭС и повышения ее эффективности при условиях различной освещенности. Также рассматривается модель солнечной панели в среде Matlab/Simulink, на основании которой были выполнены расчеты модельных вольтамперных характеристик солнечной батареи для условий различной освещенности.

Список литературы.

1. Алфёров Ж. И., Андреев В. М., Румянцев В. Д. Тенденции и перспективы развития солнечной энергетики // Физика и техника полупроводников. 2004. Т. 38, № 8. С. 937–948.
2. Фролкова Н.О. Компьютерное моделирование вольтамперных характеристик солнечных батарей /И. В. Абраменков, Н.О. Фролкова // Тезисы докладов XIV международной научно-технической конференции студентов и аспирантов. 2008. С. 381–383
3. Шарифов Б.Н., Трегулов Т.Р. Моделирование солнечной панели в программе Matlab/Simulink. УГАТУ, Т.19, №4 (70), 2015. С. 77-83.
4. Основные принципы работы солнечных элементов и их включение [электронный ресурс]. // Радиоэлектроника и светотехника [сайт] [2009]. URL: <http://radiostorage.net/?area=news/1308>

УДК 514.18

АВТОМАТИЗАЦИЯ РЕШЕНИЯ ЗАДАЧ ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ КАК ИНСТРУМЕНТ РАЗРАБОТКИ УЧЕБНЫХ ПОСОБИЙ И ОБУЧАЮЩИХ СИСТЕМ

Карабчевский В.В., Кудимов М.И.

Донецкий национальный технический университет
кафедра компьютерного моделирования и дизайна

Е-mail: kudimov96@mail.ua

Аннотация

Карабчевский В.В., Кудимов М.И. Автоматизация решения задач трехмерного моделирования как инструмент разработки учебных пособий и обучающих систем. Поставлена задача получения иллюстраций и моделей для использования в методических материалах и обучающих системах. Для ее решения предложена разработка системы автоматического решения геометрических задач. Приведены примеры работы этой системы, предложено направление ее дальнейшего развития.

Ключевые слова: решение геометрических задач, геометрическое моделирование, геометрические фигуры, визуализация.

Постановка задачи. Существует множество учебников, пособий, и других методических материалов, описывающих решение базовых геометрических задач, как печатных, так и мультимедийных. Наиболее трудоемким этапом их разработки является создание качественных иллюстраций. В мультимедийных обучающих системах иллюстрации могут быть заменены трехмерными моделями, соответствующими как отдельным этапам построения, так и показывающими окончательный результат, что еще более трудоемко.

Существующие алгоритмы решения таких задач обеспечивают возможность автоматизации соответствующих построений как на комплексном чертеже, так и в пространстве, поэтому разработка системы, позволяющей решать некоторые классы задач при произвольных начальных условиях, является реальной и актуальной. Такая система должна обеспечивать возможность запоминания результатов построений на определенных заранее этапах, что позволит обучаемому просмотреть результат, а разработчику учебных материалов скопировать геометрическую модель в ее промежуточном состоянии.

Базовым инструментом геометрического моделирования может послужить AutoCAD, а средством реализации алгоритмов решения задач – язык AutoLISP.

Исследования. Разрабатываемая система предназначена для решения подмножества геометрических задач сечения линейчатых поверхностей, а именно сечения плоскостью прямого кругового цилиндра на комплексном чертеже и в пространстве в соответствии введенным данным пользователя. Секущая плоскость может пересекать основание фигуры.

Процесс построения разбивается на несколько шагов. Состояние модели на каждом шагезапоминается, что дает возможность просмотреть результаты построения и ознакомиться комментариями, которые были взяты из учебного пособия и несколько дополнены. Процесс построения соответствует алгоритму, описанному в [1], но разбит на большее число шагов для повышения наглядности и логичности изложения материала.

Приведем пример. На рисунках 1 и 2 приведены иллюстрации из пособия, соответствующие первому и второму шагам построения сечения цилиндра.

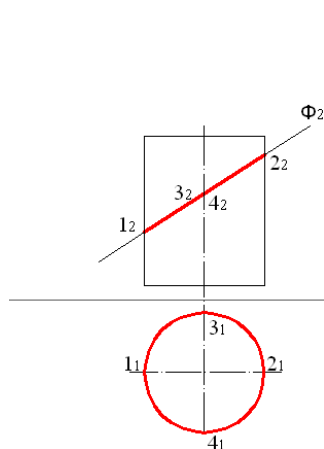


Рис. 1. Первый шаг построения.
Построены проекции цилиндра
и след фронтально-проецирующей
плоскость Φ_2

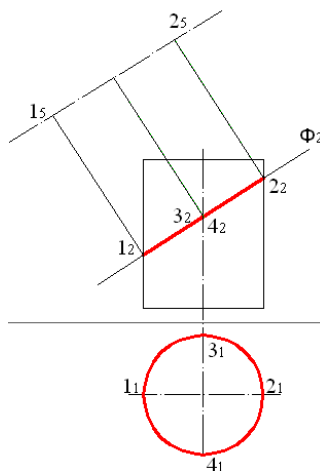


Рис. 2. Второй шаг построения.
Построены проекции большой
оси и центра эллипса

Сразу можно отметить недостаток рисунков, выполненных традиционно в бесосной системе координат. Это усложняет понимание того, как расположен объект в текущей системе координат. В разрабатываемой на языке AutoLISP[2] системе этот недочет устранен. После её запуска оси координат

автоматически настраиваются перед началом построения и устанавливаются соответствующие обозначения.

При разработке системы было принято решение разбить второй шаг, соответствующий рисунку 2, еще на две части. Первая часть, что станет в системе вторым шагом, будет описывать как использовать команду `_OFFSET` для получения отрезка параллельного фронтально проецирующей плоскости Φ . Вторая часть, что станет уже третьим шагом, расскажет о том, как при помощи команд `_LINE` и `_PERPEND` построить перпендикуляры от Φ_2 и до осевой линии.

Примеры работы системы представлены на рисунках 3-5.

ЭТАП 1. ИСХОДНОЕ ПОСТРОЕНИЕ

Построили проекции цилиндра и фронтально проецирующую плоскость Φ_2 по введенным данным пользователя

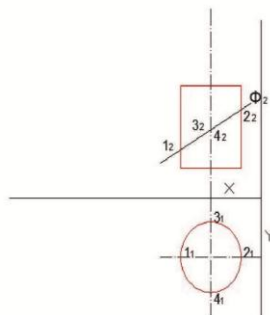


Рис. 3. Шаг 1. Исходные данные

ЭТАП 2. КОМАНДА `_OFFSET`

При помощи команды `_OFFSET` построим отрезок параллельно прямой Φ_2 . Для этого в командную строку вводим команду `_OFFSET` и жмем Enter. Далее вводим значение примерно равное двум высотам цилиндра. Затем выбираем прямую Φ_2 . Переводим курсор мыши выше прямой Φ_2 и жмем один раз правой кнопкой мыши. И теперь нужный нам отрезок будет построен.

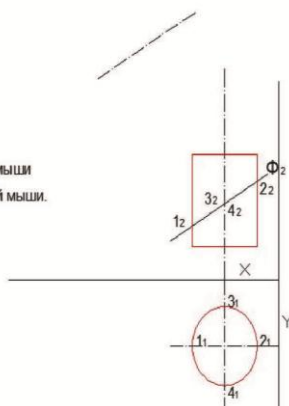


Рис. 4. Шаг 2. Результат применения команды `_OFFSET`

ЭТАП 3. КОМАНДЫ `_LINE` И `_PERPEND`

При помощи команд `_LINE` и `_PERPEND` построим перпендикуляры от проекции Φ_2 и др построенного отрезка на этапе 2.
Для этого вводим в командную строку команду `_LINE` и жмем Enter. На запрос командной строки ввести первую точку, вводим команду `_PERPEND`. Затем выбираем отрезок построенный на этапе 2. После этого указываем точку куда строить перпендикуляр. Таким образом строим 3 перпендикуляра.

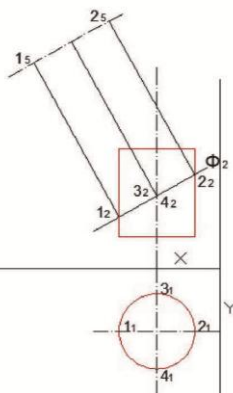


Рис. 5. Шаг 3. Результат работы команд `_LINE` и `_PERPEND`

На этих рисунках мы видим уже три шага построения, которые в пособии были представлены как два. Таким же образом были разбиты на несколько частей и другие шаги построения.

Приведем еще два рисунка демонстрирующие работу разрабатываемой системы. На рисунке 6 изображено уже конечное построение поверхностного сечения цилиндра на комплексном чертеже, а на рисунке 7 это же построение, но уже в пространстве

ЭТАП 6. КОМАНДА `_ELLIPSE`

При помощи команды `_ELLIPSE` построим эллипс который и будет нашим искомым сечением.
Для этого вводим в командную строку команду `_ELLIPSE` и жмем Enter. Далее первой точкой выбираем точку 45, второй точкой 35, третьей точкой 25 и наш эллипс будет построен. Таким образом мы нашли искомое сечение цилиндра на комплексном чертеже.

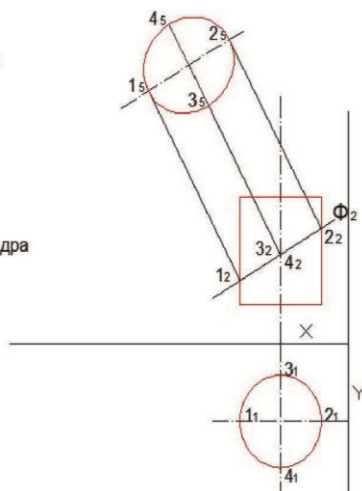


Рис. 6. Сечение цилиндра на комплексном чертеже

ЭТАП 9. КОМАНДА _SECTION

При помощи команды `_SECTION` получим искомое сечение цилиндра в пространстве. Для этого вводим в командную строку команду `_SECTION` и жмем `Enter`. Далее выбираем наш цилиндр и жмем снова `Enter`. Теперь мышкой выбираем три вершины секущей плоскости образованной на этапе 8 и наше сечение будет построено. Таким образом мы отыскали поверхностное сечение цилиндра в пространстве.

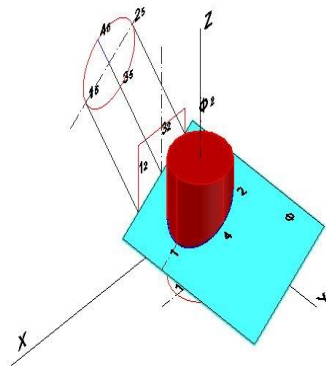


Рис. 7. Сечение цилиндра в пространстве

В учебном пособии только вкратце описано, что такое построение, как на рисунках 8 и 9, применяется в том случае, когда секущая плоскость пересекает основание фигуры. Подробное описание алгоритма решения такой задачи там не приводится. В разрабатываемой системе этот недостаток устранен. Студент получит решение задачи и на комплексном чертеже, и в пространстве, а также ему будет предоставлено подробное пояснение каждого шага построения.

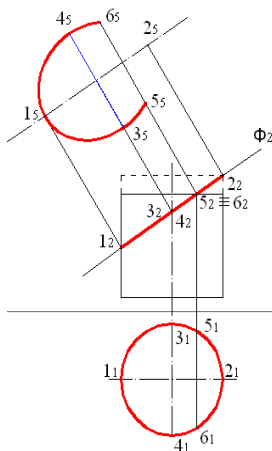


Рис. 8 Эллиптическая дуга

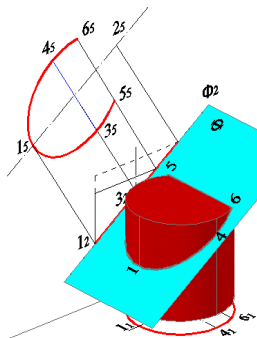


Рис. 9 Сечение в пространстве

Выводы. Ограничение на объем вышеупомянутого учебного пособия не позволило достаточно полно проиллюстрировать процесс решения некоторых геометрических задач на комплексном чертеже. Построения в пространстве и некоторые частные случаи также целесообразно иллюстрировать более подробно.

Разрабатываемая система позволяет ускорить получение таких иллюстраций пока только для задач сечения плоскостью некоторых видов поверхностей. В дальнейшем система будет дорабатываться путем реализации возможностей работы с другими видами поверхностей и расширения набора решаемых геометрических задач.

Список литературы

1. Карабчевский В.В. Методы компьютерной геометрии Донецк: ГВУЗ «ДонНТУ», Технопарк ДонНТУ «УНИТЕХ», 2010. – 178 с.
2. АВТОЛИСП - язык графического программирования в системе AutoCAD [электронный ресурс] // ВВЕДЕНИЕ В ЯЗЫК АВТОЛИСП [сайт]. URL: http://kappasoft.narod.ru/info/acad/lisp/a_lisp.htm#2

УДК 536.2

ИДЕНТИФИКАЦИЯ ТЕМПЕРАТУРОПРОВОДНОСТИ В НЕПРЕРЫВНОМ МЕТАЛЛИЧЕСКОМ СЛИТКЕ

В.К. Толстых, В.Е. Бодряга

Донецкий национальный университет, г. Донецк

кафедра компьютерных технологий

bodryaga@gmail.com

Аннотация

Толстых В.К., Бодряга В.Е. Идентификация температуропроводности в непрерывном металлическом слитке. Рассматривается идентификация температуропроводности в вертикальном цилиндрическом непрерывном слитке. Температуропроводность представляется в виде полиномиальной зависимости от температуры.

Ключевые слова: температуропроводность, параметрическая идентификация, критерий качества.

Постановка проблемы. Производство качественных слитков является актуальной проблемой. Она может решаться поиском новых технологий разлива, варьированием состава металла и температурно-временных режимов и др. По мере развития математического аппарата, стал возможен поиск наилучшего решения поставленных задач без дорогостоящих экспериментальных разливок. Однако для максимально точного учета параметров физического процесса, требуется как подготовка специалиста для выполнения математических операций, так и значительные машинные ресурсы.

Но даже в этом случае есть вероятность получения неточных результатов. Это связано с тем, что в теплофизические и гидродинамические уравнения входят эмпирические и полумпирические коэффициенты моделей. Использование таких параметров уравнений, незнание оптимальных значений этих коэффициентов для каждого конкретного объекта может стать причиной существенных погрешностей.

Задача параметрической идентификации. Будем моделировать установившиеся тепловые процессы в цилиндрическом непрерывном слитке с использованием эффективных теплофизических параметров, посредством следующего уравнения теплопроводности с соответствующими граничными условиями [1]:

$$V \frac{\partial T}{\partial z} = \frac{1}{r} \frac{\partial}{\partial r} \left(r a \frac{\partial T}{\partial r} \right), \quad (r, z) \in \Sigma = [0, R] \times [0, Z], \quad (1)$$

$$T|_{\substack{0 \leq r \leq R \\ z=0}} = T_0, \quad \left. \frac{\partial T}{\partial r} \right|_{\substack{r=0 \\ 0 < z < Z}} = 0,$$

$$T|_{\substack{r=R \\ 0 \leq z \leq z_c}} = T_c, \quad a \left. \frac{\partial T}{\partial r} \right|_{\substack{r=R \\ z_c < z \leq Z}} = -\alpha'(T - T_v), \quad (2)$$

где V – скорость литья, T – температура слитка, a – эффективный коэффициент температуропроводности, R – эффективный радиус слитка, Z – длина вертикальной части МНЛЗ, T_c – температура слитка в зоне кристаллизатора, T_0 – температура заливаемого в установку металла, z_c – нижняя граница кристаллизатора, T_v – температура охладителя в зоне вторичного охлаждения (ЗВО), $\alpha' = \frac{\alpha}{c\rho}$, α – коэффициент теплоотдачи в ЗВО, c – теплоемкость, ρ – плотность.

Предполагаем, что все параметры модели (1), (2) за исключением температуропроводности, заданы точно. Это означает, что все упрощения и погрешности модели учитываются эффективной температуропроводностью a .

Представим температуропроводность в виде полиномиальной зависимости от температуры $a(T) = \sum_{i=0}^n c_i T^i$, где вектор коэффициентов $\mathbf{c} = \{c_0, c_1, \dots, c_n\}$, который и необходимо найти из минимума расхождения модельной температуры T и экспериментально измеренной T_e на конкретной установке в виде:

$$J(\mathbf{c}) = \int_0^Z [T(r, z) - T_e(z)]^2 2\pi dz \rightarrow \min. \quad (3)$$

При решении задач такого типа может возникнуть проблема масштабируемости.

Как показали расчёты температурных режимов по (1), (2), коэффициенты вектора \mathbf{c} в степенном ряду $a(T)$, оказывают разное влияние на решение $T(r, z)$. Для выравнивания чувствительности модели (1), (2) к искомым коэффициентам было выполнено следующее масштабирование [2]:

$$a \equiv a(T) = \sum_{i=0}^n \frac{1}{T_c^i} c_i T^i, \quad (3)$$

где $\frac{1}{T_c^i}$ - коэффициент масштабирования, T_c - характерная температура процесса – температура затвердевания слитка. В нашем случае принималась $T_c = T_{sol}$.

При этом задача идентификации модели (1), (2) сводится к задаче параметрической идентификации вектора $\mathbf{c} = \{c_0, c_1, \dots, c_n\}$ размерности $n + 1$.

Численный расчет значения температуропроводности является важным шагом на пути к точному решению задач теплопроводности, где ее значение зачастую используется. С помощью эксперимента данную величину измерить невозможно. Точность искомого коэффициента (качество идентификации) будет контролироваться величиной расхождения модельной температуры T и экспериментально измеренной T_e на конкретной установке в виде:

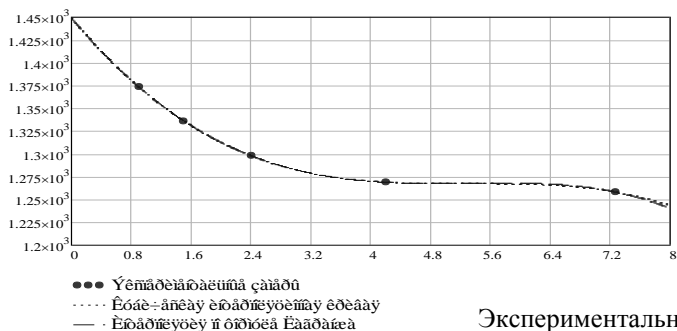
$$J(\mathbf{c}) = \int_0^z [T(r, z) - T_e(z)]^2 2\pi dz \rightarrow \min. \quad (4)$$

Для решения задачи используются значения температуры, экспериментально полученные на машине непрерывного литья заготовок (МНЛЗ) Днепровского металлургического комбината (табл.1). Замеры температуры производились на середине грани слитка с помощью оптического пирометра. На рассматриваемой нами части МНЛЗ имеется пять измерений.

Таблица 1 – Экспериментально полученные значения температуры

Величина	1	2	3	4	5
$Z, м$	0,9	1,5	2,4	4,2	7,28
$T, К$	1373	1336	1298	1269	1258

Воспользовавшись методами интерполяции (рис) получим значения температуры по всей длине рассматриваемого слитка с шагом 0,072 м.



Экспериментальные данные.

Рисунок 1 Интерполяция температуры по экспериментальным данным
 Оптимальное значение вектора \mathbf{c} рассчитывается градиентным методом:

$$\mathbf{c}^{k+1} = \mathbf{c}^k - b^k \nabla J(\mathbf{c}^k), \quad k = 0, 1, \dots$$

где $\nabla J(\mathbf{c}^k)$ – градиент целевой функции (3) на итерации k , b^k – шаг метода.

Расчёты будут проводиться со следующими значениями величин:
 $V = 1 \cdot 10^{-2} \text{ м/с}$, $R = 0,21 \text{ м}$, $Z = 7,28 \text{ м}$, $T_0 = 1799 \text{ К}$, $T_{sol} = 1709 \text{ К}$,
 $T_c = 1423 \text{ К}$, $z_c = 1 \text{ м}$, $T_h = 293 \text{ К}$.

Выводы. В постановке (1), (2) малое изменение любого компонента вектора \mathbf{c} приводило к малому изменению функции (3), т.е. чувствительность критерия качества $J(\mathbf{c})$ была приблизительно одинаковой ко всем компонентам \mathbf{c} . Это стало возможным благодаря проведенному масштабированию задачи, что, в свою очередь, позволило решать рассматриваемую задачу идентификации.

Список литературы.

1. Огурцов А.П., Недопекин Ф.В., Толстых В.К., Володин Н.А. Прямая оптимизация теплофизических процессов. - Донецк: Юго-Восток, 1997. – 150 с.
2. Толстых В.К. Идентификация теплофизических параметров в виде полиномов, зависящих от температуры / Недопекин Ф. В., Бодряга В. Е. // Технічна теплофізика та промислова теплоенергетика. – 2009. – Випуск №1. – С. 193-199.

УДК 62-83: 621.313

РАСЧЕТ ПЕРЕХОДНОГО ПРОЦЕССА ПРИ ВКЛЮЧЕНИИ ЭЛЕКТРОПРИВОДА В ОДНОФАЗНОЙ ЭЛЕКТРИЧЕСКОЙ ЦЕПИ ПЕРЕМЕННОГО ТОКА

Чепига А.А.

Донецкий национальный технический университет, г. Донецк
Факультет международных образовательных проектов, группа СПУ-15н
andreychepiga@yandex.ua

Аннотация

Чепига А.А. Расчет переходного процесса при включении электропривода в однофазной электрической цепи переменного тока. В статье рассмотрена проблема расчета переходных процессов при коммутации электропривода в однофазной электрической сети переменного тока промышленной частоты. Приводится схема замещения электрической цепи содержащей электропривод и выводится описывающая её система дифференциальных уравнений. Получено численное решение системы дифференциальных уравнений электрической цепи методом Рунге-Кутты 4-го порядка.

***Ключевые слова:** математическая модель, переходные процессы, численное решение дифференциальных уравнений, электропривод, численные методы.*

Постановка проблемы. Процессы коммутации силового оборудования сопровождаются переходными процессами. Результатом их являются искажения и скачки напряжений и токов в электрических сетях, которые приводят к ухудшению параметров электромагнитной совместимости и способны нанести вред соседним электроприемникам. При работе электропривода в составе устройства необходимо обеспечить электромагнитную совместимость с внутренними цепями, в том числе цепями управления, телеметрии, вторичного электропитания и сигнальными шинами. Расчет переходных процессов является одной из актуальных проблем управления электроприводом, т. к. это позволяет спрогнозировать их последствия и принять меры для защиты, как внутренних цепей устройства, так и для электрических сетей.

Методы расчета переходных процессов можно разделить на приближенные, основанные на эмпирических формулах и дающие примерное представление о коммутационных переходных процессах и точные, в основе которых лежит решение дифференциальных уравнений электрической цепи. Для расчета переходных процессов в цепи имеющей несколько реактивных

элементов необходима уже система дифференциальных уравнений, которую можно решить двумя способами: классическим и операторным. Результатом является точное аналитическое решение, описывающее закономерности изменения токов и напряжений на участках цепи.

По мере усложнения топологии электрической цепи и увеличении в ней количества реактивных элементов, аналитическое решение также усложняется. В ряде случаев система дифференциальных уравнений может изначально не иметь аналитического решения или же сложность её может быть такова, что получить решение фактически невозможно. В этом случае прибегают к численному моделированию переходных процессов электрической цепи.

Рассмотрим простейшую схему замещения цепи с электроприводом однофазного переменного (50 Гц) тока. Подобный электропривод широко распространен в бытовых приборах: стиральных машинах, холодильниках, пылесосах и т. д., а также имеет распространение на предприятиях малого и среднего бизнеса, вследствие его невысокой стоимости и простоты эксплуатации. Схема замещения представлена на рисунке 1, непосредственно электропривод моделируется активным сопротивлением R_2 и индуктивностью L_2 , подводящая линия электропитания активным сопротивлением R_1 и индуктивностью L_1 . Для коррекции коэффициента мощности вводится конденсатор C [1, с. 311].

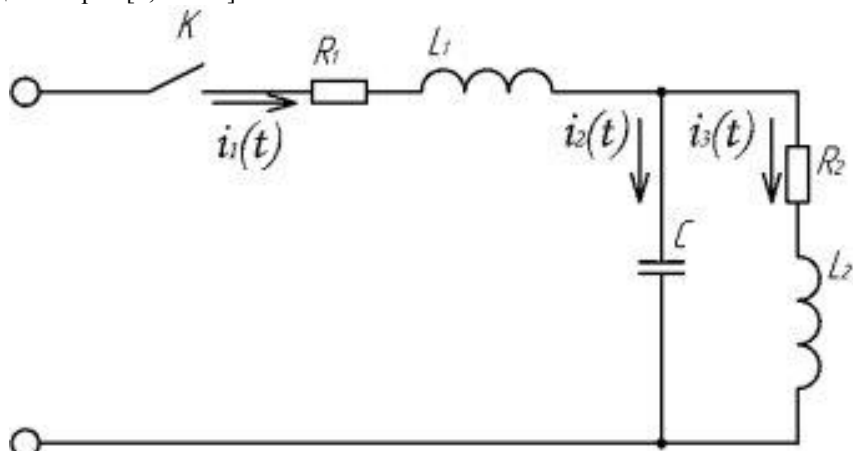


Рис. 1. Схема замещения цепи с электроприводом

Соответственно в цепи протекает три неизвестных тока i_1 , i_2 и i_3 . На основе законов Кирхгофа и учитывая законы коммутации (1 закон: ток в индуктивном элементе скачком измениться не может, т.е. ток до момента коммутации должен быть равен току в момент коммутации; 2 закон:

напряжение на емкостном элементе скачком измениться не может, т.е. до коммутации и в момент коммутации оно должно быть одинаковым) составим систему дифференциальных уравнений (СДУ), описывающую переходные процессы в схеме:

$$\begin{cases} R_1 \cdot i_1(t) + L_1 \cdot \frac{di_1(t)}{dt} + u_C(t) = u_{BX}(t) \\ R_2 \cdot i_3(t) + L_2 \cdot \frac{di_3(t)}{dt} - u_C(t) = 0 \\ i_1(t) - i_2(t) - i_3(t) = 0 \end{cases}$$

Учитывая, что ток второй ветви равен:

$$i_2(t) = C \cdot \frac{du_C(t)}{dt}$$

Преобразуем выражение (1) к виду:

$$\begin{cases} R_1 \cdot i_1(t) + L_1 \cdot \frac{di_1(t)}{dt} + u_C(t) = u_{BX}(t) \\ R_2 \cdot i_3(t) + L_2 \cdot \frac{di_3(t)}{dt} - u_C(t) = 0 \\ i_1(t) - C \cdot \frac{du_C(t)}{dt} - i_3(t) = 0 \end{cases}$$

Выразим производные искоемых токов и напряжений:

$$\begin{cases} \frac{di_1(t)}{dt} = \frac{1}{L_1} (u_{BX}(t) - R_1 \cdot i_1(t) - u_C(t)) \\ \frac{di_3(t)}{dt} = \frac{1}{L_2} (u_C(t) - R_2 \cdot i_3(t)) \\ \frac{du_C(t)}{dt} = \frac{1}{C} (i_1(t) - i_3(t)) \end{cases}$$

Преобразуем систему дифференциальных уравнений к виду :

$$\frac{dx(t)}{dt} = Ax(t) + b(t)$$

В результате получаем СДУ в матричной форме:

$$\begin{bmatrix} \frac{di_1(t)}{dt} \\ \frac{di_3(t)}{dt} \\ \frac{du_c(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_1}{L_1} & 0 & -\frac{1}{L_1} \\ 0 & -\frac{R_2}{L_2} & \frac{1}{L_2} \\ \frac{1}{C} & -\frac{1}{C} & 0 \end{bmatrix} \cdot \begin{bmatrix} i_1(t) \\ i_3(t) \\ u_c(t) \end{bmatrix} + \begin{bmatrix} \frac{u_{BX}(t)}{L_1} \\ 0 \\ 0 \end{bmatrix}$$

Рассмотрим переходный процесс при включении схемы (замыкании ключа К), в этом случае начальные условия:

$$\begin{bmatrix} i_1(t) \\ i_3(t) \\ u_c(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

. Таким образом, решение системы сводится к отысканию решения задачи Коши при нулевых начальных условиях. Решение можно производить любым известным численным методом, например: методом Эйлера, Рунге-Кутта, Кутта-Мерсона и т. д.

Для численного решения применим метод Рунге-Кутта 4-го порядка. Данный метод является одним из наиболее распространенных численных методов интегрирования обыкновенных дифференциальных уравнений. По сравнению с методом Эйлера метод Рунге-Кутта имеет более высокую точность, но невысокую скорость поиска решения, так как метод относится к классу многшаговых методов [2, с. 441]. Адаптируем его для решения СДУ:

$$K_1^i = Ax_i(t_i) + b(t_i)$$

$$K_2^i = A \left(x_i \left(t_i + \frac{h}{2} \right) + \frac{h}{2} K_1^i \right) + b \left(t_i + \frac{h}{2} \right)$$

$$K_3^i = A \left(x_i \left(t_i + \frac{h}{2} \right) + \frac{h}{2} K_2^i \right) + b \left(t_i + \frac{h}{2} \right)$$

$$K_3^i = A \left(x_i(t_i + h) + hK_2^i \right) + b(t_i + h)$$

$$x_{i+1}(t_{i+1}) = x_i(t_i) + \frac{h}{6} (K_1^i + K_2^i + K_3^i + K_4^i)$$

Результаты. Для примера, произведем расчет переходного процесса при следующих параметрах цепи $R_1=2$ Ом, $L_1=10$ мГн, $C=10$ мкФ, $R_2=100$ Ом, $L_2=100$ мГн [3, с. 250]. На рисунках 2, 3 приведены результаты расчета переходных токов в цепи с электроприводом, при включении моментов времени соответствующие фазам 0° и 90° синусоиды питающего напряжения.

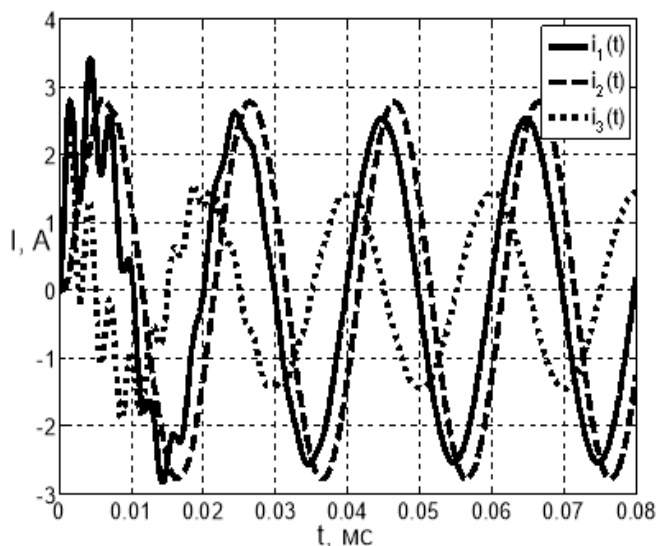


Рис. 2. Переходный процесс в цепи с электроприводом (фаза синусоиды $\varphi=0^\circ$)

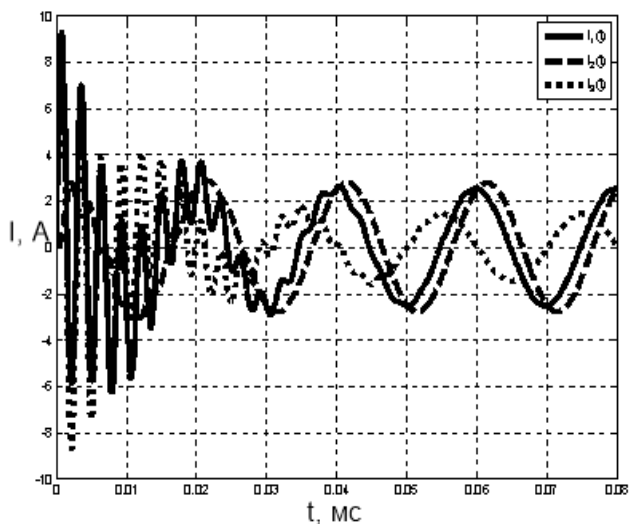


Рис. 3. Переходный процесс в цепи с электроприводом (фаза синусоиды $\varphi=90^\circ$)

Выводы. В обоих случаях имеет место скачок тока, причем при включении в момент прохождения синусоиды через ноль, скачок тока составляет 20 %, а при включении в момент максимума 210 %. Графики иллюстрируют временные диаграммы тока при переходном процессе, по которым возможно определить длительность процесса, его гармонический состав и амплитуду создаваемых им скачков тока. Метод расчета позволяет проводить численное моделирование переходных процессов при различных параметрах схемы замещения и реализуем с помощью стандартных математических пакетов (например, Mathcad, Matlab) на персональном компьютере.

Список литературы

1. Ковчин С. А., Сабинин Ю. А. Основы электропривода. — С - П.: Энергоатомиздат., 1994. — 496 с.
2. Э. Камке. Справочник по обыкновенным дифференциальным уравнениям. — М.: Наука, 1971. — 576с.
3. Башарин А.В., Голубев Ф.Н., Кепперман В.Г. Примеры расчета автоматизированного электропривода. — Л.: Энергия, 1972. — 440 с.

УДК 681.3

ОПИСАНИЕ И ВЫБОР МЕТОДА ДЛЯ ПРОЕКТИРОВАНИЯ СКС ДЛЯ ПОСТАНОВКИ ПРЕДВАРИТЕЛЬНОГО ДИАГНОЗА ПО БИОХИМИЧЕСКОМУ АНАЛИЗУ СЛЮНЫ

Д.В. Орловский

Донецкий национальный технический университет, кафедра АСУ,
aaronrooss@gmail.com

Аннотация:

Орловский Д.В. Описание и выбор метода для проектирования скс для постановки предварительного диагноза по биохимическому анализу слюны. В данной статье рассмотрены основные подходы к проектированию СКС для постановки предварительного диагноза по биохимическому анализу слюны. Приводится подробный анализ методов, на основе которого делается выбор в пользу наиболее оптимальной методики для выполнения поставленной задачи.

Ключевые слова: биохимический анализ, анализ слюны, проектирование скс, методы и подходы.

Несмотря на достаточно широкое уже распространение слюны как аналитического материала, существующие решения преимущественно решают задачу выявления её химического состава, тогда как все этапы непосредственной диагностики, составляющей достаточно большую часть временных затрат, остаётся на долю человеческого диагноста, что особенно критично в медицине.

Вместе с тем, в мировой практике диагностики широко применяются разнообразные системы автоматизированной поддержки принятия решений, позволяющие в значительной мере сократить эти временные затраты для широкого спектра достаточно часто встречающихся заболеваний.

По преимуществу, в системах автоматизированной поддержки принятия решений в медицине используются следующие подходы:

- составление баз экспертных знаний, представляющих собой наборы дедуктивных правил, составляемых экспертами в виде выражений символической логики;

- экстраполирующие методы машинного обучения, заключающиеся в представлении поставленной задачи в виде некоторой заранее неизвестной функции и последующему моделированию и оптимизации набора функций и алгоритмов, оптимизирующих эту неизвестную функцию с достаточной точностью;

– классифицирующие методы машинного обучения, ставящие более узкую задачу отнесения каждого обособленно предоставленной сущности данных к одному или более классам.[1]

Рассмотрим подробнее преимущества и недостатки каждого из них.

Формирование баз экспертных знаний — наиболее ранний и хорошо изученный из всех подходов при проектировании систем автоматизированной поддержки принятия решений. Он наиболее близок к человеческому представлению о процессе трансформации данных в знания, и более всех прочих удобен для восприятия.

Суть подхода заключается в формировании ряда предикатов символической логики (и, иногда, инкапсулированных в них выражений на вспомогательных языках, таких как математические формулы или алгоритмы), каждое из которых описывает некоторое явление, организованных в ряд связанных между собой иерархических структур согласно установленным экспертом причинно-следственным связям между ними.

В ряде случаев некоторым (или даже всем) предикатам также атрибутивно присваиваются вспомогательные выражения на человеческом языке для облегчения восприятия.

Преимущества такого подхода очевидны: высокая точность, однозначная обоснованность вынесенных системой предположений, прозрачность процесса работы системы для человека.

К недостаткам прежде всего следует отнести большие временные затраты на формирование и запись правил экспертом, при этом хранение достаточно подробных баз знаний может требовать значительных вложений на приобретение накопителей данных. Помимо этого, получаемые результаты всегда строго детерминированы набором заданных правил, т.е. моделирование стохастических процессов при таком подходе значительно затруднено.[2]

Используя экстраполяцию, можно как строить разнообразные итеративные модели машинного обучения, например, сети искусственных нейронов, тензорные представления и т.п., представляющие собой аппроксимирующие модели заранее неизвестных функций, так и решать менее обобщённые задачи распознавания образов статистических данных, позволяющие не столько аппроксимировать неизвестную функцию, сколько выяснить её постоянные члены, предполагая определённый её вид — например, при помощи регрессии.

К преимуществам экстраполирующего подхода относятся достаточно высокая точность, а в большинстве случаев — и простота оценки погрешности построенных моделей. Помимо этого, имеется огромный корпус типовых практических решений по разработке и внедрению таких моделей, что позволяет минимизировать участие человека в их оптимизации.

Что до недостатков, наибольшими для данного подхода являются повышенная требовательность к вычислительным ресурсам и практически полная непрозрачность процесса работы выходных моделей, кроме того, как правило, получение достаточно точных моделей предметной области с помощью этого подхода, как правило, требует большого объема обучающей выборки.

Базовая модель классифицирующего подхода предельно проста и заключается в том, чтобы подав на вход разработанной модели обособленную сущность данных, получить на выходе некий класс или набор классов, к которым эту сущность можно отнести.

Основные преимущества подхода — объем класса задач, которые можно с его помощью решать, и количество применимых методов.

Основной его недостаток — подход совершенно неприменим к задачам, которые невозможно или не следует сводить к задаче классификации.

Прочие же недостатки и преимущества решения задач с применением данного подхода присущи не ему самому, но, скорее, используемым с ним методам.

Критерии выбора подхода определяются, прежде всего, предметной областью. Прежде всего, поскольку предметная область относится к медицине, следует отметить, что для используемого подхода важна высокая прозрачность получения и функционирования доменной модели, поскольку это область повышенной ответственности.

Также следует отметить, что аналитическая диагностика образцов секрета слюнных желез человека в настоящее время является сравнительно новой и, как следствие, не настолько глубоко разработанной методикой по сравнению со многими известными ранее. Помимо этого, и количество специалистов по такой аналитической диагностике также сравнительно невелико. Таким образом, к критериям выбора подхода следует отнести низкую требовательность к объему корпуса доступных знаний о предметной области, подразумевая также и отсутствие требования к временным затратам медика, являющегося специалистом в рассматриваемой предметной области.

Также, проведенная исследовательская работа показала крайнюю труднодоступность статистических данных по составу секрета слюнных желез, не говоря о связанных диагнозах, что принуждает также выбрать подход, позволяющий вынесение успешных предположений даже при небольшом объеме изначально доступной статистики.

Таким образом, критерии прозрачности и нетребовательности к объему статистических данных вынуждают отказаться от использования экстраполирующего метода, а критерий нетребовательности к объему корпуса знаний — от составления базы знаний.

К счастью, задача аналитической диагностики образцов биологических материалов хорошо сводится к задаче классификации:

- в качестве обособленных сущностей данных можно представить каждый набор показателей химического состава образца каждого пациента;
- в качестве классов можно представить сами диагнозы.

Далее следует рассмотреть методы решения классификационных задач и выбрать наиболее подходящий.[3]

Наиболее распространёнными методами решения задачи определения отношения обособленной сущности данных к одному или более классам являются:

- классификация на основе базы знаний;
- классификация при помощи генетического алгоритма;
- классификация построением искусственных нейронных сетей;
- вероятностная классификация;
- классификация при помощи кластерного анализа.

Генетический, или эволюционный алгоритм представляет собой эвристический алгоритм поиска, используемый для решения задач моделирования с помощью случайного подбора, комбинирования и изменения параметров, сходных с генетическими процессами в биологии, откуда и происходит его название.

Вкратце, суть метода такова: данные поставленной задачи формируются таким образом, чтобы их можно было представить в виде вектора булевых величин заранее известной длины.

На каждой итерации моделирования определённым образом формируется популяция таких векторов: на первой, иницилирующей итерации они формируются случайным образом, на второй и последующих — отбором некоторой наиболее оптимальной части популяции и генерации остальных членов популяции на их основе с использованием эволюционных приёмов («кроссинговера», т.е. образованием вектора путём совмещения двух других по заранее заданной закономерности и «мутации», т.е. псевдослучайного изменения одного вектора).

Отбор осуществляется путём подсчитывания для каждого вектора («особи») заданной оптимизационной функции, называемой «фитнесс-функцией».

Итерирование заканчивается, когда значение фитнесс-функции одной или более особей достигает некоторого определённого заранее порогового значения, или после заданного числа итераций.

Основным достоинством, а равно и недостатком этого метода является его стохастичность и, как следствие, непредсказуемость: сравнимы вероятности того, что решение с высокой оптимальностью может быть достигнуто за малое число итераций, и того, что за большое число итераций не будет достигнуто сколько-либо оптимального решения.

Все методы классификации вероятностным путём так или иначе опираются на теорему Байеса о связи вероятностей взаимозависимых событий А и В:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

где $P(A)$ — полная вероятность наступления события А, или априорная вероятность истинности гипотезы А, $P(B)$ — полная вероятность наступления события В, $P(A|B)$ — вероятность наступления события (истинности гипотезы) А при наступлении события В, и $P(B|A)$ — вероятность наступления события В при наступлении события (истинности гипотезы) А. Здесь событие А также называется гипотезой потому, что предполагается, что оно влечёт за собой наступление события В.

При использовании основанных на теореме Байеса методов для классификации наборов данных, классы становятся гипотезами, а данные – гипотетически зависимыми от них событиями.

Основное достоинство решений на основе вероятностной классификации — в их обучаемости. Даже самые примитивные вероятностные классификаторы способны эффективно обучаться на сравнительно небольших наборах данных, а также продолжать дальнейшее самообучение без изменений базовой конфигурации при минимальном вмешательстве человека.[4]

Недостаток же вероятностных классификаторов заложен в их основополагающем принципе: от них можно получить лишь результат, верный с некоторой вероятностью, которая будет тем ниже, чем меньше будет объём обучающей выборки.

Обобщённо поставленная задача кластерного анализа заключается в упорядочивании объектов, характеризуемых наборами числовых или сводимых к числовым свойств, в однородные группы.

В общих чертах, моделирование с помощью кластерного анализа состоит из следующих шагов:

- приведение выборки объектов к наборам числовых показателей;
- определение меры сходства или различия между всеми объектами выборки на основе показателей;
- определение групп исходя из заранее заданных предположений.

К достоинствам кластерного анализа можно отнести наглядность, т.к. каждый объект и мера сходства между любыми двумя объектами описывается и вычисляется однозначно, и, для некоторых его методов, способность к выявлению новизны, то есть таких объектов, которые могут не относиться ни к одной из ожидаемых групп.

К недостаткам следует отнести однозначность результатов при том, что мера сходства (или различия) не имеет собственного смысла и не позволяет прозрачно оценить корректность результатов классификации.

Выбирая метод для решения поставленной задачи предметной области, следует руководствоваться также и практическими соображениями.

Прежде всего, скорость работы решения на основе выбранного метода должна быть достаточно высокой, поскольку одна из подзадач — сокращение временных затрат на постановку диагноза медиком.

Кроме того, ввиду распространения автоматических анализаторов биологического материала, в том числе и секрета слюнных желез, представляется целесообразным выбрать такой метод, который не налагал бы строгих ограничений на требуемую для работы основанного на нём решения вычислительную мощность, и позволял бы применять данное решение в том числе и на маломощных встраиваемых устройствах, а также на физических «тонких клиентах» (маломощных рабочих станциях, ориентированных на потребление информации, производимой сторонними серверами).

Классификатор, полученный на основе генетического алгоритма обеспечивает достаточное быстродействие и не требователен к аппаратным ресурсам при выполнении своих непосредственных функций, однако, его дальнейшее самостоятельное обучение не отвечает этим требованиям, и часто требует вмешательства специалиста в информационных технологиях. Кроме того, прозрачность его работы оставляет желать лучшего.

Вероятностный классификатор на основе теоремы Байеса работает быстро как в диагностическом режиме, так и в режиме обучения, и при этом наиболее прозрачен из всех рассмотренных методов, но при этом точность его результатов никогда не достигает 100%. Следует также отметить, что у вероятностного классификатора есть отличительная особенность — он способен содействовать методике дифференциальной диагностики, показывая несколько наиболее вероятных диагнозов.

Наконец, кластерный классификатор точен, но медлителен при постановке диагноза, т.к. должен высчитывать меру сходства каждого результата анализа как минимум со всеми объектами одной из групп, а возможно и со всеми объектами обучающей выборки. Кроме того, он ещё более медлителен при дополнительном самообучении. Прозрачность его работы выше, чем у полученного с помощью генетического алгоритма классификатора, но ниже, чем у вероятностного ввиду меньшей информативности оценок.

Таким образом, наиболее подходящим для реализации поставленной задачи представляется вероятностный классификатор, основанный на теореме Байеса.[5]

Предположим, что причинно-следственные связи есть между каждым диагнозом и каждым формализованным показателем анализа секрета слюнных желез. Предположим также, что непосредственной причинно-следственной связи нет между диагнозами, и что её также нет между показателями анализа.

Тогда, в условиях ограниченной доступности обучающей статистики, представляется целесообразным использовать набор «наивных классификаторов Байеса», объединённых в классификационную сеть.

Определим множество диагнозов как множество гипотез $D = \{D_i\}$, где D_i — гипотеза об истинности диагноза i . Отметим, что данное множество обязательно включает в себя диагноз D_0 («здоров») во избежание избыточной диагностики.

Каждый показатель анализа, в свою очередь, может принимать достаточно широкий спектр значений, что затрудняет использование Байесова классификатора. К счастью, использование понятия медицинской нормы позволяет любое значение, с учётом пола и возраста пациента, отнести к одному из трёх состояний: «значение ниже нормы», «значение нормально» и «значение выше нормы».

Исходя из этого, определим множество возможных значений показателей анализа для каждого пациента как $A = \{A_j^s\}, j \in [1; M] \subset N, s \in \{-; 0; +\}$, где A_j^s обозначает событие, соответствующее нахождению j -го показателя в одном из состояний s .

Также перед началом обучения ввиду отсутствия обоснованных пресуппозиций, согласно агностическому принципу установим, что для каждого $D_i \forall j, k, s, t: P(A_j^s | D_i) = P(A_k^t | D_i)$.

Таким образом, обучаемая Байесова сеть будет иметь вид:

Для обучения с учётом пола и возраста каждого пациента на момент забора биоматериала будем приводить показатели к одному из трёх вышеупомянутых состояний.

Выводы: Таким образом, были рассмотрены наиболее распространённые методы поддержки принятия решений в условиях неопределённости, и ограничения налагаемые на процесс их отбора как условиями поставленной задачи, так и условиями использования её решения.

В рамках этих ограничений отобран наиболее оптимальный согласно критериям отбора подход и метод решения задачи и построена математическая модель для последующей алгоритмизации.

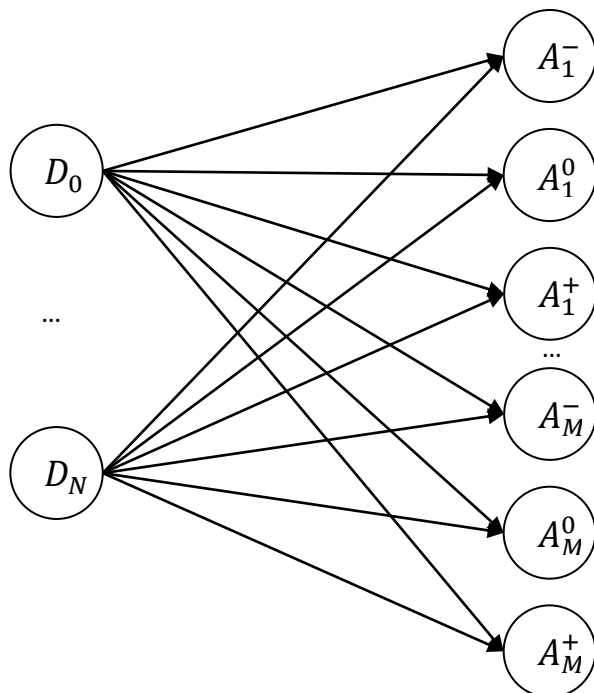


Рис. 1 – общий вид классифицирующей сети Байеса

Список литературы:

1. Григорьев И.В., Уланова Е.А., Артамонов И.Д. Белковый состав смешанной слюны человека: механизмы психофизиологической регуляции // Вестник РАМН. 2004. № 7. — С. 36–47.
2. Григорьев И.В. Роль биохимического исследования слюны в диагностике заболеваний / И.В.Григорьев, А.А.Чиркин // Клини. лаб. Диагностика - 1998. - №6. - С.18-20.
3. Принципы и алгоритмы искусственного интеллекта/ Э.А. Бабкин, О.Р. Козырев, И.В. Куркина. – Н. Новгород: Нижегород. гос. техн. ун-т., 2006. – 132 с.
4. Оптимизационная модель выбора начального плана управляющих воздействий для моделирования информационных систем/ О.Н. Чопоров, К.А. Разинкин// Системы управления и информационные технологии, 2011. – Т. 46. № 4.1. – С. 185-187.
5. Кобринский Б.А. Логика аргументации в принятии решений в медицине// НТИ, сер.2. – 2001. – №9. – С.1-8.

Секция № 6
Системы искусственного
интеллекта



УДК 004.89

МОДУЛЬ ВИЗУАЛИЗАЦИИ ДВИЖЕНИЯ АВТОТРАНСПОРТА ПО КУРСУ «ОСНОВЫ ПРАВИЛ ДОРОЖНОГО ДВИЖЕНИЯ»

А.И. Ольшевский, В.В. Марченко

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
a_olshevskiy@mail.ru

Аннотация

Ольшевский А.И., Марченко В.В. Модуль визуализации движения автотранспорта по курсу «Основы правил дорожного движения». Рассмотрены общие положения правил дорожного движения. Выполнен анализ последовательности изучения теоретического материала и определена методика обучения с элементами мультимедийных технологий. Определен подход формирования визуализации последовательности проезда различных типов перекрестков, схожих с реальными дорожными ситуациями.

***Ключевые слова:** форматы графических изображений, визуализация движения, автотранспорт, правила дорожного движения, перекресток.*

Общая постановка проблемы. Для процесса обучения первостепенную актуальность приобретает задача использования возможностей информационных технологий в моделировании исследовательской и профессиональной деятельности. Современные образовательные процессы не могут проходить без включения в процесс обучения широкого спектра информационных ресурсов, без развития умений обработки и представления информации [1].

В настоящее время знание правил дорожного движения (ПДД), а именно теоретической части при обучении вождению является высоко актуальной, которой ни в коей мере нельзя пренебрегать.

Многие начинающие водители большую часть своих занятий посвящают совершенствованию практических навыков вождения, что, в принципе, является грубой ошибкой. Наверняка многие из тех, кто когда-либо учился водить автомобиль, сталкивались с подобными людьми и наглядно наблюдали за тем, насколько сложно они адаптируются и переучиваются заново в отношении некоторых нюансов вождения. В связи с этим, опытные водители с многолетним стажем и квалифицированные инструкторы советуют начинать именно с изучения свода ПДД.

Элементарное знание правил дорожного движения также важно знать потому, что область их использования необычайно широка. Ведь в ПДД рассматриваются такие важные составляющие вождения как остановка,

стоянка, парковка, расположение транспортного средства на проезжей части, маневрирование, проезд перекрестков и многое другое.

Постановка задачи. Объектом разработки является информационно–обучающая система ПДД, структуры, модели и методы визуализации движения автотранспорта при вождении схожими с реальными дорожными ситуациями.

Цель проекта – повышения эффективности обучения правилам дорожного движения, а так же формирование у пользователей системы правильного представления о приемах вождения автомобиля, разбор наиболее популярных ситуаций на дорогах города, формирование у обучаемых аналитико-мыслительной деятельности, за счет использования мультимедийных технологий включая в себя игровой аспект.

Разрабатываемый программный продукт будет способствовать развитию водительского мышления и подготовке к сдаче практического экзамена в ГАИ (изучение дорог, дорожных знаков, сложных перекрестков и т.д.).

Применение мультимедийных технологий обеспечивает лучшее восприятие учебного материала учащимися. Иллюстрации являются зрительной опорой, которая помогает наиболее полно усвоить материал. Программа включает в себя стимулирующий игровой аспект, что повышает интерес к изучению материала у всех возрастных групп.

Исследования. Программный продукт будет полезен каждому, кто изучает ПДД, учится в автошколе или просто хочет увеличить свои шансы сдать экзамены в ГАИ. Программа несет практическую пользу так же водителям, желающим повторить практическую часть ПДД – вспомнить и закрепить в памяти те требования Правил, которые чаще всего применяются на практике.

Теоретическое обучение мастерству вождения включает в себя раскрытие следующих вопросов:

- первые навыки водителя;
- простейшие маневры;
- безопасное движение в типичных дорожно-транспортных ситуациях;
- восприятие и прогнозирование дорожно-транспортной обстановки;
- типичные опасные ситуации и ошибки водителей;
- вождение в сложных условиях;
- формирование водительского мастерства.

Данная методика обучения достаточно полно раскрывает содержание учебного материала при иллюстрации примеров преподавателем. Позволяет воспроизвести различные ситуации, недоступные непосредственному наблюдению. Визуализация дает возможность увидеть то, что в действительности происходит в разное время и в различных дорожных ситуациях.

Кроме того программа включает в себя игровой аспект, что оказывает стимулирующее воздействие на весь спектр учебных способностей пользователя, позволяет использовать полученные ранее знания, повышает интерес к изучению материала разных возрастных категорий обучаемых.

Программная реализация. При программировании следует отделять интерфейс (функции, модуля, класса) от его реализации и ограничивать доступ к ненужной информации. Небрежное даже в мелочах программирование может привести к огромным затратам на поиск ошибок на этапе отладки. Этапы проектирования и программирования совмещены во времени: в идеале сначала проектируется и кодируется верхний уровень, затем – следующий, и так далее. Такая стратегия применяется потому, что в процессе кодирования может возникнуть необходимость внести изменения, отражающиеся на модулях нижнего уровня [2].

Выполненный обзор существующих интеллектуальных обучающих систем дает следующие виды технологий в интеллектуальных обучающих системах:

- построение последовательности курса обучения;
- интеллектуальный анализ ответов обучаемого;
- интерактивная поддержка в решении задач;
- помощь в решении задач, основанная на примерах.

Программа предоставляет пользователям теоретическую информацию об основных положениях правил дорожного движения, с помощью средств визуализации, а именно видео фрагментов (анимация), иллюстрированного текста.

Разрабатываемый программный продукт представлен в виде программной визуализации ситуаций состоящей из нескольких модулей:

- модуль инициализации уровней, реализован в виде дорожных ситуаций (поля игры) с дорогой и готовыми для проезда автомобилями;
- модуль визуализации участников дорожного движения (автомобили, мотоциклы, трамваи, инспекторы ГАИ);
- модуль отображения заставки, отвечает за загрузку основной формы;
- модуль проезда перекрестка, представляет собой движение авто по координатам заданного пути, при условии правильного выбора очередности;
- модуль проверки очередности, в котором реализована проверка правильного выбора движения автомобилей (база знаний);
- модуль отображения справочной информации, который можно вызывать нажав предусмотренные «горячие» клавиши;
- модуль сохранения результатов проделанной работы, для интеллектуального анализ ответов обучаемого (накопление баллов по ходу прохождения записываются в отдельный файл).

Программная реализация создана на языке С# и состоит из главного меню и последовательно вызываемых форм [3].

В формах, в свою очередь, размещены элементы интерфейса, взаимодействие с которыми вызывает другие элементы или формы. На рисунке 1 показана общая структура вызовов элементов программы и их взаимодействия.

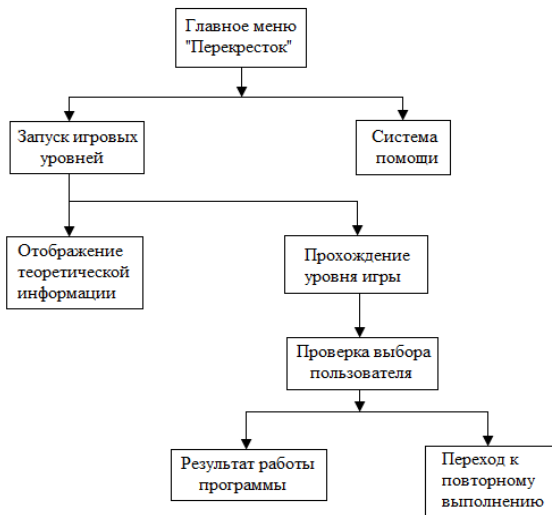


Рисунок 1 – Общая структура вызовов элементов программы и их взаимодействия

В программе будут представлены различные типы перекрестков, которые наиболее часто встречаются в реальной жизни, на каждом перекрестке будет разное количество транспортных средств с необходимостью двигаться в определенных направлениях. Пользователю отгадываясь от присутствующих знаков дорожного движения нужно будет выбрать правильную последовательность движения автотранспорта, не нарушая ПДД.

На рисунке 2 показан пример одного из перекрестков разрабатываемого программного продукта. На перекрестке имеется 2 различных знака, а именно: знак «Главная дорога» с указанием под ним ее направления, и знак «Уступи дорогу» который всегда используется в сочетании с предыдущим. Так же на перекрестке присутствует 3 легковых автомобиля, с необходимостью движения в определенном направлении.

Исходя из правил дорожного движения первым должен двигаться желтый автомобиль, т.к. он находится на главной дороге и у него нет препятствий для движения, затем красный, который не мог двигаться первым по «Правилу

левой руки» - при повороте на лево необходимо уступить дорогу встречному автомобилю движущемуся прямо и направо, после чего повернет синий автомобиль. Задача пользователя состоит в том, что б выбрать автомобили в правильном порядке. Ниже показан рисунок 2 с примером перекрестка.

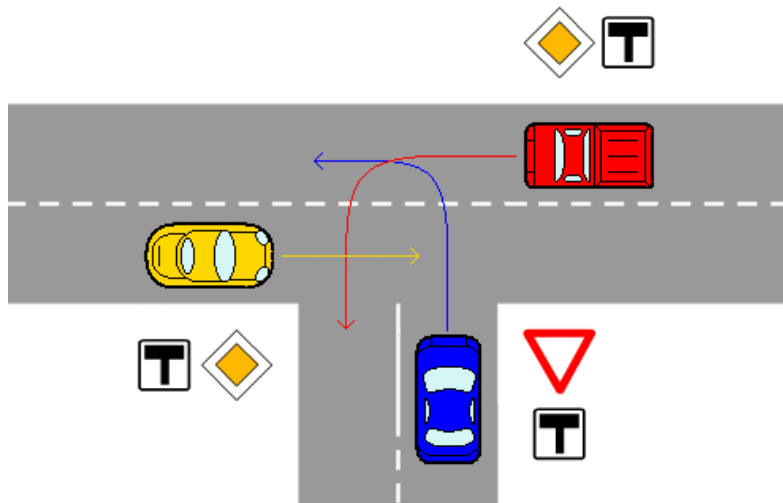


Рисунок 2 – Пример перекрестка в программе

Выводы. Выполнен анализ последовательности изучения теоретического материала правилам дорожного движения и определена методика обучения с элементами мультимедийных технологий. Определен подход формирования визуализации последовательности проезда различных типов перекрестков, схожих с реальными дорожными ситуациями. Был разобран программный модуль и проведен анализ эффективности использования разрабатываемого модуля, на основе контроля результатов накопления баллов по ходу прохождения обучения пользователем.

Список литературы

1. Юрков Н. К. Интеллектуальные компьютерные обучающие системы / Юрков Н. К. [Электронный ресурс]. – Режим доступа: http://www.mtas.ru/upload/library/MONOGRFIYa_IKOS_2010.pdf
2. Павловская Т. А. С/С#. Программирование на языке высокого уровня / Павловская Т. А. – СПб.: Питер, 2010.
3. Давыдов В. Г. Технологии программирования. С#: [руководство]/ В. Г. Давыдов. –СПб.: БХВ-Петербург, 2005.

УДК 004.023

МОДИФИЦИРОВАННЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ СОСТАВЛЕНИЯ РАСПИСАНИЯ В УСЛОВИЯХ УЧЕБНОГО ЗАВЕДЕНИЯ

М.А. Безуглый, А.И. Секирин

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

maksimdonntu@lenta.ru, alx09@list.ru

Аннотация

Безуглый М.А., Секирин А.И. Модифицированный генетический алгоритм составления расписания в условиях учебного заведения. В статье рассматривается генетический алгоритм и его операторы, адаптированные для составления расписания в высшем учебном заведении.

Ключевые слова: расписание занятий, генетический алгоритм, представление хромосомы, генетические операторы.

Постановка проблемы. Важной задачей организации учебного процесса является своевременное составление качественного расписания, благодаря которому учебный процесс должен быть построен таким образом, чтобы обеспечивать равномерную нагрузку для студенческих групп и преподавательского состава.

От эффективности составленного расписания зависит: качество обучения, комфортность учебы студентов и работы профессорско-преподавательского состава, возможность проводить учебный процесс без сбоев, а также избегать необходимости корректировки расписания после начала учебного семестра [1].

Качество расписания напрямую зависит от учета требований методического и организационного характера, преподавателей и руководства высшего учебного заведения. Трудоемкость решения задачи напрямую зависит от количества таких требований. Для поиска оптимального решения необходимо произвести полный перебор всех возможных вариантов, что не всегда возможно сделать ввиду ограниченности ресурсов. Построение распределения занятий таким при увеличении размерности задачи приводит к логарифмическому росту затрачиваемых ресурсов для нахождения решения. В связи с этим на практике применяют эвристические методы, характеризующихся сочетанием полиномиальной зависимости времени счета от размерности задачи и точностью, близкой к оптимальной [2,3].

Таким образом выявлены следующие основные недостатки ранее применяемых алгоритмов для составления расписания занятий: трудоемкость, экспоненциальный рост временных затрат при увеличении размерности задачи, сложность учета предпочтений. Следовательно, необходимы

дальнейшие исследования и разработка алгоритмов, позволяющих составлять эффективные расписания [4].

Математическая постановка задачи. Расписание можно представить в виде функции $S(P, G, A, T, Z, V)$, зависящей от 6 переменных.

Тогда таблицу расписания можно представить в виде декартова произведения

$$S = G \times P \times A \times T \rightarrow \{0, 1\} \quad (1)$$

где G – множество обучающихся групп;

A – множество аудиторий;

P – множество преподавателей;

T – множество учебных пар (временных интервалов проведения занятий).

Данное выражение расшифровывается следующим образом: если в определенное время T преподаватель P у группы G проводит занятие в аудитории A , то значение выражения равно 1; иначе 0.

Также при решении задачи используются следующие данные:

– множество занятий Z ;

– рабочие учебные планы V (трехмерная матрица, определяющая для каждого занятия преподавателей, которые его проводят, и группы, которые на нем присутствуют).

Введем основные ограничения, предъявляемые к расписанию.

Отсутствие накладок в расписании преподавателя:

$$\forall (p_i, t_j) : p \in P, t \in T \left(\sum_{k=1}^{Z_{t_j}} z_{t_j}^k : (p_i = p_k) \leq 1 \right) \quad (3)$$

где Z^{t_j} – множество блоков занятий, проводимых во время пары t_k

$z_{t_j}^k$ – занятие из множества Z_{t_j} проводимое во время пары t_j

p_i – преподаватель, по которому проверяется наличие накладок

p_k – преподаватель, который ведет пару $z_{t_j}^k$

Это ограничение гарантирует отсутствие накладок для преподавателей: существует либо единственный блок занятий, которые ведет данный преподаватель во время заданной пары, либо этого блока не существует вообще.

Отсутствие накладок у групп:

$$\forall (g_i, t_j) : g \in G, t \in T \left(\sum_{k=1}^{Z_{t_j}} z_{t_j}^k : (g_i = p_k) \leq 1 \right) \quad (4)$$

где Z^{t_j} – множество блоков занятий, проводимых во время пары t_k

$z_{t_j}^k$ – занятие из множества Z_{t_j} проводимое во время пары t_j

g_i – группа, по которой проверяется наличие накладок

g_k – группа, присутствующая на паре $z_{t_j}^k$

Это ограничение обеспечивает отсутствие накладок для учебных групп.

Во время конкретной пары группа находится на одном занятии, или проводится занятие только у одной из подгрупп, либо у обеих, либо занятий нет вообще.

Решение задачи. Основными достоинствами генетического алгоритма и его модификаций являются:

- алгоритм работает с кодами, в которых представлен набор параметров, напрямую зависящих от аргументов целевой функции;
- в процессе поиска алгоритм использует несколько точек поискового пространства, а не переходит от точки к точке;
- в процессе работы алгоритм может не использовать никакой дополнительной информации о задаче, но, если таковая имеется, можно ускорить сходимость алгоритма;
- генетический алгоритм использует как вероятностные правила для порождения новых точек поиска, так и детерминированные для перехода от одних точек к другим;
- для задач высокой размерности скорость работы алгоритма можно «регулировать» размерами популяции.

Целевая функция. Для оценки расписаний, удовлетворяющих требованиям, предлагается использовать в качестве целевой функции P – сумму штрафов за нарушение предпочтений, предъявляемых к расписанию.

$$P = \sum_{i=1}^N \sum_{j=1}^{NW} Z_i^{rj} * W_j \quad (5)$$

где NW – количество требований;

W – список штрафных коэффициентов длиной NW , соответствующий требованиям;

N – количество занятий в расписании (количество генов в хромосоме);

Z – список занятий длиной N ;

z_i^r – список из булевых длиной NW значений, где каждому значению соответствует требование и штрафной коэффициент из списка W .

Расписания, которые не удовлетворяют ограничениям будут считаться не жизнеспособной особью, в результате чего нет необходимости оценивать данное решения.

Кодирование хромосомы. Предлагается провести кодирования расписания основанной на схеме представленной в работе [Ошибка! Источник ссылки не найден.], при этом предлагается изменить смысл гена аудиторий. Вместо номера аудитории данный ген будет кодировать порядковый номер аудитории из списка подходящих помещений для проведения занятия.

Занятие – это сущность, которая определяет проведение одной пары (рис.1), и содержит всю необходимую для этого информацию о преподавателях, которые проводят занятие и группах, присутствующих на занятии, список номеров аудиторий, в которых можно проводить данное занятие и о проводимом занятии.



Рис.1. Схематическое представление сущности занятия

Составляется список занятий Z , при этом в данный список помещается столько экземпляров занятия, сколько должно быть проведено пар в течение двух недель.

Тогда особь будет состоять из двух хромосом (рис.2):

1) время проведения занятия;

2) номер аудитории из допустимого множества аудиторий для занятия.

Позиция каждого гена в хромосоме однозначно соответствует занятию из списка Z .

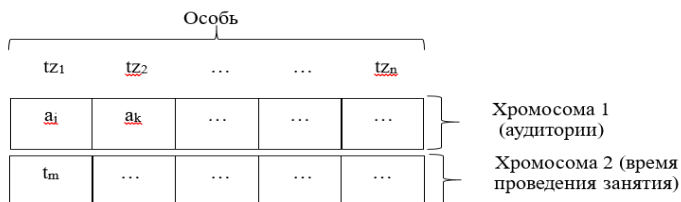


Рис.2. Структура хромосомы

Каждое значение во второй хромосоме представляет собой время проведения занятия (действительное число обозначает номер пары). В рассматриваемом примере составляется расписание, состоящее из двух недель, ежедневно может проводиться 5 пар, рабочими считаются дни с понедельника по пятницу, следовательно, количество возможных значений гена равно 50.

Операторы генетического алгоритма. Для реализации алгоритма были адаптированы следующие операторы:

1. оператор селекции – пропорциональный отбор (метод «ранжирования»);
2. оператор рекомбинации – дискретная рекомбинация;
3. оператор редукции – пропорциональный отбор;

4. оператор мутации – изменяет значение нескольких генов на случайное, выбранное из диапазона возможных значений (1-50).

Оператор мутации восстановления. Для данного представления расписания в виде хромосомы не все комбинации генов являются допустимыми, в результате чего особь становится нежизнеспособной. Для реализации генетического алгоритма предлагается ввести оператор мутации восстановления.

Для этого за каждым лектором и группой закрепляется их собственный вектор расписания, который состоит из натуральных чисел. Длина вектора равна максимальному номеру пары в расписании, в данном случае 50.

Номер ячейки в векторе соответствует номеру пары в расписании; если данная ячейка пуста (равна 0), это означает, что в данное время у преподавателя (или группы), за которой закреплен данный вектор, нет занятия.

Оператор мутации восстановления состоит в том, чтобы, последовательно просматривая особь, заполнять векторы расписания для каждого из участников учебного процесса. Перед проверкой очередной особи вектора расписания очищаются. После чего оператор берет первый ген, который соответствует первому занятию в списке занятий Z . Проверяет, являются ли пустыми ячейки, соответствующие времени проведения данного занятия у преподавателей и групп. Если данные ячейки являются пустыми, тогда оператор заносит в них номер проверяемого гена (который соответствует номеру занятия в списке Z) и переходит к следующему гену. Как только оператор встречает не пустую ячейку, он производит сложение векторов. Если после этой операции получен вектор, содержащий пустые (нулевые) ячейки, то занятие переносится на новое время, соответствующее полученной нулевой ячейке. Если в процессе сложения векторов не остается пустых ячеек, оператор завершает работу и сигнализирует о нежизнеспособности особи.

Реализация алгоритма. На основе вышеизложенных материалов была составлена тестовая программа. Для генерации начальной популяции был использован метод теории графов, при этом каждое занятие было представлено в виде вершины графа. Если занятия не могут проводиться в одно и тоже время, они соединяются ребром, после чего задача сводится к раскраске графа в заданное количество цветов [5] (в данном случае 50) Тестовая программа не учитывает задачу распределения аудиторного фонда в результате чего особь состоит только из одной хромосомы.

В качестве предпочтений были выбраны:

- количество окон у преподавателей;
- количество окон у студентов;
- количество повторяющихся пар в день у студентов.

Алгоритм был выполнен в двух разных режимах с одинаковыми настройками. Первый режим использовал функцию восстановления особей, во втором режиме данный оператор игнорировался. На графике зависимости

значения фитнес-функции от количества популяций (рис.4) представлены зависимости значения фитнес-функции от количества популяций. Сплошной линией обозначен график работы программы в первом режиме, точечной – во втором.



Рис.4. График зависимости значения фитнес-функции от количества популяций в разных режимах

Выводы. Проведенный эксперимент доказывает возможность применения генетического алгоритма к задаче составления расписания.

Разработка специализированных операторов может значительно ускорить время работы алгоритма и повысить его результативность, что было показано путем добавления простейшего оператора восстановления.

Перспективами последующих исследований являются модификации генетического алгоритма для решения задачи распределения аудиторного фонда, разработка проблемно ориентированных операторов генетического алгоритма и подбор эффективных параметров алгоритма.

Список литературы

1. Низмова Г. Ф. Математическое и программное обеспечение составления расписания учебных занятий на основе агрегативных генетических алгоритмов: автореф. дисс. канд. техн. наук. / Низмова Г. Ф. – Уфимский государственный авиационный технический университете, Уфа, 2006.
2. Клеванский Н.Н. Формирование расписания занятий высших учебных заведений // Образовательные ресурсы и технологии – 2015. Вып. С.34-45;
3. Будиловский, Д. М. Оптимизация решения задач теории расписаний на основе эволюционно-генетической модели распределения заданий: дисс. канд. техн. наук / Д. М. Будиловский. – Ростов-на-Дону: Издательский центр ДГТУ, 2007.
4. Безуглый М.А., Секирин А.И. Методы повышения эффективности составления расписания в условиях учебного заведения // Международная научно-техническая конференция студентов, аспирантов и молодых ученых Компьютерная и программная инженерия ДонНТУ – 2015.
5. Зимин С.Н. Составление учебного расписания, используя теорию графов // Современные наукоемкие технологии. – 2007. Вып. 11. С.89-90.

УДК 004.891.2

ИССЛЕДОВАНИЕ ПРОГРАММ И ИНТЕРНЕТ-РЕСУРСОВ ДЛЯ СОСТАВЛЕНИЯ КОМПЛЕКСА ФИЗИЧЕСКИХ ТРЕНИРОВОК

Д.В. Свиридов, Д.М. Бочаров

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
dennedswirriws@mail.ru

Аннотация

Свиридов Д.В., Бочаров Д.М. Исследование программ и интернет-ресурсов для составления комплекса физических тренировок. Выполнен анализ распространенных программ и интернет-ресурсов для составления комплекса физических тренировок. Определены их основные достоинства и недостатки, а также выявлены общие недоработки, и рассмотрены методы борьбы с ними.

Ключевые слова: программы, интернет-ресурсы, составление комплекса физических тренировок, физические тренировки.

Постановка проблемы. Все аспекты жизни человека зависят от его физического состояния: успехи в учебе, карьерные достижения, семейное благополучие. В связи с этим, важно поддерживать физическое здоровье в норме. Таким образом, возникает необходимость нахождения оптимального комплекса для физических тренировок для различных специализаций. Для реализации этого необходимо выполнить следующее:

- проанализировать распространенные программы и интернет-ресурсы для составления комплекса физических тренировок, выделить их достоинства и недостатки;
- определить правила, помогающие выбрать наиболее подходящий комплекс физических тренировок.

Цель статьи – провести анализ распространенных программ и интернет-ресурсов для составления комплекса физических тренировок, и определить перечень правил, помогающих выбрать наиболее подходящий комплекс физических тренировок.

Постановка проблемы и актуальность исследования. В современном мире присутствует огромное количество факторов, которые усугубляют здоровье человека: экология, стрессы, ухудшение качества пищи и другие. Все эти факторы выливаются в общее ухудшение физического здоровья человека, что в свою очередь влияет на все факторы жизни. Физическое здоровье влияет также и на психическое здоровье. Успехи в учебе, карьерные достижения, семейное благополучие – все это напрямую зависит от того, насколько человек здоров психологически.

Плохое состояние физического здоровья, в наши дни, повлияло на уменьшение средней продолжительности жизни, а также увеличение числа людей, которые страдают ожирением. Основными причинами этого являются неграмотное питание и, зачастую, отсутствие физических тренировок. Из этого следует, что исследование программ и интернет-ресурсов для составления комплексов физических тренировок остается крайне актуальным.

Анализ распространенных программ и интернет-ресурсов для составления комплекса физических тренировок.

“Бодибилдинг” – Android-приложение для составления программы тренировок в спортзале [1].

В приложении есть два основных раздела: “Упражнения” и “Тренировки”. Если вы хотите составить себе программу тренировок, то вам нужен второй раздел. Там вы можете выбрать подходящую вам 2-, 3-, 4- или 5-дневную программу в зависимости от того, как часто вы хотите посещать спортзал. Есть также и круговая тренировка, в которой задействованы сразу все группы мышц.

“Бодибилдинг” позволяет отслеживать статистику тренировок. То есть смартфон сможет заменить вам тетрадь или блокнот для ведения спортивного дневника.

Преимущества: “компактность” программы (Android, IOS), отслеживание статистики тренировок, вариативность количества дней тренировок.

Недостатки: отсутствие спецификации тренировок, отсутствие внимания к травмам и состоянию здоровья пользователя.

“Fitocracy”. Приложение отчасти похоже на социальную сеть, только для занимающихся спортом людей. Вы заполняете профиль, указываете свой рост, вес и состояние тела. А дальше можем завести друзей, следить за их результатами, давать полезные советы и подбадривать их различными комплиментами. Помимо этого, после выполнения определенных упражнений вы будете набирать очки и получать виртуальные трофеи, которые являются неотъемлемой частью отличной мотивации.

Правда, нужно отметить, что для пользования приложением вам необходимо знать названия упражнений на английском языке. Среди них вы сможете выбрать необходимые именно вам и выполнять их ежедневно или через определенные промежутки времени.

Преимущества: “компактность” программы (Android, IOS), возможность коммуникации с другими людьми, система трофеев.

Недостатки: необходимость знания английского языка, возможность некомпетентного советчика в тренинге, отсутствие экспертного составления комплекса тренировок.

“JEFIT Pro”. У приложения “JEFIT Pro” интерфейс, может быть, выглядит немного загруженным, по сравнению с другими фитнес-приложениями, зато в нем содержится более 1300 различных упражнений с анимацией. В нем

находится множество полезной информации и комплексов, рассчитанных на разные группы мышц. Приложение прекрасно отобразит общую картину состояния вашего тела и подскажет, на что стоит сделать упор при выполнении упражнений. Помимо этого, вы можете поделиться с друзьями своими достижениями и тем самым замотивировать их на новые подвиги в области фитнеса.

Преимущества: “компактность” программы (Adnroid, IOS), обширная библиотека упражнений, система трофеев.

Недостатки: плохая структура интерфейса, отсутствие экспертного составления комплекса тренировок, отсутствие внимания к травмам и состоянию здоровья пользователя.

“Nike Training Club”. Полноценный комплекс тренировок, который содержит планы более чем 60 занятий, разработанных профессиональными спортсменами и тренерами Nike. Общее количество упражнений, представленных в программе, составляет более 100 штук, большинство выполняются только с весом собственного тела. Подробные описания каждого комплекса и входящих в него упражнений, включая видео-демонстрацию. По мере продвижения вперед, предоставляются дополнительные инструкции, и даже награды за выполнение упражнений.

Преимущества: “компактность” программы (Adnroid, IOS), система трофеев.

Недостатки: отсутствие экспертного составления комплекса тренировок, отсутствие внимания к травмам и состоянию здоровья пользователя.

Общие проблемы. В результате проведенного исследования, было выявлено, что основными недостатками программ и интернет-ресурсов для составления комплекса физических тренировок являются: отсутствие экспертного составления комплекса тренировок, отсутствие внимания к травмам и состоянию здоровья пользователя, а также отсутствие спецификации тренировок.

Отсутствие экспертного составления комплекса тренировок является основной проблемой [2], так как программы и интернет-ресурсы предоставляют пользователю большой выбор упражнений, но, чаще всего, пользователь не знает, какое из упражнений более эффективно. Данная проблема решается путем составления программы по предпочтениям пользователя.

Отсутствие внимания к травмам и состоянию здоровья пользователя – не менее важный фактор составления комплекса физических тренировок. Для решения данной проблемы необходимо для каждого нового пользователя проводить анкетирование, собираю всю необходимую информацию для дальнейшего составления комплекса тренировок.

Отсутствие спецификации тренировок менее важная, но необходимая часть составления комплекса тренировок. Тренировки могут быть направлены

как на повышение силы или мускулатуры, так и на похудение или поддержание здоровья. Данная проблема решается указанием направленности тренинга в первоначальной анкете.

Правила выбора комплекса физических тренировок. Правила выбора комплекса физических тренировок наиболее целесообразно представлять в виде базы знаний [3]. База знаний формируется на основе анкетирования. Анкетирование строится из двух типов вопросов: базовые и специальные. Базовыми вопросами являются: пол, возраст, уровень физической подготовки, спецификация тренировок. В специальных вопросах должны быть вопросы о проблемах со здоровьем и травмах: проблемы с основными опорными частями тела (позвоночник; поясница; коленные, локтевые, плечевые суставы; запястья, ступни).

Предпочтительным методом обработки данных из базы является обратный логический вывод [4]. Причиной выбора данного метода является то, что данный метод лучше всего подходит под метод работы программного продукта: программа собирает у пользователя все необходимые данные и использует их для поиска подходящих комплексов упражнений.

Выводы. Произведен анализ распространенных программ и интернет-ресурсов для составления комплекса физических тренировок. Определены правила комплекса физических тренировок, в зависимости от данных, полученных из анкетирования пользователя. Проведена оценка общих проблем и методов их решений. Результаты показали, что для создания оптимальной программы для составления комплекса физических тренировок необходимо провести анкетирование пользователя по рассмотренным вопросам.

Список литературы

1. «Бодибилдинг» – Android-приложение для составления программы тренировок в спортзале [электронный ресурс] // Лайфхакер: [сайт]. [2015]. URL: <https://lifehacker.ru/2015/06/17/body-building/>
2. Кусякова Р.Ф. Критерий составления комплекса физических упражнений для преподавателей // Современные проблемы науки и образования. – 2016. URL: <https://www.science-education.ru/ru/article/view?id=24949>
3. Базы знаний [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: https://ru.wikipedia.org/wiki/База_знаний
4. Прямой и обратный логический вывод [электронный ресурс] // Искусственный интеллект : [сайт]. [2000-2003]. URL: <http://www.rriai.org.ru/pryamoy-i-obratnyiy-logicheskij-vyivod-5.html>

УДК 004.942

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ПРИ РАСПОЗНАНИИ ЛИЦА ЧЕЛОВЕКА ПО ЕГО МИМИЧЕСКИМ ИЗОБРАЖЕНИЯМ

А.П. Семенова, А.С. Миненко

Донецкий национальный технический университет

sam_dntu@mail.ru, nastena-semenova19@rambler.ru

Семенова А.П., Миненко А.С. Информационные технологии при моделировании и распознавания лица человека по его мимическим изображениям. Работа посвящена разработке методов, моделей, алгоритмов и компьютерных средств, для моделирования и распознавания эмоциональной составляющей фотографических изображениях лица и трехмерных моделях человека.

Ключевые слова: аппроксимация, мимические проявления, эмоции, алгоритм, сплайн-аппроксимация.

Введение. Математические методы, компьютерные технологии для анализа и синтеза лица человека, а также, его эмоций исследуются и разрабатываются в ведущих научных организациях мира, в частности, в Массачусетском технологическом институте, Оксфордском, Московском, Санкт-Петербургском университетах, др. В Украине исследованием таких проблем занимаются в Институте кибернетики им. В.М.Глушкова НАН Украины, Киевском национальном университете имени Тараса Шевченко, ГОУВПО Донецкий национальный технический университет и других учреждениях.

Приведем обзор теорий, которые объясняют механизмы возникновения и проявления эмоций [1-5].

Использование контурных моделей на базе сплайн кривых для анализа мимических проявлений эмоций. Для анализа мимических проявлений эмоций, имея в виду формальную модель эмоций состояния человека, предлагается оригинальная технология. Основная ее идея заключается в том, чтобы предложить алгоритм автоматического получения гибких шаблонов контуров бровей, глаз и рта. Для построения этих шаблонов нужно предложить:

- метод который даёт возможность выделить наиболее важные части лица человека, а именно: нос, брови, глаза, губы и щеки.
- алгоритм получения точечных кривых, которые соответствуют контурам лица, которые будут анализироваться;
- алгоритм аппроксимации полученных точечных кривых с помощью сплайн кривых.

Формальная модель эмоций. Для формализации эмоций, чтобы избежать двусмысленностей при их феноменологическом описании, предлагается перейти к изучению ситуаций, в которых эти эмоции возникают [1]. То есть, при определении эмоций, в наиболее общем виде описывается ситуация, в которой они возникают. Различать будем эмоции и их обозначение. Под обозначением будем иметь в виду вектор (Em) (то есть абстрактное понятие) со следующими признаками:

$$Em_i^n = (\xi_1, \xi_2, \xi_3), i = \overline{1,8}, \quad (1)$$

где ξ – бинарный признак, который классифицирует эмоции:

ξ_1 – признак, который определяет знак эмоции – позитивная (1) эмоция или негативная (0). Будем называть эмоцию позитивной, если она возникает в связи с удовлетворением потребности или достижением цели, и, следовательно, негативной – в связи с неудовлетворением или недостижением;

ξ_2 – признак, который определяет время возникновения эмоции относительно действия (предвидевшая (0) и констатирующая (1) эмоции). Предвидящие эмоции возникают до действия связанного с достижением (недостижением) цели, предвидят ее;

ξ_3 – признак, который определяет направление эмоции. По этому признаку выделяют эмоции направленные на себя (1) и направленные на внешние объекты, на других людей (0). Сокращенная функция эмоции лежит в том, что эмоции подготавливают организм к определенному событию которое возникнет. Эмоции служат для решения определенных жизненных трудностей, сложных положений. Каждая эмоция готовит человека к какому-то событию. Это событие может происходить с внешним объектов или с самим человеком. Например, гнев нацелен на устранение преград для достижения цели, и, таким образом, направленный на внешний объект. Грусть готовит человека обходиться той целью, которую не удалось достичь, и направлена на себя.

Рассматривается задача моделирования мимических проявлений эмоций формального описания базовых эмоций. Для поиска пространства характеристических признаков, построения базиса этого пространства, воспроизведение производных эмоциональных состояний со следующим использованием выпуклой комбинации, предлагается следующее:

– создание множества фотографических изображений, на которых актерами воссоздаются ситуации (ξ_1, ξ_2, ξ_3), у которых возникают базовые эмоции, и описание мимики, свойственной этим эмоциям;

– анализ полученного множества с целью выявления областей, которые содержат характеристические признаки эмоций и описание их (используя анатомические признаки и методику Facial Action Coding System);

– создание в пространстве характеристических признаков базиса для следующего расписания по нему произвольных векторов мимических проявлений эмоциональных состояний (как выпуклой комбинации базисных эмоциональных состояний);

– анализ характеристических признаков и ранжирование их по степени влияния в рамках предложенного базиса.

Для анализа полученного множества фотографических изображений с целью выявления областей, которые содержат характеристические признаки эмоций, был использован подход, предложенный авторами методики FACS [1-9]. В ходе исследования, анализируя мимические проявления в разрезе зон лица, было получено 21 характеристический признак, комбинация которых образует базис мимических проявлений эмоций. То есть, мимические проявления эмоций (Em) были представлены в виде вектора:

$$Em_i^n = (\mu_1, \dots, \mu_{21}), i = \overline{1,8}, \quad (2)$$

где $\mu_j \in [0;1]$ – характеристический мимический признак (при $\mu = 0$ – нет признака, а при $\mu = 1$ влияние признака максимально).

Набор 8-ми векторов, указанных таким образом, создает базис (B_{ij} , где $i = 1, \dots, 21, j = 1, \dots, 8$) пространство мимических проявлений эмоциональных состояний.

Таким образом, произвольных вектор $b = (\mu_1, \dots, \mu_{21})$, полученных путем анализа изображений с каким либо эмоциональным состоянием, можно разложить по базису B получить описание эмоции, как выпуклой комбинации:

$$x = (B^T B)^{-1} B^T b \quad (3)$$

Где B – базисная матрица эмоциональных состояний; B^T – транспонирована матрица B ; b – вектор, который описывает мимическое проявление произвольного эмоционального состояния; $x = (a_1, \dots, a_8)$, где a_1 – коэффициенты выпуклой комбинации ($\sum a_1 = 1, a \in [0;1]$) для каждой из 8-ми базовых эмоций.

Так как невозможно мимически определить источник происхождения эмоций (η), получаем следующее множество из 21-й эмоции, которую можно определить предложенными способами.

Использование методов сплайн-аппроксимации для моделирования биологических объектов. В статье предложено для выделения характерных признаков черт лица использовать модели, которые задаются с помощью неравномерных рациональных базисных сплайнов кривых.

Рассмотрим сплайн кривые. Допустим, что задано массив опорных точек p_0, \dots, p_m . Необходимо найти функцию $p(u) = [x(u), y(u), z(u)]^T$, обозначенную на интервале $u_{\min} \leq u \leq u_{\max}$, такую, чтобы она была достаточно гладкой и проходила, в определенном смысле, около опорных точек.

Допустим, что есть последовательность узлов u_0, u_1, \dots, u_n , такая что:

$$u_{\min} = u_0 \leq u_1 \leq \dots \leq u_n = u_{\max}. \quad (4)$$

При использовании аппроксимации сплайнами функция $p(u)$ имеет вид полинома степени d на интервале между соседними узлами:

$$p(u) = \sum_{j=0}^d c_{jk} u^j, u_k < u < u_{k+1}. \quad (5)$$

Таким образом, чтобы найти сплайн степени d необходимо будет найти $n(d+1)$ трёхмерный вектор-коэффициентов c_{jk} . Необходимые для этого уравнения можно получить, рассматривая разного рода ограничения, связанные с непрерывностью функции и критерием близости к опорным точкам. Такой подход к формированию сплайна является глобальным – необходимо решить систему из $n(d+1)$ уравнений относительно $n(d+1)$ неизвестных, а значит, каждый полученный коэффициент будет зависеть от всех опорных точек. Хотя такая методика определения коэффициентов сплайна может обеспечить получение гладкой кривой, которая проходит через заданные опорные точки, но она не очень хорошо согласовывается со спецификой задач компьютерной графики.

Подход, выбранный для формирования В-сплайнов, состоит в том, чтобы обозначить сплайн в терминах базисных функций, каждая из которых отличная от нуля только на интервале в несколько узлов. Итак, можно записать функцию $p(u)$ в виде:

$$p(u) = \sum_{i=0}^m B_{id}(u) p_i, \quad (6)$$

где каждая функция $B_{id}(u)$ есть полином степени d на интервале в несколько узлов и равняется нулю за границами этого интервала. Существует

много способов обозначения базисных функций, но особенное место принадлежит одному из них – методу рекурсивных функций Кокса-де Бура [4]:

$$B_{k,0} = \begin{cases} 1, & \text{если } u_k \leq u \leq u_{k+1} \\ 0 & \text{иначе} \end{cases} \quad (7)$$

$$B_{k,d} = \frac{u - u_k}{u_{k+1} - u_k} B_{k,d-1}(u) + \frac{u_{k+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}(u) \quad (8)$$

Выводы. В работе разработан и реализован алгоритм, который позволяет решить задачу моделирования и распознавания мимических проявлений эмоциональных состояний. Анализ полученных результатов дает основание сделать такой вывод: создан алгоритм, который реализует целостную информационную технологию для моделирования и распознавания мимических проявлений эмоционального состояния на лице человека.

Список литературы

1. Ильин Е.П. Эмоции и чувства / Ильин Е.П. – СПб: Питер, 2001. – 752 с.
2. Джемс У. Психология / Джемс У. – М.: Педагогика, 1991 – 368 с.
3. Ланге Г. Душевные движения / Ланге Г. – СПб.: Питер, 1996. – 180 с.
4. С. De Boor A Practical Guide to Splines / De Boor С. – New York: Springer-Verlag, 1978. – 392p.
5. Friesen W. EMFACS-7: Emotional Facial Action Coding System. Unpublished manual/ W. Frisen, P. Ekman. – California: University of California, 1983.
6. P. Ekman. Facial Action Coding System: A Technique for the Measurement of Facial Movement / P. Ekman, W. Friesen. – Consulting Psychologists Press, Palo Alto, 1978.
7. Кривонос Ю.Г. Моделирование и анализ мимических проявлений эмоций / Ю.Г. Кривонос, Ю.В. Крак, Г.М. Ефимов // Доповіді НАНУ. – 2008. – №12. – С. 51-55.
8. Миненко А.С. Аналитичность свободной границы в одной задаче осесимметричного течения / А.С. Миненко // Укр. мат. журнал. – 1998. – №12. – С. 1693-1700.
9. Миненко А.С. Проблема минимума одного класса интегральных функционалов с неизвестной областью интегрирования / А.С.Миненко // Мат. физика и нелинейная механика. – 1993. – Вып. 16. – С. 48-52
10. Миненко А.С. Вариационные задачи со свободной границей / Миненко А.С. – Киев: Наукова думка, 2005. – 354 с.

УДК 004.056.3

ПРОГРАММНОЕ СРЕДСТВО СОЗДАНИЯ РЕЗЕРВНЫХ КОПИЙ ДАННЫХ

Е.В. Бычкова, В.С. Нестеренко

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
elena.bichkova@gmail.com, nes.vadim.s@gmail.com

Аннотация

Бычкова Е.В., Нестеренко В.С. Программное средство создания резервных копий данных. Описаны современные проблемы сохранения данных. Проанализированы характеристики существующих популярных программных средств для создания резервных копий данных. Приведена авторская идея реализации программного средства: проблемы создания и их решения.

Ключевые слова: система резервного копирования данных, хранилище данных, защита и шифрование данных.

Постановка проблемы. В настоящее время большинство данных хранятся на компьютерных накопителях, содержащих жёсткие магнитные диски, считывающие головки и прочие компоненты. Ввиду присутствия таких факторов как перепады напряжения, износ рабочих частей техники, внешние физические факторы, приводящие к деформации, стало необходимым создавать копии важных данных и хранение их на других накопителях.

Цель статьи – исследование существующих проблем и способов создания резервных копий данных, а также анализ разрабатываемого авторского программного средства для создания резервных копий данных.

Резервная копия данных. Резервная копия – это копия данных на носителе, предназначенная для последующего восстановления данных в случае их повреждения или уничтожения.

Довольно часто возникают ситуации, когда необходимые данные становятся невозвратно потерянными без возможности их восстановления. Например, деформация накопителя данных, его потеря, повреждения, вызванные износом внутренних частей накопителя и пр.

С целью предотвращения потери данных создаётся резервная копия, т.е. копирование данных на другой накопитель с использованием различных технологий, например, создание нескольких резервных копий. Если данные постоянно изменяются (например, в базах данных), то необходимо хранить не одну копию, а несколько, возможно даже с разными версиями (т.е. не несколько одинаковых копий, а несколько копий, созданных в разное время, чтобы иметь возможность восстановить данные, сохраненные в конкретное время). Всё зависит от поставленной цели и варьируется от случая к случаю,

что и является основной проблемой – наличие различных задач выполнения резервного копирования.

Характеристики существующих программных средств для создания резервных копий данных. Существование различных специфических задач положило начало созданию различных систем резервного копирования, отличающихся функционалом. Например: создание копии всего накопителя информации, баз данных на различных носителях, отдельных файлов по расписанию (с указанием количества копий, с шифрованием, с управлением системой резервных копий на разных носителях через систему на сервере, с автоматической возможностью восстановления данных и пр.).

Одной из популярных систем создания резервных копий баз данных для предприятий является SqlBak[1]. Сама система резервного копирования представляет собой программный комплекс, разделенный на две части, одна из которых является серверной, а другая, соответственно, клиентской.

Клиентская часть имеет такие функции как возможность архивирования, подключения к серверной части программного комплекса и отправки данных на сервер.

Серверная часть имеет такой функционал как подключение к аккаунту на сервере нескольких компьютеров с целью централизации создания задач по управлению резервными копиями через меню на сервере. Задачи включают следующие функции:

- настройка гибкого расписания выполнения действий над данными;
- создание резервных копий данных в произвольном количестве и загрузка их на различные накопители или сервера;
- восстановление данных, а также восстановление, просмотр последнего и следующего времени выполнения задачи;
- просмотр выполненных действий над данными;
- поиск конкретной задачи и отправка уведомлений на email о статусе задач;
- просмотр состояния активности пользовательских ЭВМ;
- просмотр количества свободного пространства на серверных и клиентских накопителях.

Следует указать на недостатки данной системы:

- все операции по управлению резервными копиями происходят через сервис SqlBak, что делает невозможным создание резервных копий в локальной сети при отсутствии подключения к сети Интернет;
- отсутствие кроссплатформенности; возможность работы данного сервиса только в операционной системе Windows;
- ориентированность на SQL-базы данных, что приводит к определенным трудностям при настройках создания задач резервного копирования сторонних баз данных и файлов;

– платная ежемесячная подписка за использование данного сервиса в размере 9\$ за один сервер.

Другой популярной системой резервного копирования является Acronis Backup[2]. В отличие от SqlBak, данная система - кроссплатформенная, не ориентирована на один тип базы данных, что позволяет создавать резервные копии различных файлов и операционной системы пользователя. Также присутствуют различные методы шифрования данных.

Как и в любых системах, в данной также присутствуют «минусы»:

- платная подписка за 25\$ в месяц;
- все операции по управлению резервными копиями происходят через сервис Acronis backup, что делает невозможным создание резервных копий в локальной сети при отсутствии подключения к сети Интернет.

При наличии данных недостатков в сервисах SqlBak и Acronis Backup, а также видимого нереализованного потенциала данных сервисов, была начата авторская разработка по созданию новой системы резервного копирования, повторяющей и расширяющей функционал программных продуктов, описанных выше.

Авторская реализации программного средства для создания резервных копий. Так как в настоящее время нет доступного программного средства по управлению резервными копиями с помощью собственного централизованного сервиса, то проблема создания данного программного комплекса является актуальной. Идея заключается не только в повторении функционала популярных систем резервного копирования, но также в расширении функционала и пересмотре ценовой политики, которая будет бесплатной, но с возможностью добровольной поддержки и присутствия так называемых «вех» – этапов сборов сумм, необходимых для реализации дополнительного функционала в будущем. Данная ценовая политика программного продукта, повторяющего и расширяющего функционал других популярных продуктов, станет преимуществом перед другими сервисами резервного копирования данных.

Следующим главным отличием разрабатываемого программного комплекса станет возможность установки серверной части программного комплекса на собственный сервер, т.е. появится централизованная возможность управления созданием резервных копий на различных ЭВМ в локальной сети без подключения к сети Интернет.

Однако данная разработка требует существенных временных затрат.

Проблемы создания и улучшения авторского программного средства.

Разработка программного средства на данном этапе разработки будет представлена в виде одиночного приложения, устанавливаемого на ЭВМ пользователя без возможности централизованного управления созданием резервных копий на различных ЭВМ с помощью сервера.

Разрабатываемое программное средство будет иметь следующий функционал:

- наличие расписания задач создания резервных копий;
- управление задачами;
- возможность загрузки резервных копий на ftp-сервер;
- просмотр статуса задач;
- логирование действий выполнения задач;
- возможность хранения опционального количества резервных копий;
- уведомления о статусе задач.

Программный комплекс разрабатывается на языке программирования Java, вследствие чего и появляется кроссплатформенность. Также использование данного языка программирования позволит перенести функционал из программного комплекса данного этапа разработки в программный комплекс следующего этапа.

При разработке серверной части программного средства можно выделить следующие основные проблемы:

- реализация приёма-передачи сообщения о задаче между клиентской и серверной частями программного средства;
- восстановление выполняемых задач и просьба повтора передачи данных при перезапуске сервера;
- обработка множественной передачи данных в одно и то же время;
- обработка проблемы отсутствия свободного места на накопителе;
- установка защищенного соединения при передаче данных;
- шифрование передаваемых данных;
- шифрование пользовательских данных.

Несмотря на сложность и объёмность вышеперечисленных проблем, разрабатываемый программный продукт будет уникальным на сегодняшний день, благодаря пересмотру ценовой политики и добавлению большего функционала по сравнению с конкурентными решениями.

Выводы. Описаны современные проблемы сохранения данных. Проанализированы характеристики существующих популярных программных средств создания резервных копий данных. Описаны проблемы создания и улучшения авторского программного средства, а также методы их решения.

Список литературы

1. SqlBak [электронный ресурс] // SqlBak - SqlBak: [сайт]. [2017]. URL: <https://sqlbak.com/>
2. Acronis [электронный ресурс] // Acronis | Решения для резервного копирования, восстановления и защиты данных: [сайт]. [2002-2017]. URL: <http://www.acronis.com>

УДК 004.89

СОПОСТАВЛЕНИЕ СПОСОБОВ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТОВЫХ ДОКУМЕНТОВ

Бурлаева Е.И., Ермоленко Т. В.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: ekaterina0853@mail.ru

Аннотация

Бурлаева Е.И., Ермоленко Т.В. Анализ методов классификации текста. Приведен сравнительный анализ различных исследований в области классификации текстовых документов. На основе опубликованных результатов проведен сравнительный анализ алгоритмов классификации текстовых документов, в результате которого сделан вывод о необходимости в повышении качества и скорости классификации текста за счет комбинирования преимуществ этих методов машинного обучения.

Общая постановка проблемы. С ростом текстовой информации в электронном виде задача автоматической классификации текстов приобретает все большую актуальность. Информационные системы являются неотъемлемой частью повседневной человеческой деятельности.

Классификация текстовых документов является задачей автоматического определения документа в одну или несколько категорий (рубрик, тематик) на основании содержания документа. В зарубежной литературе получил широкое распространение термин Text Categorization [1, 2].

Целью классификации документов является предоставление мощных инструментов для превращения неструктурированной коллекции документов в структурированную согласно схеме, предложенной в качестве входных данных, тем самым облегчив хранение, поиск и просмотр по данной коллекции [2].

Задача классификации заключается в автоматическом распределении текстов или фрагментов текстов по заранее заданному множеству классов. В общем случае задача классификации может быть формализована следующим образом. Имеется множество объектов $T = \{t_i\}$, не обязательно конечное, а также множество $C = \{c_i\} i = 1..N_c$, состоящее из N_c классов объектов. Каждый класс c_i представлен некоторым описанием F_i , имеющим некоторую внутреннюю структуру. Процедура классификации f объектов $t \in T$ заключается в выполнении преобразований над ними, после которых либо

делается вывод о соответствии t одной из структур F_i , что означает отнесение t к классу c_i , либо вывод о невозможности классификации t .

Целью данной работы является исследование эффективности методов машинного обучения, используемых для классификации текстовых документов.

Обзор существующих решений. Сравнение методов построения классификаторов является довольно сложной задачей по причине того, что разные входные данные могут приводить к различным результатам. Классификация является многоэтапной процедурой, на каждом шаге которой должна решаться отдельная задача выбора наиболее адекватного способа реализации, влияющего на последующие этапы

В 80-е годы XX века наиболее популярным подходом к классификации текстовых документов был метод, основанный на базах знаний. В данном методе эксперт в предметной области работал над созданием системы автоматической рубрикации текстов. Система состояла из правил (условий) и с учетом этих условий документы разделялись по рубрикам. Однако это был очень затратный метод в плане человеко-часов. В 90-е данный метод был улучшен использованием машинного обучения. Модель обучалась на ранее размеченной коллекции документов и на ее основе классифицировала новые документы. Методы с использованием машинного обучения и баз знаний давали лучшие результаты чем просто методы на основе баз знаний [2].

Существуют различные подходы к решению задачи автоматической классификации текстовых документов: методы, основанные на базах знаний, методы, основанные на машинном обучении. Рассмотрим эффективность классификации при использовании методов автоматической классификации текстовых документов на основе баз знаний и на основе машинного обучения.

Методы на основе баз знаний. Данные методы используют базу знаний для построения модели рубрикации текстовых документов, например, тезаурус или онтологию. На основе этой базы эксперты строят правила отнесения документов к классам:

– метод рубрикации на основе тезауруса РуТез предложенный М. Агеевым и Н. Лукашевичем [3]. Тезаурус – это иерархическая сеть, состоящая из понятий, терминов и отношений между ними (например, синонимия, антонимия и др.). Данный метод показывает высокие результаты на наборе данных РОМИП - 72,9%. Проблема метода, основанного на тезаурусе, заключается в том, что он требует большого количества человеко-часов для описания каждой рубрики формулой.

– метод автоматической рубрикации текстовых документов, основанный на онтологии, предложенный Janik M.[4]. Онтология – это формальное описание некоторой области знаний, сделанное с помощью концептуальной схемы, которая состоит из классов (в случае задачи классификации терминов или понятий), связей между ними и правил, принятых в этой области [5].

Рассмотренный метод на коллекции новостных документов CNN5 получил еще более высокий результат, а именно 80,77 %. Основное преимущество рассмотренного метода в том, что для него не нужна размеченная коллекция текстовых документов, достаточно иметь только сами документы. Однако недостатком этого метода является то, что он требует ручного отражения рубрик коллекции на рубрики онтологии.

Таким образом, создание достаточно большой, последовательно отрубризированной текстовой коллекции является серьезной организационной проблемой.

Методы на основе машинного обучения. Процедура автоматической классификации текстов методами машинного обучения включает две основные части: представление текстов в виде векторов признаков и построение классификатора на созданном массиве векторов. Необходимость представления текстов в виде векторов признаков определяется тем обстоятельством, что все методы классификации требуют, чтобы классифицируемые объекты были представлены в виде последовательностей чисел одинакового размера и одинакового формата.

Текст можно рассматривать как иерархическую структуру и анализировать на любом уровне как последовательность отдельных составляющих его элементов (символов, словоформ, грамматических классов и т.д.) или групп элементов длиной n , называемых n -граммами. Исходя из того, какой элемент текста рассматривать в качестве компонента вектора признаков, выделяют два основных способа представления текста: полиграммная модель (или модель n -грамм) и модель терм – документ.

Основная идея модели терм-документ очень проста: текст описывается лексическим вектором в евклидовом пространстве, каждой компоненте вектора соответствует некоторый объект, содержащийся в тексте (слово, словосочетание, названия фирм, должности, имена и т. п.), который называется термом. Каждому используемому в тексте терму ставится в соответствие его вес (значимость), определяемый на основе статистической информации об его встречаемости в отдельном тексте. Размерность вектора – это количество термов, которые встречаются в текстах.

Использование векторно-пространственной модели представления текста не лишено недостатков. К факторам, усложняющим или делающим невозможным применение методов машинного обучения для автоматической классификации текстов, можно отнести следующие [19]:

– Чаще всего для улучшения качества классификации необходимо удалить из текста стоп-слова. При удалении различного набора стоп-слов получается разный результат работы алгоритма.

– При отсутствии простейшей дополнительной обработки, такой как морфологический анализ, существенно снижается качество классификатора.

Причина в том, что разные формы одного слова считаются разными терминами, а морфологический анализ – весьма нетривиальная задача, требующая для ее решения привлечения лингвистов.

– Размерность векторов признаков зависит от общего количества терминов в обучающей выборке текстов, поэтому полученное признаковое пространство будет сильно разреженным и высокой размерности, что в реальных задачах приводит к необходимости разрабатывать альтернативные структуры данных, отличные от векторов.

– Словарь терминов может не охватывать всех документов, подлежащих классификации, так что анализируемые документы могут содержать значимые термины, не вошедшие в обучающую выборку. Это обстоятельство отрицательно сказывается на адекватности модели.

Классическими методами построения текстовых классификаторов на основе методов машинного обучения являются деревья решений, линейные классификаторы, метод Rocchio, метод опорных векторов (SVM), байесовский наивный классификатор, байесовские сети, нейросети различной архитектуры. В данном обзоре нет описания вышеперечисленных методов, цель работы – исследовать эффективность их использования для классификации текстов.

Большое число исследований эффективности методов автоматической рубрикации проводится на популярной коллекции финансовых сообщений информационного агентства Рейтер - Reuters-21578 [6], которая была специально создана для тестирования методов автоматической рубрикации текстов. Для этой коллекции характерны следующие особенности:

– тексты сообщений небольшие по величине и принадлежат узкой предметной области финансовых и биржевых новостей;

– рубрикатор, включающий 135 рубрик, относительно прост, без иерархии, причем обычно [7, 8] для тестирования используются лишь 10 наиболее частотных рубрик;

– присвоение рубрик проводилось с контролем качества работы экспертов.

В частности, 40% из имеющихся 21578 документов не рекомендуются к использованию из-за того, что присвоение рубрик к ним признано некачественным. Оставшиеся 12902 документа помечены как «качественно отрубрицированные». Для 10 наиболее частотных рубрик коллекции Reuters-21578 результаты применения машинного обучения весьма высоки - в среднем около 84%. Сравнительные исследования эффективности методов машинного обучения на коллекции Reuters-21578 [9, 10, 11] показали, что наиболее эффективным методом является метод опорных векторов SVM по сравнению с методами Байеса, ближайших соседей, Rocchio, деревьев решений C4.5, нейронных сетей, байесовских сетей [12].

С результатами других экспериментов можно ознакомиться в [13] (сравниваются байесовские сети, деревья решений, методы Байеса и опорных

векторов) и [14] (сравниваются методы Байеса, Rocchio, k -ближайших соседей, опорных векторов и деревья решений). Автор статьи [15] отмечает, что его результаты немного отличаются от опубликованных в [14], но классификатор, построенный на базе метода опорных векторов, также имеет небольшое преимущество перед остальными.

Анализ опубликованных работ позволяет отметить следующее:

- большинство методов тестируется на коллекции Reuters-21578, состоящих из коротких сообщений с очень простым рубрикатом;
- для тестирования выбираются только рубрики с большим количеством примеров.

Существуют востребованные практикой задачи, в которых применение описанных в литературе методов машинного обучения неэффективно [12, 16]. В реальных задачах для построения качественной обучающей выборки по-прежнему часто используются ручной труд экспертов, а также системы рубрикации, основанные на вручную задаваемых правилах [17, 18].

На основании представленного выше краткого анализа можно сделать выводы о том, что наиболее быстрым является метод SVM.

Рассмотренные классические методы служат отправной точкой для разработки более эффективных методик. Широко применяемая технология – использование комбинации классификаторов, что позволяет повысить точность классификации [20]. Идея заключается в построении S классификаторов N_1, \dots, N_C и объединении их результатов классификации. В машинном обучении широкое распространение получили методы «bagging» и «boosting» [21], которые основаны на изменении обучающего множества.

В методе «bagging» построение S классификаторов N_i осуществляется независимо друг от друга на обучающих множествах, полученных из исходного случайной заменой документов (размер обучающего множества остается прежним, просто одни документы отсутствуют, а другие встречаются несколько раз). Результат классификации определяется простым большинством голосов элементов комбинаций классификаторов.

Идея метода «boosting» заключается в последовательном построении S классификаторов, при котором на классификатор Φ оказывают влияние N_1, \dots, N_{i-1} . Классификатор N_i строится на исходном обучающем множестве, документы t_j которого участвуют в обучении с некоторыми весовыми коэффициентами h_j^i . После обучения классификатор N_i проверяется на исходной обучающей выборке и происходит пересчет коэффициентов. Коэффициент h_j^{i+1} уменьшается, если документ t_j классифицирован верно, и увеличивается в противном случае. В методе «boosting» используется взвешенная линейная комбинация голосов элементов комбинаций классификаторов.

В работе [22] приводятся теоретическое обоснование и результаты эксперимента, которые показывают, что комбинации независимых классификаторов

наиболее эффективны. Все большую популярность получают методы, в которых обучение отдельных элементов комбинаций классификаторов осуществляется независимо на различающихся подмножествах признаков [23,24].

Выводы. Методы классификации, основанные на знаниях, требуют трудоемкой работы специалистов по созданию описаний рубрик, а также многоэтапной процедуры тестирования этих описаний. Таким образом, при выполнении практических задач рубрикации необходимо анализировать имеющиеся ресурсы для правильного выбора подходящей технологии.

Существенным условием эффективной работы методов машинного обучения является наличие большой, качественно отрубрицированной коллекции. Кроме того, часто методы машинного обучения порождают трудно интерпретируемые результаты, что затрудняет анализ результатов экспертами и использование результатов в автоматизированных режимах.

При решении сложных задач классификации часто оказывается, что ни один из алгоритмов не обеспечивает желаемого качества восстановления зависимости. Таким образом для достижения желаемого результата требуется применение не одного метода анализа текстов, а целого набора взаимосвязанных методов, обеспечивающих комплексное решение поставленной задачи. В таких случаях имеет смысл строить композиции алгоритмов, в которых ошибки отдельных алгоритмов взаимно компенсируются.

Литература

1. Sebastiani F. Machine Learning in Automated Text Categorization // ACM Computing Surveys. 2002. V. 34, N. 1. pp. 1-47.
2. Sebastiani F. Text Categorization // Text Mining and Its Applications. WIT Press, Southampton, UK, 2005. pp. 109-129.
3. Агеев М. С., Добров Б. В., Лукашевич Н. В. Автоматическая рубрикация текстов: методы и проблемы // Учёные записки Казанского государственного университета. – 2008. – Т. 150. – №. 4. – С. 25-40.
4. Janik M., Kochut K. J. Wikipedia in action: Ontological knowledge in text categorization // Semantic Computing, 2008 IEEE International Conference on. – IEEE, 2008. – pp. 268-275.
5. Лапшин В. А. Онтологии в компьютерных системах // М.: Научный мир. – 2010
6. Агеев М.С., Журавлев С.В., Ламбурт В.Г. Подготовка Web-версий традиционных изданий // Открытые Системы. 2000. №12. С.31-35.
7. Агеев М.С., Добров Б.В., Макаров-Землянский Н.В. Метод машинного обучения, основанный на моделировании логики рубрикатора. // RCDL'2003 Электронные библиотеки: перспективные методы и технологии, электронные

коллекции: Пятая всероссийская науч. конф. — Санкт-Петербург, 2003. С. 150-158.

8. Агеев М.С., Журавлев С.В., Захаров В.А. Опыт построения полнотекстовой информационной системы на базе автоматизированной лингвистической обработки текстов с использованием Интернет-технологий Oracle // Научный сервис в сети ИНТЕРНЕТ: Труды Всероссийской науч. конф. Новороссийск, 1999.

9. Ageev M., Dobrov B., Loukachevitch N. Text Categorization Tasks for Large Hierarchical Systems of Categories // SIGIR 2002 Workshop on Operational Text Classification Systems / Eds. F.Sebastiani, S.Dumas, D.D.Lewis, T.Montgomery, I.Moulinier Univ. of Tampere, 2002. pp. 49-52.

10. Ageev M., Dobrov B., Makarov-Zemlyanskii N. On-line Thematic and Metadata Analysis of Document Collection // New Trends in Intelligent Information Processing and Web Mining'2004: Proceedings of the International Conference. Springer, Advanced in Soft Computing. Zakopane, Poland, May 2004. pp. 279-286.

11. Dumais S., Lewis D., Sebastiani F. Report on the Workshop on Operational Text Classification Systems // SIGIR-02, Tampere, Finland. –2002. pp. - 4. // URL: <http://www.sigir.org/forum/F2002/sebastiani.pdf>.

12. Rose T., Stevenson M., Whitehead M. The Reuters Corpus Volume 1 – from Yesterday News to tomorrow's Language//In Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, May 2002. pp. 29-31.

13. Dumais S.T., Platt J., Heckerman D., Sahami M. Inductive learning algorithms and representations for text categorization // Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management, Bethesda, MD. 1998. pp. 148-155.

14. Joachims T. Text categorization with support vector machines: learning with many relevant features // Proceedings of ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany. 1998. pp. 137-142.

15. Yang Y., Liu X. A re-examination of text categorization methods // Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, CA. 1999. P. 42-49.

16. Wasson M. Classification Technology at LexisNexis //SIGIR 2001 Workshop on Operational Text Classification. // URL: <http://www.daviddlewis.com/events/otc2001/presentations/otc01-wasson-paper.txt>

17. Hofmann T. Probabilistic latent semantic analysis. // In Proc. of Uncertainty in Artificial Intelligence, UAI'99, Stock-holm, 1999. pp. 289-296.

18. Quinlan J. Induction of decision trees // Machine Learning. 1998. Vol. 1, N. 1. pp. 81-106.

19. Dumais S., Lewis D., Sebastiani F. Report on the Workshop on Operational Text Classification Systems // SIGIR-02, Tampere, Finland. –2002. pp. - 4. // URL: <http://www.sigir.org/forum/F2002/sebastiani.pdf>.

20. Dietterich T. G. Machine learning research: four current directions // AI Magazine. 1997. V. 18. pp. 97-136.

21. Quinlan J. R. Bagging, Boosting, and C4.5 // Proceedings of AAA/IAAI. 1996. pp. 725-730.

22. Oza N. C, Turner K. Decimated input ensembles for improved generalization // Proceedings of the International Joint Conference on Neural Networks, Washington, DC. 1999.

23. Bryll R. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets // Pattern Recognition. 2003. V. 36. pp. 1291-1302.

24. Bay S. D. Nearest neighbor classifiers from multiple feature subsets // Intelligent data analysis. 1999. V. 3. pp. 191-209.

УДК 004.932

СИСТЕМА РАСПОЗНАВАНИЯ ЛИЦ ПРИ ПОМОЩИ СИСТЕМ ВИДЕОНАБЛЮДЕНИЯ. АЛГОРИТМ ПОИСКА ЛИЦ

Ковалёв Д.В., Бычкова Е.В.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
elena.bichkova@gmail.com

Аннотация

Ковалёв Д.В., Бычкова Е.В. Система распознавания лиц при помощи систем видеонаблюдения. Алгоритм поиска лиц. Проанализированы современные виды распознавания образов при помощи систем видеонаблюдения. Рассмотрены основные технологии, тенденции и перспективы развития распознавания лиц.

***Ключевые слова:** распознавание образов, системы безопасности, видеоаналитика, алгоритм поиска лиц, биометрическая идентификация, каскады Хаара.*

Постановка проблемы. Видеонаблюдение зарождалось как охранная замкнутая система, предназначенная только для решения вопросов безопасности. Ограничения аналогового видеонаблюдения не позволяли использовать оборудование иначе. Интеграция видеонаблюдения с цифровыми системами открыла возможность автоматизировано получать различные данные, анализируя последовательность изображений.

Цель статьи – проанализировать основные принципы интеллектуальных систем видеонаблюдения, основанных на методах и алгоритмах распознавания образов и обработки изображений.

Анализ литературы. Проведен анализ программно-аппаратных средств современных интегрированных систем безопасности. В статье описаны основные виды и технологии распознавания лиц, а также алгоритмы поиска лиц.

Комплексные системы безопасности. Программно-аппаратные средства современных интегрированных систем безопасности способны решать задачи любой сложности на всевозможных объектах промышленного, социального и бытового назначения. Очень важными инструментами охранных комплексов являются системы видеонаблюдения; требования, предъявляемые к функциональным возможностям сегмента, неуклонно растут.

Единая платформа интегрированных систем безопасности включает в себя модули охранно-пожарного оборудования, контроля и управления доступом, видеонаблюдения или охранного телевидения (СОТ). Функции последнего до недавнего времени ограничивались видео-мониторингом и регистрацией ситуации на объекте и прилегающей территории, архивацией и хранением данных. Классические видеосистемы обладают целым рядом существенных недостатков, основным из которых является человеческий фактор. Неэффективная работа оператора при трансляции большого объема информации. Невозможность оперативного вмешательства, несвоевременный анализ. Значительные временные затраты для поиска и идентификации события. Развитие цифровых технологий привели к созданию "умных" автоматизированных систем.

Видеоаналитика. Базовым принципом интеллектуальной системы видеонаблюдения является видеоаналитика - технология, базирующаяся на методах и алгоритмах распознавания образов и обработки изображения, автоматизированного сбора данных в результате анализа видеопотока. Такое оборудование без участия человека способно обнаружить и отследить в реальном времени заданные цели (автомобиль, группу людей), потенциально опасные ситуации (задымление, возгорание, несанкционированное вмешательство в работу видеокamer), запрограммированные события и своевременно выдать тревожный сигнал. За счет фильтрации не представляющих интереса видеоданных значительно снижается нагрузка на коммуникационные каналы и архивную базу. Наиболее востребованное средство видеоаналитики - система распознавания лиц. В зависимости от выполняемых функций и поставленных задач к оборудованию предъявляются определенные требования.

Методы распознавания образов. Любые биометрические системы распознавания базируются на выявлении соответствия считываемых физиологических характеристик личности определенному заданному шаблону. Сканирование происходит в режиме реального времени. IP-камера транслирует видеопоток на терминал, и система распознавания лиц определяет соответствие изображения хранящимся в базе данных фотографиям. Существует два основных метода. Первый основан на статических принципах: по результатам обработки биометрических параметров создается электронный образец в форме уникального числа, соответствующего конкретной личности. Второй метод моделирует "человеческий" подход и характеризуется самообучаемостью и робастностью. Идентификация личности по видеоизображению происходит с учетом возрастных изменений и других факторов (наличие головного убора, бороды или усов, очков). Такая

технология позволяет работать даже со старыми фотографиями и, в случае необходимости, с рентгеновскими снимками.

Алгоритм поиска лиц. Самая распространенная методика детектирования лиц – с использованием каскадов Хаара (наборов масок). Маска представляет собой прямоугольное окно с различной комбинацией белых и черных сегментов (рис. 1). Механизм работы программы следующий: видеокадр покрывается набором масок и по результатам свертки (подсчет пикселей, попавших в белые и черные секторы) подсчитывается разность, сравниваемая с некой пороговой величиной. Для улучшения работы классификатора создаются положительные (кадры, где присутствуют лица людей) и отрицательные (без таковых) обучающие выборки. В первом случае результат свертки выше порогового значения, во втором - ниже. Детектор лиц с допустимой погрешностью определяет сумму свертков всех каскадов и при превышении порога сигнализирует о присутствии лиц в кадре.

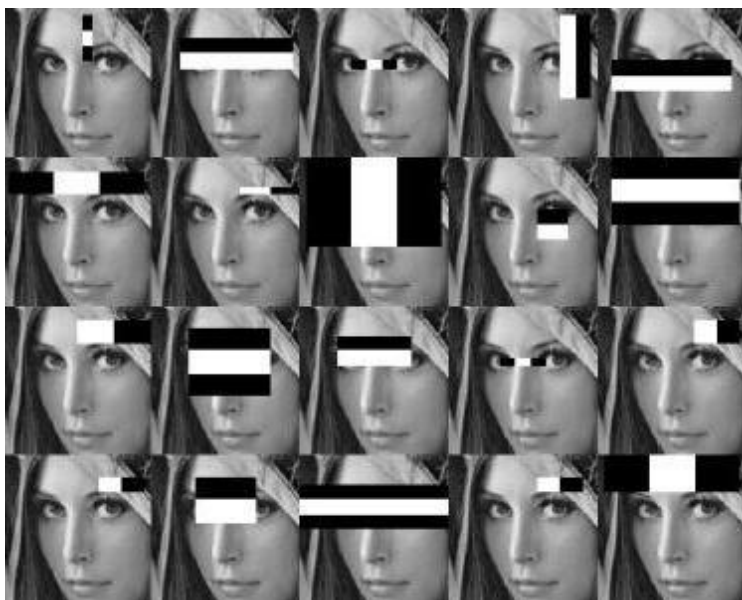


Рисунок 1 – Методика детектирования лиц с использованием масок Хаара

Следующий алгоритм – это метод сравнения на графах. Программа представляет лица, отсканированные системой видеонаблюдения, в виде

графов с определенными вершинами и гранями (рис. 2). Сутью метода является эластичные изменения сопоставления ребер, одно из которых является эталонным, а другие деформируются с целью оптимальной подгонки к оригиналу. В различных алгоритмах графы могут иметь структуру прямоугольника, где ключевые точки привязаны к двумерным координатам, или трехмерную структуру неравномерных многоугольников, вершины которых связаны с основными антропометрическими показателями лица.

Недостатком данного метода является довольно существенная сложность процесса распознавания и, как следствие, высокие требования, которые предъявляются к вычислительным мощностям обслуживающей техники. Процедура запоминания новых эталонов имеет низкую технологичность. Кроме того, существует прямая зависимость скорости работы системы с объемом базы данных. Рекомендуется использовать для систем распознавания, объем которых не превышает 500 эталонов.



Рисунок 2 – Метод сравнения на графах: а) прямоугольный граф;
б) антропометрическая решетка

Тенденции и перспективы развития. Очень востребованы системы распознавания лиц в банковском секторе. По итогам прошлого года руководству "Почта Банка" после установки в своих офисах 50000 интеллектуальных видеокамер удалось сэкономить миллионы рублей за счет профилактики мошенничества в сегментах кредитования и платежей. Специалисты утверждают, что к 2021 году будет создана необходимая инфраструктурная сеть, и любые операции в банкоматах станут возможными

только после биометрической идентификации лица клиента. В ближайшее десятилетие высокие технологии позволят открыть сеть магазинов полного самообслуживания: покупатель проходит перед витринами, выбирает понравившийся товар и уходит. Система распознавания лиц и образов определит личность покупателя, покупки и спишет с его счета необходимую сумму. Ведутся работы по созданию систем распознавания психоэмоционального состояния. Анализ человеческих эмоций будет востребован в мультимедийных сферах: анимации, кинематографе, индустрии создания компьютерных игр.

Выводы. Распознавание лиц – одна из самых сложных задач в области видеоаналитики. С одной стороны, все понятно и давно используется. С другой стороны, решения идентификации в толпе людей все еще стоят очень дорого и не дают абсолютной точности.

При распознавание лиц необходимо обеспечить устойчивое слежение за несколькими объектами в кадре даже в случае их совместного пересечения или временного «пропадания» за препятствием; считать любое количество объектов, пересекающих определенную зону и учитывать направления пересечения; знать, когда появляется и исчезает предмет/объект в кадре. Вариативность лиц людей огромная, поэтому и система имеет невысокую вероятность в распознавании.

Следовательно, при решении данного класса задач имеются самые широкие перспективы и возможности.

Список литературы

1. Система распознавания лиц при помощи систем видеонаблюдения [электронный ресурс] // Технологии распознавания: [сайт]. [2012]. URL: <http://fb.ru/article/285754/sistema-raspoznavaniya-lits-pri-pomoschi-sistem-videonablyudeniya-algoritm-poiska-lits>

2. Видеоаналитика: распознавание лиц, детектор очередей, поиск объектов на видео. [электронный ресурс] // Путь Ivideon к видеоаналитике: [сайт]. [2013]. URL: <https://geektimes.ru/company/ivideon/blog/286168/>

3. Видеонаблюдение распознавание лиц – алгоритмы и программы [электронный ресурс] // Метод сравнения на графах: [сайт]. [2015]. URL: <http://ohranivdome.net/videonablyudenie/programmnoe-obespechenie/videonablyudenie-raspoznavanie-lic-algoritmy-i-programmy.html>

4. Видеонаблюдение и системы безопасности [электронный ресурс] // Видеоаналитика и распознавание лиц: [сайт]. [2014-2017]. URL: http://video-praktik.ru/st_videonanalitika.html

УДК 004.023

МЕТОДЫ АНАЛИЗА СОЦИАЛЬНЫХ ГРУПП НА БАЗЕ ИХ ПОВЕДЕНИЕ В СОЦИАЛЬНЫХ СЕТЯХ

Черкасс Д.В., Кравец Т.Н.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа

cherkassdmitry@gmail.com

Аннотация

Черкасс Д.В., Кравец Т.Н. Методы анализа социальных групп на базе их поведение в социальных сетях. В данной статье излагаются особенности методов анализа социальных сетей, предложены новые методы, а также их программная реализация и визуализация полученной информации в виде графов. На примере социальных сообществ показан метод нахождения самых влиятельных и активных участников, а также доказана необходимость их нахождения. Предложены задачи, которые необходимо решить для анализа социальных сетей.

Ключевые слова: социальная сеть, графовые модели, графовые сети, методы анализа социальных сетей.

Введение. Социальные сети становятся крупнее и имеют все более сложную структуру. Визуализация помогает естественным образом свести воедино информацию о сетях и сделать ее более доступной для понимания. Визуализация в сочетании с взаимодействием помогают аналитикам в описании социальных сетей.

Целью этого направления является также поиск ответа на вопрос, как различные модели могут быть использованы для изучения различных аспектов сетей, таких как структура и семантика. Важным является создание алгоритмов, сочетающих методы анализа и визуализации, чтобы улучшить понимание структуры и динамики сети. При анализе социальных сетей решается довольно большой круг задач и применяются методы из различных областей знаний.

Один из самых известных примеров анализа сетей был проведен в 1970-е гг. американским социологом Марком Грановеттером. Он показал, что для многих социальных задач, таких как поиск работы, слабые связи оказываются намного эффективнее, чем сильные. Этот эффект он назвал «силой слабых связей».

Постановка задачи. Цель статьи заключается в рассмотрении и разработке методов, которые анализируют информацию по данным из социальной сети Вконтакте и визуализируют полученные данные в виде графа,

для выявления самых влиятельных пользователей для распространения информации. Основой для работы стали исследования, проведенные в области социальных сетей и существующие похожие приложения. На данный момент существует множество приложений построения графов по данным из социальной сети Вконтакте, но зачастую эти приложения кроме графов друзей никакую информацию не предоставляют. Разработка новых методов и приложения, заключается не только в удобной для восприятия визуализации данных из социальной сети Вконтакте, но и анализ и предоставление пользователю полученной информации.

Входными данными приложения анализа будет служить информация о пользователях и группах из социальной сети Вконтакте. Выходными данными является следующий набор графов:

- граф друзей;
- граф друзей друзей;
- граф друзей с количеством лайков или репостов;
- граф пересечения пользователей разных групп Вконтакте;
- поиск самых влиятельных и активных пользователей;

После анализа информации полученной в ходе построения графов, можно будет узнать, какие пользователи и группы самые надежные в распространении информации, что может применяться как в таргетной рекламе, так и для завлечения новых пользователей в группы в Вконтакте.

Описание используемых методов анализа. В качестве первого метода был исследован и реализован наиболее популярный и простой способ анализа – это визуализация в виде графа друзей и друзей-друзей пользователя. Такой граф наглядно демонстрирует всю масштабность или незначительность социальных связей человека, что показывает, насколько он социально активен.

Вторым методом исследования социальной активности пользователя, является подсчет количества оставленных другими людьми лайков и репостов. Данный метод показывает, насколько человек интересен другим пользователям сети и стоит ли его использовать в качестве распространителя информации. Если у человека много друзей, но количество лайков и репостов мало, то возможно в друзьях находятся «фейки» или «боты», которые не чувствуют в социальной деятельности.

В качестве третьего способа был разработан метод, который подсчитывает количество общих подписчиков выбранной группы. Его суть заключается в том, чтобы узнать сколько подписчиков конкретной группы подписаны на другие группы. Если количество таких пользователей в других группах велико, то стоит задуматься о рекламе своей группы в таких сообществах, так как возможно там будут люди имеющие вкусы или мировоззрение, которые соответствуют вашей группе, и они смогут ею заинтересоваться. Пример такого графа представлен на рисунке 1.

Последним разработанным методом является подсчет «влияния» и «активности» конкретного пользователя. Под активностью стоит подразумевать сколько пользователей сети он может распространить информацию в одиночку. Понятие влияние намного сложнее, оно характеризует, скольким пользователям конкретный человек сможет распространить информацию не только в одиночку, но и с помощью друзей из социальной сети. Данный метод точно демонстрирует, насколько человек компетентен в распространении информации, так как чем больше влияние, тем больше людей смогут узнать информацию, отправленную им.

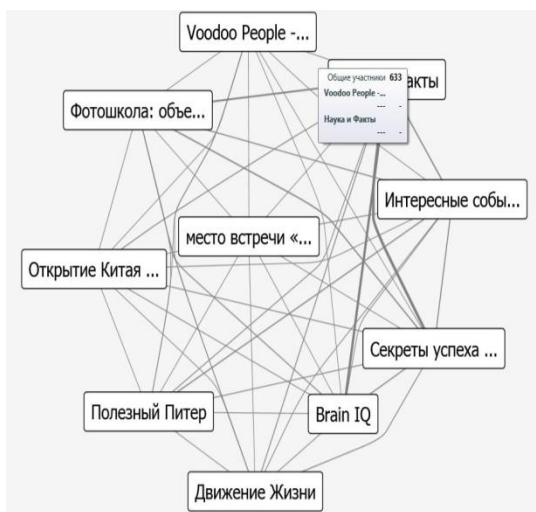


Рисунок 1 – Граф общих подписчиков группы

Сравнение методов с существующими приложениями. В настоящее время существует множество приложений и сервисов для анализа информации в социальных сетях, а также визуализации этой информации в виде графов. Большинство из них несут в себе лишь развлекательный характер и не предоставляют пользователю важной информации. Также существуют платные сервисы, которые могут полностью проанализировать ваше сообщество или вашу персональную страницу, но такие сервисы достаточно дорогостоящие и предоставляют вам только информацию на один раз, без предоставления самого приложения анализа.

В данный момент приложений, которые визуализируют информацию, полученную из социальной сети Вконтакте очень мало и у всех очень скудный функционал, что и дало толчок к созданию приложения именно для

визуализации данных в виде графа. К примеру, приложение «Интерактивный граф друзей». Это приложение создает граф с вашими друзьями в качестве вершин. С практической точки зрения это приложение не дает пользователю новой информации, ведь он и без использования этого приложения знает и может посмотреть своих друзей на своей персональной странице, но конечно не в виде красивого графа. Как итог, данное приложения несет в себе только развлекательный характер «на один раз», и необходим лишь для эстетически красивого отображения друзей. Примерно таким же функционалом обладают и другие подобные приложения с использованием графов, что делает разработку собственного приложения с более богатым функционалом актуальным.

В отличие от предыдущего, разрабатываемое приложение имеет такие полезные функции, как создание графов не просто с друзьями в качестве вершин, но еще анализирует сколько «лайков» или «репостов» оставил вам каждый из друзей. Еще одной полезной функцией, является нахождение «активных» и «влиятельных» пользователей. Это дает возможность узнать, кто из пользователей наиболее компетентен в качестве распространения информации. Главной функцией разрабатываемого приложения, является анализ общих подписчиков сообществ. Выходным графом данной функции является список сообществ, на которые подписаны участники анализируемого сообщества. Полученную информацию можно использовать для рекламы своего сообщества в списке полученных, что гарантирует приток новых подписчиков. При исследовании приложений анализа социальной сети Вконтакте данная функция отсутствовала, что делает разрабатываемое приложение еще актуальней.

Конечно существуют сервисы с детальным анализом практически любой информации в социальной сети Вконтакте и имеют большой функционал, но такие приложения зачастую платные и предоставляют информацию за конкретный период, и при повторном анализе, допустим через месяц, придется снова платить. Такими приложениями зачастую пользуются только крупные сообщества, которые не являются целевой аудиторией для разрабатываемого приложения. Также они не используют графы для предоставления полученной информации, что делает сравнение с разрабатываемым приложением бессмысленным.

Из всех существующих приложений для анализа информации Вконтакте, предоставляющих эту информацию в виде графа, нет ни одного, с подобным функционалом, что делает его привлекательным для использования не только для развлечений, но и для серьезного анализа своей страницы или сообщества. Особенно данное приложение заинтересует рекламодателей или развивающиеся сообщества, которые заинтересованы в увеличении своей аудитории.

Выводы. Среди наиболее актуальных проблем анализа социальных сетей можно назвать создание интегральной теории сетей, так как разнообразие подходов к их анализу приводит к проблеме объединения результатов, полученных с помощью каждого из подходов, в рамках единого исследования. Необходимо подобрать универсальный набор мер расстояний, позволяющих определять дистанции между элементами сети, а также необходимо систематизировать различные меры полноты сетей.

В исследовании атрибутов участников сети, связей между ними, выявлении закономерностей построения сетей между участниками могут быть полезны новые методы статистического анализа, комбинации их с алгоритмами из теории графов.

Иногда отношения между участниками сети удобно рассматривать как вероятностные (стохастические) характеристики для описания процесса эволюции сетей. Задачу поиска человека в социальной сети можно в некотором смысле сравнить с задачей поиска релевантного документа в коллекции документов, снабженных ссылками.

Используя информацию, полученную с помощью предложенных методов анализа, можно сделать выводы о том, какие объекты наиболее влиятельны в сети, какие способствуют наиболее эффективному распространению информации в ней. Если, например, необходимо внутри этой сети распространить некую информацию, имея ограничение в количестве обращений, то очевидно, что более эффективным будет начать распространение с объектов, которые имеют больше влияния.

Список литературы

1. Социальная сеть [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2015]. URL: [https://ru.wikipedia.org/wiki/Социальная_сеть_\(социология\)](https://ru.wikipedia.org/wiki/Социальная_сеть_(социология))
2. Социальные сети и социометрические исследования [электронный ресурс] // Российская ассоциация электронных библиотек: [сайт]. [2012]. URL: http://www.aselibrary.ru/press_center/journal/irr/irr3648/irr36483711/irr364837113734/irr3648371137343743/
3. Чураков А. Н. Анализ социальных сетей: СоцИс 2001. № 1. с. 109–121.
4. Social Mining [электронный ресурс] // BaseGroup Labs – технологии анализа данных: [сайт]. [2010]. URL: <https://basegroup.ru/community/articles/information-flows-social-networks>
5. Батура Т.В. Методы анализа компьютерных социальных сетей. Журнал: Вестник Новосибирского государственного университета. 2012. 26 с.
6. Основные направления исследования социальных сетей [электронный ресурс] // Учебный сайт [сайт]. [2012]. URL: http://evgavrilenko.ucoz.ru/iad/lekcija_20.pdf

УДК 004.93

ПРОГРАММНОЕ СРЕДСТВО РАСПОЗНАВАНИЯ ПЕЧАТНОГО ТЕКСТА

Е.В. Бычкова, А.А. Шумский

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
shymskiyalexander@gmail.com, elena.bichkova@gmail.com

Аннотация

Бычкова Е.В., Шумский А.А. Программное средство распознавания печатного текста. Выполнен анализ проблемы распознавания печатного текста. Проанализированы существующие популярные системы распознавания печатного текста. Приведены конкурирующие варианты реализации программного средства.

Ключевые слова: распознавание печатного текста, предварительная обработка фото, мобильное приложение.

Постановка проблемы. В наш век информационных технологий и электронных подписей обычные бумажные документы ничуть не утратили своей силы. Работникам различных сфер деятельности зачастую приходится вручную набирать объемные тексты из-за отсутствия или невозможности достать электронные исходники. Для автоматизации данного процесса существуют системы распознавания печатного текста. Данные системы бывают в виде сервисов и приложений под разные платформы (Windows, Android и другие).

На текущий момент большинство подобных приложений и сервисов платное, с закрытым исходным кодом и имеет ограничения по объему обрабатываемых данных. В связи с этим создание бесплатной системы для распознавания печатного текста с открытым исходным кодом является крайне востребованной задачей.

Цель статьи – исследование существующих проблем и способов распознавания печатного текста, анализ разрабатываемого программного средства и сравнение с конкурирующими продуктами на рынке программного обеспечения.

Метод распознавания текста. В теории распознавания образов[1] можно выделить два основных направления:

- изучение способностей к распознаванию, которыми обладают живые существа, объяснение и моделирование их;
- развитие теории и методов построения устройств, предназначенных для решения отдельных задач в прикладных целях.

Распознавание текста принадлежит ко второму направлению. В качестве методов распознавания текста выделяют:

- сравнение с шаблоном;
- распознавание по критериям;
- распознавание при помощи самообучающихся алгоритмов.

Последний из методов подразумевает использование нейронных сетей, что позволяет значительно повысить качество распознавания текста.

Работу программного средства распознавания печатного текста можно поделить на несколько этапов (см. рис. 1).



Рисунок 1 – Этапы работы программного средства распознавания печатного текста

Входными данными являются фотографии исходного текста. Для повышения точности распознавания необходимо провести обработку изображения, что позволяет избавиться от шумов. Обычно применяется медианный фильтр с последующим восстановлением изображения, а затем монохромный. Далее обработанное изображение поступает на модуль сегментации, где выделяются сгустки пикселей в потенциальные буквы, а сами сгустки объединяются в потенциальные слова. Данные буквы классифицируются с помощью нейронной сети, а затем идет поиск и сравнение итоговых слов со словарем.

Описание работы фильтров предобработки. Монохромный фильтр преобразует исходное цветное изображение в новое, содержащее свет одного цвета (длины волны), воспринимаемый как один оттенок.

Монохромный фильтр необходимо реализовать с применением метода препарирования[2], в котором граница перехода высчитывается путем нахождения среднего значения яркости всех пикселей изображения (см. рис. 2). Такой подход позволяет при сегментации выделить буквы даже при плохом освещении на цветной бумаге.

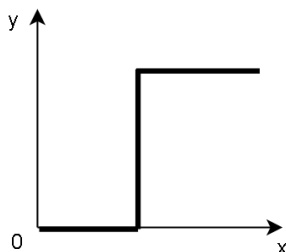


Рисунок 2 – Функция препарирования в монохромном фильтре

Медианный фильтр[3] позволяет избавиться от импульсного шума. Значения отсчетов внутри окна фильтра сортируются в порядке возрастания или убывания. Значение, находящееся в середине упорядоченного списка, поступает на выход фильтра. В случае чётного числа отсчетов в окне выходное значение фильтра равно среднему значению двух отсчетов в середине упорядоченного списка. Окно перемещается вдоль фильтруемого сигнала и вычисления повторяются. Последующее восстановление не позволяет тексту размыться, поскольку фильтр затрагивает лишь сильно измененные пиксели.

Характеристики существующих мобильных приложений по распознаванию печатного текста. Сравнение наиболее распространенных мобильных программных средств по распознаванию печатного текста с фотографий представлено в табл. 1

Описание разрабатываемой системы распознавания печатного текста. Данная система предназначена для использования на мобильных устройствах. Главная задача системы - автоматическое распознавание печатного текста и сохранение его в файл для последующей обработки пользователем.

Графический интерфейс приложения будет предоставлять следующие возможности:

Таблица 1 – Сравнение мобильных приложений по распознаванию текста

Название	Достоинства	Недостатки
FineScanner	<ol style="list-style-type: none"> 1. Высокая точность распознавания текста. 2. Поддержка различных расширений. 3. Множество фильтров и настроек сканирования. 4. Мультиязычность. 	<ol style="list-style-type: none"> 1. Функция распознавания текста платная. 2. Распознавание длится дольше, чем у конкурентов.
Image to text	<ol style="list-style-type: none"> 1. Приложение бесплатное. 2. Мультиязычность 	<ol style="list-style-type: none"> 1. Низкая точность распознавания. 2. Отсутствие настроек. 3. Низкая стабильность.
PDF Scanner +OCR	<ol style="list-style-type: none"> 1. Приложение бесплатное. 2. Поддерживает большое количество языков. 3. Большое количество настроек. 	<ol style="list-style-type: none"> 1. Низкая точность распознавания. 2. Низкая скорость распознавания. 3. Устаревший дизайн.

- загрузка изображения для распознавания из галереи устройства;
- загрузка изображения для распознавания напрямую с камеры устройства;
- сохранения результирующего текста в файл;
- загрузка результирующего файла в облако Google Drive;
- исправление ошибок распознавания программы посредством ее обучения.

Выводы. Проведено исследование существующих проблем и способов распознавания печатного текста. Осуществлен анализ разрабатываемого программного средства и сравнение с конкурирующими продуктами на рынке программного обеспечения.

Список литературы

1. Теория распознавания образов [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: www.wikipedia.org/wiki/Теория_распознавания_образов
2. Препарирование изображения [электронный ресурс] // Научная библиотека: [сайт]. [2009]. URL: http://sernam.ru/book_kir.php?id=9
3. Медианный фильтр [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: www.wikipedia.org/wiki/Медианный_фильтр

УДК 004.853

АВТОМАТИЗАЦИЯ ПРОЦЕССА ОБУЧЕНИЯ ПРИ ПОМОЩИ ВИРТУАЛЬНЫХ ЛАБОРАТОРИЙ

А.С. Столбунская, И.Д. Паламарчук, Д.М. Бочаров

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
nastassja96@mail.ru

Аннотация.

Столбунская А.С., Паламарчук И.Д., Бочаров Д.М. Автоматизация процесса обучения при помощи виртуальных лабораторий. В данной работе проведено исследование способов автоматизации процесса обучения путем использования виртуальных обучающих сред. Выполнен обзор ошибок создания виртуальных сред обучения. Определены основные функции создаваемых виртуальных лабораторий.

Ключевые слова: автоматизация обучения, виртуальная лаборатория, ошибки создания виртуальных сред, тестирование.

Постановка проблемы. Для современного обучения характерен статистический подход, больше изучать теории, чем смотреть наглядные примеры на практике. Проблема повышения эффективности и качества образования сегодня остается особо актуальной и ее решение в большей степени зависит от профессиональной компетентности учителя. На данный момент много учебных заведений частично или полностью переходят на интерактивное обучение, что дает больше знаний и меньше затрат на оборудование, а также обеспечивает безопасную среду для изучения.

Перед реализацией виртуальной машины по любому из учебных курсов необходимо выполнить следующее:

- провести обзор достоинств виртуальных обучающих сред;
- рассмотреть существующие разработки в области электронных учебных пособий с целью определения их сильных и слабых сторон;
- определить целесообразность и функции создаваемой виртуальной лаборатории.

Цель статьи – исследование способов и подходов к автоматизации процесса обучения путем разработки виртуальных обучающих сред. Выделение основных достоинств и недостатков использования таких сред и систем в учебном процессе.

Виртуальные лаборатории и их применение. Мировое образование нуждается в виртуальных системах обучения, так как содержать дорогостоящее оборудование не всегда возможно и для него требуется место.

Проще показать, как проходит эксперимент, с экранов мониторов, с заранее известным его исходом.

Виртуальная среда обучения – учебная оболочка, позволяющая преподавателю размещать материалы и настраивать активности, в которых могут принимать участие ученики. Они нужны для подготовки к реальным лабораториям. Так же это очень удобный способ демонстрации эксперимента, если отсутствуют соответствующие условия, материалы, реактивы и оборудование, а так же если опыт возможен только в виртуальной среде.

По всему миру было разработано много виртуальных сред и оболочек, которые были созданы для работы в разнообразных областях [1-3].

1. Виртуальные лаборатории STAR – программа по разработке виртуальных лабораторий для исследований и обучения. Деятельность программы заключается в разработке обучающих и исследовательских приложений по общей биологии, биохимии, генетике, гидрологии, в области распределенных вычислений.

2. Виртуальные лаборатории VirtualLab – проект по разработке виртуальных лабораторных работ для учащихся по физике, химии, биологии, экологии. Отличаются узкой специализацией, в большинстве случаев линейностью опыта: вся последовательность действий и результаты опыта заданы заранее.

3. Algodoo – программа предназначена для физических 2D симуляций. Имеет очень богатый инструментарий для создания различных объектов, механизмов и систем с целью моделирования их физического взаимодействия и свойств.

4. PhET – проект, включающий большое множество виртуальных лабораторий, демонстрирующих различные явления в области физики, биологии, химии, математики, наук о Земле.

5. Проект Wolfram Demonstrations Project – наглядная демонстрация концепций современной науки и техники. Wolfram претендует на роль единой платформы, позволяющей создать объединенный каталог онлайн-интерактивных лабораторий.

Достоинства виртуальных лабораторий. Основными преимуществами виртуальных лабораторий являются [4]:

1) отсутствие необходимости приобретения дорогостоящего оборудования и реактивов;

2) возможность моделирования процессов, протекание которых принципиально невозможно в лабораторных условиях, и наглядная визуализация их на экране компьютера;

3) безопасность использования виртуальных лабораторий в случаях, где идет работа, например, с высокими напряжениями или химическими веществами;

- 4) быстрое проведение при помощи ЭВМ серии опытов с различными значениями входных параметров;
- 5) экономия времени и ресурсов для ввода результатов в электронный формат;
- 6) использование виртуальной лаборатории в дистанционном обучении, когда в принципе отсутствует возможность работы в лабораториях университета.

Ошибки проектирования виртуальных лабораторий. Виртуальная среда является местом, где слушатели, например, дистанционного обучения проводят много времени при прохождении обучения. Ошибки, допущенные при проектировании виртуальной лаборатории, могут привести к значительному снижению качества обучения и привести к негативной реакции на обучение у слушателей.

Если анализ виртуальной среды дистанционного обучения показывает наличие одной из ниже перечисленных ошибок проектирования, это означает, что существует высокая вероятность снижения качества проводимого впоследствии дистанционного обучения [5].

1. Использование сложных средств управления виртуальной средой дистанционного обучения может привести к тому, что обучаемый будет больше времени тратить на изучение средств управления, чем

2. Плохо формализованные условия упражнений могут привести к тому, что обучаемым приходится обращаться к другим элементам виртуальной среды с целью получения дополнительной информации необходимой для выполнения упражнений.

3. Во время выполнения упражнений обучаемому должны быть доступны: текст упражнений, справочные материалы, различные элементы виртуальной среды. Необходимо свести к минимуму время, которое тратится на поиск информации, требуемой для завершения работы над упражнением.

4. Важнейшей проблемой при проведении дистанционного обучения является обеспечение идентификации слушателя, проходящего дистанционное обучение.

5. Неравномерная загрузка слушателей дистанционного обучения. Объем работ, который должен быть выполнен слушателем во время обучения, должен быть одинаковым в рамках одинаковых временных отрезков на всем протяжении обучения.

6. Обширная география слушателей дистанционного обучения может привести к возникновению трудностей при организации их общения.

Выводы. Все ошибки, допущенные на этапе проектирования виртуальных лабораторий, крайне трудно поддаются исправлению после того, как начато обучение. Следует очень тщательно подходить к проблеме

построения виртуальной лаборатории и обязательно использовать сторонний опыт, чтобы не допустить простых ошибок, которые могут повлечь возникновение серьезных проблем при проведении обучения.

Исходя из проведенного исследования, можно сделать следующие выводы:

1) использование виртуальных лабораторий в системе дистанционного обучения позволит обучаться студентам, которые ранее не могли присутствовать очно на занятиях;

2) виртуальные среды помогут закрепить навыки выполнения лабораторных работ у студентов дневного отделения, что улучшит показатели обучения;

3) лаборатории должны предоставить пользователю возможность выполнить лабораторную работу, не выходя из дома, но имея при этом постоянную возможность консультирования у преподавателя.

Кроме того, было определено, что в структуре виртуальной лаборатории должны быть реализованы и тщательно продуманы средства для:

- авторизации учащихся;
- предоставления теоретического материала или справки по изучаемым методам;
- визуализации при помощи компьютерной техники результатов выполнения заданий в лаборатории;
- тестирования полученных знаний, после чего преподаватель сможет просмотреть ответы и подтвердить оценку.

Список литературы

1. Виртуальные лаборатории. [Электронный ресурс]. – Режим доступа : <http://kpfu.ru/docs/F324157708/Virtualnye.laboratorii.pdf>

2. Технологии обучения. Виртуальные лаборатории для дистанционного обучения, вузов и школ. [Электронный ресурс]. – Режим доступа : <http://zillion.net/ru/blog/247/virtual-nyie-laboratorii-dlia-distantcionnogho-obuchienija-vuzov-i-shkol>

3. Виртуальная лаборатория VirtuЛаб. [Электронный ресурс]. – Режим доступа : <http://www.virtulab.net/>

4. А.В. Трухин. Об использовании виртуальных лабораторий в образовании // Открытое и дистанционное образование. Томск, 2002. №4 (8). С.67.

5. Ошибки проектирования виртуальной среды дистанционного обучения [Электронный ресурс]. – Режим доступа : <http://www.web-learn.ru/biblioteka-online/50-error-designing-of-the-virtual-elearning-environment>

УДК 004.89

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКОВ ПРЕДПРИЯТИЯ

Е.Д. Горлов, Д.М. Бочаров

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
zhenya.gorlov@mail.ru

Аннотация

Горлов Е.Д., Бочаров Д.М. Разработка информационной системы учета рабочего времени сотрудников предприятия. Выполнен анализ методов и подходов к разработке информационной системы учета рабочего времени сотрудников предприятия. Проведен анализ существующих разработок, определены их основные достоинства и недостатки, а также выявлены общие недоработки. Были представлены результаты многочисленных исследований, направленных на разумное использование рабочего времени.

***Ключевые слова:** информационная система, интерфейс, учет рабочего времени, методы реализации.*

Постановка проблемы. Эффективная работа персонала любой организации предполагает в первую очередь рациональное использование рабочего времени. Улучшение использования имеющегося производственного потенциала одна из ключевых задач сегодняшнего этапа социально-экономического развития нашей республики, поскольку для того чтобы осуществить перевооружение производства, заменить устаревшее оборудование прогрессивным, предприятие должно само заработать необходимые для этого средства. Значительную роль в ее решении призван сыграть экономический анализ, и, прежде всего, анализ использования рабочего времени.

Для реализации этого необходимо выполнить следующее:

- проанализировать системы учета рабочего времени;
- сделать обзор аналогичных информационных систем;
- рассмотреть планирование и распределение рабочего времени руководителем;
- проанализировать методы реализации интерфейса информационных систем.

Цель статьи – провести анализ методов и подходов к разработке информационной системы учета рабочего времени сотрудников предприятия на основе обзора аналогичных информационных систем.

Актуальность исследования. Системы учета и контроля рабочего времени предназначены для автоматической регистрации времени прихода и ухода сотрудников с работы. Информационная система ведет учет общего количества отработанных каждым сотрудником часов, больничных, отпусков, командировок, проводит учет отгулов. Система позволяет получать отчеты по всем перечисленным параметрам, составлять таблицу рабочего времени, оперативно предоставлять руководству сведения об отсутствующих сотрудниках.

Контроль мотивирует как персонал, так и начальников отделов. Оперативное реагирование на нарушение позволяет, во-первых, влиять на ситуацию, когда она еще не усугублена и не привела к последствиям. Во-вторых, руководство сможет контролировать трудовую дисциплину даже удаленно. С этой точки зрения, учет рабочего времени выступает действенным инструментом повышения эффективности бизнес-процессов [1].

Оперативное выявление нарушений позволяет наладить в коллективе трудовую дисциплину и повысить ответственность. Это также позволяет формально отследить объективную загрузку отделов и сотрудников.

Оценив преимущества работы с автоматизированными системами учета рабочего времени, работодатели уже вряд ли смогут отказаться от такого надежного помощника. Сейчас в этом направлении актуальным является совершенствование систем по удобству и доступности использования. Особенно актуальным вопрос учета рабочего времени становится в компаниях нематериального производства, где расходы на трудовые ресурсы являются основной статьей затрат.

Плюсы внедрения системы учета рабочего времени для менеджеров:

- упрощение планирования рабочего времени, так как появляются сведения об активностях по различным направлениям;
- более эффективное распределение трудозатрат на различные проекты компании;
- сокращение активностей, не несущих требуемого результата, но съедающих время;
- четкое распределение внутренней себестоимости работ между заказчиками внутри организации;
- сокращение перегрузки работников, переработок.

Как показал практический опыт эксперта, с разумными затратами учет рабочего времени позволяет довольно быстро достичь следующих целей:

- распределение рабочего времени по проектам, что позволяет вести более правильный учет затрат на владение ими;
- распределение рабочего времени по основным заказчикам;
- распределение рабочего времени по основным видам работ (совещания, коммуникации, документооборот).

При внедрении системы учета рабочего времени рекомендуется придерживаться принципов разумности и достаточности, контроля, креативности [2].

Анализ распространенных информационных систем учета рабочего времени. Проанализировав существующие аналоги подобных программ, можно обобщить, что системы контроля сотрудников делятся на три основных типа.

Первые Kickidler делают основной упор на записи действий персонала за компьютером на видео, онлайн-наблюдение и контроль нарушений. Безусловным плюсом такого подхода является тот факт, что от видео спрятаться невозможно. Минус – для хранения видео необходимо достаточное место на диске.

Другие – StaffCop, Lanagent и Стахановец – собирают максимальный объем данных о действиях пользователя (письма, файлы, сообщения) и предлагают использовать для их анализа отчеты. Плюс такого похода в том, что путем поиска по полученным и отправленным письмам, сообщениям и прочему можно найти возможные нарушения сотрудника. Минус – очень тяжело проанализировать большой объем информации.

Третья группа – системы УРВ – сохраняют минимум данных: только посещенные сайты и запущенные программы. Такого рода программы, к которым относятся CrocoTime, BitCop, WorkTime Monitor и Yaware, делят активность на продуктивную, непродуктивную, нейтральную и формируют отчеты, в которых работодатель сможет увидеть, на что тратили время его подчиненные [3]. Плюс подобных систем в их простоте и малом объеме данных для хранения и передачи. Уникальным функционалом программного комплекса является возможность наблюдать онлайн за сотрудниками всей компании одновременно. К уникальному функционалу также можно отнести и запись видео активности сотрудников. Такой комплекс возможностей вместе с гистограммой нарушений, возможностью удаленного управления и мощными отчетами по эффективности использования рабочего времени позволит решить весь комплекс задач, стоящих перед руководителями и службой безопасности.

Для очень небольших компаний, в которых нет собственного системного администратора, а уровень знаний приходящего весьма низок, где нет возможности выделить отдельный сервер, рекомендуется Mirko Employee

Monitor. Пусть набор возможностей данного продукта гораздо скромней, зато эксплуатировать его сможет человек, далекий от сферы IT.

Функционал BitCop, CrocoTime, SearchInform и Yaware практически одинаков (только Yaware позволяет делать скриншоты) и, как показывает опыт, обычно недостаточен для полноценного контроля сотрудников, а не просто составления таблиц рабочего времени. Плюсом BitCop, CrocoTime и Yaware является возможность размещения данных в облаке, то есть установка подобного рода программ крайне проста и сводится к установке только агентов [4].

Возможности LanAgent и Стахановец во многом схожи с StaffCop, однако по набору функционала и удобству использования Mirko в настоящее время выигрывает у конкурента.

Анализ методов и подходов реализации интерфейса информационной системы. Учет рабочего времени любая организация ведет, прежде всего, для контроля над соблюдением трудовой дисциплины.

Проанализировав существующие программы, прогнозируется создать программный продукт, который будет запускаться с ярлыка на рабочем столе. Далее появляется окно входа пользователей в систему. У каждого пользователя будут свои права доступа, исходя из его должности (см. рис.1).

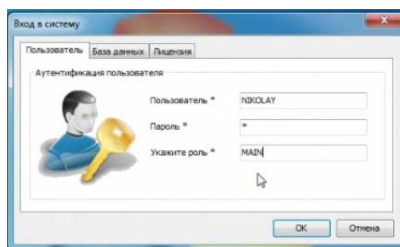


Рисунок 1 – Окно входа пользователя в систему

Для начала контроля рабочего времени необходимо заполнить в специальном справочнике список людей. Это могут быть сотрудники организации, учащиеся школы или любые другие коллективы (см. рис.2).

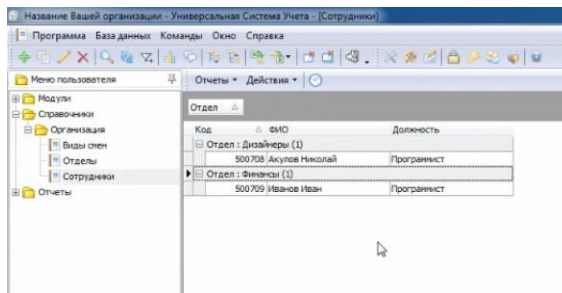


Рисунок 2 – Заполнение списка людей

Каждому человеку программа будет давать уникальный номер, его можно будет распечатать в виде бейджа со штрих кодом. Когда список имен заполнен, остается заполнить график работы или учебы, каждому пользователю можно выделить свой график посещаемости. Компьютер с программой необходимо будет установить на охране или проходной, что бы каждый мог отмечаться, когда приходит и уходит. В результате за любой период времени можно будет вывести детальную отчетность по посещаемости всего коллектива в целом и каждого в отдельности. Программа сможет построить графики и диаграммы, которые покажут, кто в какие дни и на какое время опаздывает.

Среди основных направлений учета рабочего времени персонала можно выделить:

- контроль своевременного прихода на работу, который позволяет выявить нарушителей дисциплины, опаздывающих на работу или не выходящих вовсе;
- контроль нахождения работников на рабочем месте на протяжении дня помогает выделить сотрудников, покидающих рабочее место, а также дать оценку соблюдению графика обеденного перерыва;
- контроль своевременного ухода сотрудников в конце рабочего дня помогает обнаружить подозрительные несанкционированные задержки на работе или уход раньше времени;
- учет фактически отработанного времени – помогает вести учет отпусков, больничных, производственных простоев и других моментов;
- учет прогулов.

Результаты, полученные в ходе реализации названных направлений учета, оцениваются руководством, рассматриваются уважительные причины и объяснительные сотрудников, применяются дисциплинарные наказания.

Выводы. Произведен анализ систем учета рабочего времени, выполнен обзор аналогичных информационных систем, проанализированы методы

реализации интерфейса информационных систем. Эффективная работа персонала любой организации предполагает в первую очередь рациональное использование рабочего времени. Были представлены результаты многочисленных исследований, направленных на разумное использование рабочего времени. Использование рабочего времени анализируют путем сравнения плановых и фактических данных об отработанном времени, выявляют причины отклонений, изучают динамику потерь рабочего времени.

Список литературы

1. Антосенков Е.М. Анализ фонда рабочего времени на предприятии /Е.М. Антосенков – СПб. : Питер, 2008 – 155 с.
2. Выбираем систему учета рабочего времени [Электронный ресурс]. – Режим доступа: <http://www.ualock.kiev.ua/decicions/time.htm>
3. Лучшие системы учета рабочего времени [Электронный ресурс]. – Режим доступа: <https://employee-monitoring-software.ru/>
4. Система учета реального времени, посещаемости [Электронный ресурс]. – Режим доступа: http://usu.kz/app_rabochee_vremya.php

УДК 004.021

АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ И МЕТОДОВ РАСПОЗНАВАНИЯ ДИНАМИЧЕСКИХ ЖЕСТОВ

А.А. Потопахин, К.А. Ручкин

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
alexey_potopahin@hotmail.com

Аннотация

Потопахин А.А., Ручкин К.А. Анализ программных средств и методов распознавания динамических жестов. Выполнен анализ существующих программных средств и методов распознавания динамических жестов. Также проанализированы необходимые для распознавания технические средства. Выделены преимущества и недостатки библиотек, используемые алгоритмы и необходимые для работы устройства ввода.

Ключевые слова: динамические жесты, методы распознавания жестов, технические средства для распознавания жестов.

Введение. В настоящее время существующие методы распознавания динамических жестов человека основываются на современных математических, эволюционных алгоритмах и статических методах (Скрытые Марковские Модели), которые могут быть использованы в бесконтактном взаимодействии пользователя с компьютером.

Для «захвата» жестов человека могут использоваться различные устройства, например, ультразвуковые локаторы, кинематические датчики, системы структурированной подсветки и т.д. Но наиболее распространенным для получения данных о жестах пользователя является видеокамера и устройства подобные трехмерному сенсору Kinect от Microsoft, который помимо цветной видеокамеры, также состоит из двух сенсоров глубины. Применение двух и более видео-датчиков позволяет запечатлеть дополнительную информацию о трехмерном объекте (отверстия, невидимые с другой камеры форма боковых проекций, и др.) [1].

Постановка задачи. В данной работе рассматриваются и анализируются существующие методы и программные средства для распознавания жестов человека. Выделяются преимущества и недостатки подходов, а также выбираются лучшие по результатам анализа.

Методы распознавания динамических жестов. Особенностью методов, основанных на анализе внешних признаков жеста, является анализ только внешнего вида (формы, позиции и т.д.) целевого объекта. Для распознавания не хранится никакой информации о физических свойствах рассматриваемого объекта. Рассмотрим известные работы и методы, посвященные распознаванию жестов руки человека на основе анализа внешних признаков жеста.

Наиболее распространенным является подход с использованием Скрытых Моделей Маркова (СММ). Методы распознавания жестов на основе СММ представляют каждый жест набором состояний, которые связаны с тремя вероятностями (начальное, переходное, наблюдение) вычисляемые из заранее подготовленных эталонов. СММ выбирают модель с наилучшей вероятностью и классифицируют жест к соответствующей категории. Хотя системы распознавания на основе СММ выбирают модель с наилучшей вероятностью, но не гарантируется, что шаблон действительно похож на опорный жест. Для того, чтобы получить хорошие результаты, СММ должна быть хорошо обучена и настроена. Основными недостатками этого подхода является то, что для калибровки моделей требуется большое количество образцов и длительное время обучения.

Распознавание жестов в режиме реального времени остается сложной задачей из-за неопределенности в методах определения границ жестов. Подход, основанный на движении роя частиц (Particle Swarm Movement) решает эту проблему, упрощая сегментацию жестов [2]. В предлагаемом подходе избегается преждевременная идентификация жеста, тем самым увеличивая точность результата распознавания. Данный распознаватель требует меньше вычислительных и временных ресурсов по сравнению с СММ и является хорошим кандидатом для реализации в приложениях, выполняющие распознавание в режиме реального времени.

Программные средства для распознавания динамических жестов. Рассмотрим наиболее популярные прикладные программные средства для распознавания.

Gesture Recognition Toolkit (GRT) представляет собой кроссплатформенную библиотеку, которая реализована на C++ с открытым исходным кодом, разработанную для того, чтобы выполнять распознавание жестов. Акцент сделан на простоту использования, в то же время поддерживается гибкость и настраиваемость. Инструментарий обладает широким спектром алгоритмов классификаций и регрессии, имеет обширную поддержку для построения систем реального времени. Библиотека включает в себя алгоритмы обработки сигналов, извлечения признаков и автоматическое распознавание жестов [3].

XKin – платформа с открытым исходным кодом для Kinect, обеспечивающая более естественную и интуитивно понятную коммуникацию между человеком и компьютером. Программный пакет состоит из полезных инструментов для обучения системы для работы с жестами. Проект XKin полностью реализован на C с открытым исходным кодом и позволяет выполнять распознавание в режиме реального времени как статические позиции рук, так и динамические жесты [4].

HandVu – это программный комплекс для распознавания жестов и управление графическим пользовательским интерфейсом на основе компьютерного зрения. При использовании практически любой цветной камеры и достаточной вычислительной мощности это программное обеспечение может предоставить возможность управлять компьютерным интерфейсом. HandVu обнаруживает руку, а затем отслеживает ее и распознает ключевые позиции – все в реальном времени и без необходимости калибровки камеры. Результаты распознавания жестов доступны по технологии клиент-сервер в настраиваемом формате и в виде пакетов OSC [5].

Сравнение программных средств. Сравнение приведенных и наиболее популярных программных средств представлено в табл. 1. Эта таблица показывает основные реализованные алгоритмы, которые используют библиотеки, а также внешние зависимости и устройства ввода.

Таблица 1 – Сравнение программных средств

Библиотека	Gesture Recognition Toolkit	Dlib-ml	Java-ML	XKin	HandVu
Алгоритмы	Adaptive Boosting (бустинг), Дерево принятия решений, Алгоритм динамической трансформации временной шкалы (DTW), Скрытые Модели Маркова (CMM)	Сеть радиально-базисных функций	Алгоритмы кластеризации	CMM	Camshift
Зависимость	–	–	–	OpenCV	OpenCV
Язык реализации	C++	C++	Java	C	C++
Устройство ввода	Видеокамера	Видеокамера	Видеокамера	Kinect	Видеокамера

Библиотеки, реализованные на С и С++ менее затратны по вычислительной сложности, чем библиотеки на Java (Java-ML) или Python, однако являются сложными для настройки и требуют дополнительную калибровку устройств ввода. Однако главным преимуществом является работа в режиме реального времени и простые, недорогие технические средства.

Выводы. В работе рассмотрены и проанализированы существующие методы и программные средства для распознавания жестов человека. Также выделены преимущества и недостатки методов.

Главными недостатками существующих методов являются чувствительность к изменениям освещения, потребность обучения системы для каждого оператора, невысокое качество распознавания жестов и небольшая скорость распознавания.

Для распознавания сложных жестов рук необходимо использовать специализированные устройства ввода, а также проводить настройку и обучение программных средств. Но для динамических жестов, где важно только знать траекторию движения руки (влево, вправо, вверх, вниз) достаточно использовать не сложные математические алгоритмы.

Таким образом, актуальной является задача создания новых моделей, методов и алгоритмов распознавания жестов руки, которые могут быть использованы для создания систем бесконтактного человеко-машинного взаимодействия.

Список литературы

1. А.В. Терехин. Диссертация. «Алгоритмы определения безразмерных признаков изображений проекций трехмерных объектов и их распознавание». МИ (филиал) ВлГУ, 2015г. – 187с.
2. Nyirarugira C., Choi H. Hand Gesture Recognition Using Particle Swarm Movement. Mathematical Problems in Engineering. Volume 2016.
3. Gillian N., A. Paradiso J. The Gesture Recognition Toolkit. Journal of Machine Learning Research 15 (2014).
4. Pedersoli F., Adami N. XKin: Extendable hand pose and gesture recognition library for Kinect. Proceedings of the 20th ACM international conference on Multimedia.
5. Kölsch M. Vision Based Hand Gesture Interfaces for Wearable Computing and Virtual Environments. Ph. D. Dissertation, August 2004.

УДК 004.832.28

ФУНКЦИИ И СТРУКТУРА КЛАССОВ В МОДЕЛИРОВАНИИ ИСКУССТВЕННОЙ ЖИЗНИ

Нечаев А.В., Харитонов Ю.Е.

ГОУВПО ДонНТУ

кафедра экономической кибернетики

E-mail: holdem-10@bk.ru

Аннотация

Нечаев А.В., Харитонов Ю.Е. Структура и функциональность классов для моделирования искусственной жизни. Рассмотрена практика моделирования экономических систем. Определены классы необходимые для моделирования.

Общая постановка проблемы

В последнее время моделирование экономических систем крайне затруднительно по причине стихийности современной экономики, как локальной (ДНР), так и глобальной. Из-за стихийности многих факторов экономики традиционные способы моделирования утратили свой потенциал в данной практической области.

Исследования

Искусственная жизнь – это понятие, введенное Крисом Лангтоном для обозначения множества компьютерных механизмов, которые используются для моделирования естественных систем. Искусственная жизнь применяется для моделирования процессов в экономике, поведения животных и насекомых, а также взаимодействия различных объектов.

Искусственная жизнь может быть описана как теория и практика моделирования биологических и экономических систем. С помощью моделей можно управлять созданной средой, проверять различные гипотезы и наблюдать, как системы и среда реагируют на изменения [1].

Агентные системы (алгоритмы) имеют различия в двух аспектах: с позиций анализа свойств и поведения агентов в процессе функционирования в целом; с позиций изучения и конструирования свойств агента, определяющих его внутренние процессы (получение знаний, выработка целей, принятие решений и т.д.).

Можно считать устоявшейся традицию выделения трех базовых классов архитектур агентных систем и соответствующих им моделей интеллектуальных агентов: делиберативные архитектуры и модели, содержащие точную символическую модель мира и принимающих решения на основе логического вывода; реактивные архитектуры и модели, использующие множество простых сценариев поведения агентов в рамках установленных

реакций на определенные события окружающей среды; гибридные архитектуры и модели, использующие ситуативный анализ всех возможных активностей агентов.

В данной статье используется гибридный агентный алгоритм в плане искусственной жизни [2].

В эволюционной экономике вводится ключевое понятие относительно стабильных в краткосрочной и среднесрочной перспективе организационных рутин – используемых фирмами стандартных правил и процедур ведения деятельности, фиксирующих методы получения информации, выработки и проведения в жизнь стратегических и тактических решений, распределения задач между исполнителями. Использование рутинных процедур позволяет экономить ресурсы, необходимые для принятия решений в многократно повторяемых ситуациях, хотя порой оно приводит к тому, что принятые решения оказываются неоптимальными. Это подчеркивает важность фактора случайных ошибок, составляющих неотъемлемую часть как процесса накопления опыта хозяйственной деятельности, так и самого процесса эволюции.

Процесс изменений в экономической системе моделируется в терминах отбора агентов, наиболее адаптированных – с точки зрения применяемых организационных рутин – к особенностям внешней среды. При этом предполагается, что повышающие уровень адаптированности рутины могут восприниматься агентами путем имитации или подражания образцам, а изменение условий внешней среды обуславливает необходимость экспериментального изменения рутин, ведущего в процессе отбора к выработке нового комплекса «успешных» рутин.

Характерной особенностью интерпретации данного эволюционного процесса в рамках рассматриваемого алгоритма является представление о наследовании приобретенных признаков, например, воспроизведении рутин, воспринятых методом имитации.

Накопление поведенческих изменений, увеличивающих адаптированность индивидуального агента, может изменить саму внешнюю среду, т. е. привести в конечном итоге к изменению критериев адаптированности.

В частности, осуществление одним из агентов базовой инновации, ведущей к созданию прибыльной рыночной ниши, может побудить других агентов осуществлять имитационные инновации, ведущие к падению нормы прибыли и «замораживанию» в данной производственной сфере чрезмерного объема экономических ресурсов, которые могли бы найти более производительное применение в других отраслях экономики.

Моделируется отбор двух типов организационных рутин: с одной стороны, рутин, управляющих использованием фирмами ресурсов при заданном продуктивном наборе и заданной технологии, и с другой стороны

рутин более высокого уровня, на которые опирается процесс создания новых продуктов и разработки новых производственных технологий.

В ответ на изменение ситуации агент может начать изменения внутри себя, включая поиск новых организационных, технологических и прочих приемов и методов, а может предложить новый продукт, найти новые рынки и новых потребителей своей продукции и т.д. Выбор типа реакции зависит от оценки транзакционных издержек в условиях несовершенства информации и неопределенности. И эти издержки в данном случае включают, кроме всего прочего, издержки поиска нужного партнера, переговоров, обучения партнера и т.д. Если речь идет о нововведении, то часто оказывается, что на рынке нет «частей», нужных фирме, чтобы из них, пусть и в принципиально новой комбинации, «сложить» новый продукт или технологию. Динамические транзакционные издержки оказываются очень большими. Внедрение нового знания наиболее экономным способом зависит от природы нового знания и от существующей на рынке конфигурации возможностей.

Проектирование, разработка, производство и маркетинг нового продукта не идентичны изобретательской деятельности и, более того, не обязательно осуществляются в рамках одного и того же агента. Такая дифференциация принципиально важна, хотя изобретательство и инноваторство часто взаимодействуют, и процесс внедрения инноваций служит катализатором последующих изобретений.

Целью каждого агента в модели является не максимизация прибыли, а максимизация выпуска при имеющейся технологии и располагаемых ресурсах [3].

Наиболее важной особенностью этого алгоритма является то, что каждый моделируемый объект состоит из множества распределенных звеньев, действующих параллельно без глобального контроля, ответственного за поведение этих звеньев. Более того, действия каждого звена зависят от состояния и действий некоторого ограниченного числа других звеньев, и результирующее состояние системы определяется конкуренцией и координацией среди звеньев, подчиненных структурным ограничениям. Сложность системы, таким образом, увеличивается в большей степени от взаимодействий среди звеньев, чем от увеличения сложности самих индивидуальных звеньев. Более того, локальная сеть взаимосвязей, соединяющая отдельные звенья друг с другом, постоянно перекомбинируется и пересматривается. В частности, постоянно создаются ниши, которые могут быть использованы через адаптацию, а их использование, в свою очередь, приводит к появлению новых ниш, так что существует постоянное обновление [4].

Заимствования и аналогии из синтетической науки поведения в сфере искусственной жизни наблюдаются в терминах схожести и различия с экономической эволюцией в плане алгоритма искусственной жизни [5].

Основные функции искусственной жизни в сфере экономики описаны в диаграмме классов, позволяющей увидеть взаимосвязи самих классов (рис.1).



Рисунок 1. Диаграмма классов

В диаграмме классов показаны структура и функциональность классов, необходимые для моделирования искусственной жизни. В процессе реализации данного алгоритма на каком-либо языке объектно-ориентированного программирования эта диаграмма будет расширяться по мере увеличения детализации моделирования жизненных циклов.

Выводы. Таким образом искусственную жизнь можно использовать в целях изучения биологических, а также экономических и производственных. Так, смоделировав жизненный цикл группы предприятий-агентов, работающих в одной производственной сфере, можно определить оптимальное решение в тех или иных ситуациях, условия которых исходят из внешней среды.

Список литературы

1. CALResCo. Концепция исследования искусственной жизни для самоорганизующихся систем (The Complexity & Artificial Life Research Concept for Self_Organizing Systems). (<http://www.calresco.org>)
2. Digital Life Lab at Caltech. Программное обеспечение для Aveda (Aveda Software). (<http://dllib.caltech.edu/aveda>)
3. Лангтон К. Что такое искусственная жизнь (Langton C. What Is Artificial Life). (<http://www.biota.org/papers/cgalife.html>)
4. МакЛеннан Б. Домашняя страница Брюса МакЛеннана (MacLennan B. Bruce MacLennan's Home Page). (<http://www.cs.utk.edu/~mclennan/>)
5. МакЛеннан Б. Искусственная жизнь и синтетическая теория поведения (MacLennan B. Artificial Life and Synthetic Ethology). (<http://www.cs.utk.edu/~mclennan/alife.html>)

УДК 004.8

ПЕРСОНАЛИЗИРОВАННАЯ ВЫДАЧА РЕКОМЕНДАЦИЙ НА ОСНОВЕ ТЕОРИИ АДАПТИВНОГО РЕЗОНАНСА

Михайлович Ф., Харитонов Ю.Е.

ГОУВПО Донецкий национальный технический университет
кафедра экономической кибернетики
E-mail: fmihajlovic@mail.ru

Аннотация

Михайлович Ф., Харитонов Ю.Е. Применение теории адаптивного резонанса в системах выдачи рекомендаций. Рассмотрена система выдачи рекомендаций в поисковых системах и предложен алгоритм выдачи рекомендаций с помощью метода теории адаптивного резонанса.

Ключевые слова: *теория адаптивного резонанса, кластеризация, персонализация.*

Современные поисковые системы преследуют цель дать правильный ответ на заданный запрос с учетом предпочтений и интересов пользователей. Одним из последних нововведений в алгоритмы поисковых систем стала персонализированная выдача, т.е. выдача результатов, которые учитывают предпочтения конкретного пользователя и показываются только ему.

Данная статья посвящена знаменитому алгоритму Гроссберга и Карпендера, ART1, который был первым в семье алгоритмов теории адаптивного резонанса.

Алгоритм ART1 концептуально прост и легок в реализации. Более ранние алгоритмы, такие как алгоритм кластеризации McQueen, хотя и были проще, но имели ряд существенных недостатков. Например, они не позволяли создавать новые кластеры (кластеры задавались в начале работы алгоритма). Кроме того, в ранних алгоритмах не было параметра, позволяющего изменять размеры класса для кластеров.

Недостаток всех алгоритмов (ранних и ART1) заключается в том, что конечный набор кластеров (и векторов прототипов) может изменяться в зависимости от порядка, в котором проводилось обучение.

Было создано множество версий алгоритма ART1 как с целью усовершенствования, так и для решения различных проблем. Алгоритм ART1 работает с дискретными данными, а алгоритм ART2 позволяет классифицировать непрерывный поток данных (например, временные диаграммы).

ARTMAP – это измененный алгоритм ART, который может изучать изменяющиеся двоичные схемы. Он представляет собой синтез ART и нечеткой логики. Существуют и другие алгоритмы из семейства ART.

Практическое применение и работа алгоритма ART1 будет продемонстрирована на примере персонализации (данная проблема также известна как система выдачи рекомендаций).

Теория адаптивного резонанса неотъемлемо связана с алгоритмом кластеризации. Clustering algorithm (Алгоритм кластеризации) – это метод, благодаря которому данные разделяются и объединяются в небольшие группы (кластеры) по принципу аналогии. По тому же принципу осуществляется отделение несхожих данных, поэтому главной задачей при разбивке данных на кластеры является классификация. Хотя классификация используется во многих случаях, ее основное предназначение – изучение данных в кластерах для выявления различий между ними. Более специфическое использование кластеров будет рассмотрено далее.

Алгоритм ART1 работает с объектами, которые называются Feature vector (векторами признаков). Вектор признаков является группой значений в двоичном коде, которые представляют определенный тип информации. Примером вектора признаков может служить выбор покупок. Каждый объект вектора признаков показывает, приобрел ли покупатель товар (если да, то значение равно 1, если нет – 0). Этот вектор признаков описывает покупательную способность путем идентификации приобретенных покупателем предметов (о которых мы имеем информацию). Собираются векторы признаков покупателя, к которым затем применяется алгоритм ART1, чтобы разделить данные на кластеры. Идея состоит в том, что группа схожих данных о покупателе (содержащаяся в кластере) будет сообщать интересную информацию о схожих параметрах для группы покупателей.

Таким образом, алгоритм проходит через все векторы признаков и сравнивает их со всеми векторами прототипами. Хотя все векторы уже размещены по кластерам, проверка необходима. Она позволяет убедиться в том, что векторы расположены в нужных кластерах. Дело в том, что последующие тесты векторов признаков могли создать новые кластеры, поэтому необходимо выполнить дополнительную проверку и удостовериться, что векторы не нужно перемещать в другие кластеры.

После проверки всех векторов признаков, которая не потребовала дополнительных изменений, процесс формирования кластеров можно считать завершенным. Чтобы избежать перемещения вектора признаков между двумя векторами прототипами, алгоритм выполняет несколько итераций, чтобы объединить кластеры. Количество итераций должно быть достаточно большим, чтобы избежать преждевременного слияния.

Идея персонализации не нова, многие компании занимаются ей в течение определенного времени. Повышением своей популярности персонализация

обязана, в первую очередь, преимуществами ее использования в Глобальной сети. Internet-магазины позволяют осуществлять персонализацию практически в режиме реального времени.

Перед тем как покупатель делает заказ, сайт может рекомендовать другие товары, которые могут больше подойти посетителю. Фактор времени для продавца очень важен, поскольку до покупки он может воздействовать на посетителя таким образом, чтобы изменить его решение в сторону увеличения своей прибыли.

Персонализация включает ввод определенной информации и вывод рекомендаций, предназначенных для пользователя, на основании некоторых расчетов. Метод осуществления персонализации различается в зависимости от типа данных на входе, требований к рекомендациям на выходе и в некоторых случаях от скорости и точности выполнения расчета.

Многие компании пользуются разнообразными способами персонализации. Большинство алгоритмов (или методов настройки существующих алгоритмов) хранятся в секрете, так как являются стратегически важными для владельцев. Amazon.com использует метод, который называется «Фильтрация на основе сотрудничества». Этот метод выдает рекомендации на основании сходства между покупками данного покупателя и других покупателей. Для выделения подгруппы покупателей из общей массы используется мера схожести. Получив подгруппу покупателей, система выдает рекомендации в зависимости от различий между членами подгруппы.

Персонализация с использованием алгоритма ART1 состоит из двух этапов. Сначала выполняется стандартный алгоритм ART1 для векторов признаков (данных о покупателях). Далее, чтобы получить рекомендацию, анализируется вектор признаков (отображающий покупателя, которому нужно дать рекомендацию), а также новый элемент, так называемый вектор суммирования (Sum vector). Вектор суммирования, который не входит собственно в алгоритм ART1, представляет собой сумму столбцов векторов признаков в кластере.

Рассмотрим процесс выдачи рекомендации на примере. Предположим, что покупатель, которому мы должны дать рекомендацию, представлен вектором признаков u , входящим в кластер A . Содержимое вектора признаков находится в так называемой (истории покупок клиента). Сначала по вектору суммирования необходимо определить, какие товары (столбцы) представлены в кластере (то есть не равны 0). Затем алгоритм находит самое большое значение в векторе суммирования, которое соответствует объекту в векторе признаков покупателя со значением 0. Оно представляет товар, не приобретенный покупателем, но популярный для кластера. Такая информация является основой для рекомендации.

Предположение (или статистическое предвидение) заключается в следующем: 66% покупателей в кластере уже приобрели, этот товар значит, высока вероятность того, что данный клиент тоже пожелает его купить.

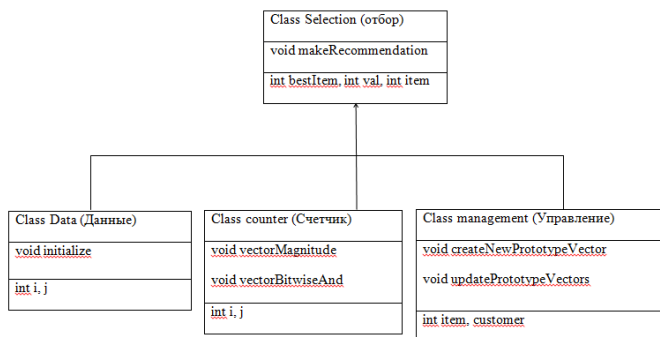


Рис. 1. Диаграмма классов

Алгоритм ART1 предоставляет возможность классификации данных в отдельные сегменты (кластеры). Классификация может быть полезна как средство исследования классов (типов) кластеров. Кроме того, как видно по алгоритму персонализации, изучение членов отдельного кластера позволяет получить множество информации. При персонализации в Internet данные могут включать не только отображение содержания Web страницы, но и время, которое было потрачено на ее просмотр. Тип и отображение данных зависят от алгоритма, который выполняет персонализацию. При правильной кодировке в векторах признаков алгоритм ART1 может работать с широким диапазоном данных, отображающим многие аспекты поведения покупателя в сети Internet. Именно этот алгоритм наиболее подходит для реализации выдачи рекомендаций.

Таким образом, алгоритм является очень универсальным, так как может использоваться в различных отраслях информационного бизнеса и упрощать действия клиента предполагаемого сервиса.

Список литературы

1. Wolfram Research. Мир математики Эрика Вайштайна (Eric Weisstein's World of Mathematics). Доступно по адресу <http://mathworld.wolfram.com/>.
2. Галлант С. Обучение в нейронных сетях (Gallant S. Neural Network Learning. – Cambridge, Mass.: MIT Press, 1994).
3. Карпентер Д., Гроссберг С. Массивная параллельная архитектура для само стоятельной машины, распознающей нейронные модели (Carpenter G., Grossberg S. A Massively Parallel Architecture for a SelfOrganizing Neural Pattern Recognition Machine // Computer Vision, Graphics and Image Processing, 37: 54–115, 1987).

УДК 004.932

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОСМОТРА И РЕДАКТИРОВАНИЯ ГРАФИЧЕСКИХ ФАЙЛОВ

Семенова А.П., Кудрявых А.Ю.

Донецкий национальный технический университет

E-mail: kornext@gmail.com

Аннотация

Семенова А.П., Кудрявых А.Ю. Разработка программного обеспечения для просмотра и редактирования графических файлов. Рассмотрены основные принципы построения программного обеспечения для просмотра и редактирования графических файлов, проведен анализ структуры графического файла. Описаны особенности построения и обработки изображения, цветовые модели. Реализован графический редактор и использованием различных алгоритмов растровой графики.

Ключевые слова: *графический редактор, цветовая модель, растровая графика, графический файл.*

Общая постановка проблемы

Благодаря развитию информационных технологий практически все источники информации перешли от физических носителей в электронный вид. Не стало исключением и изображение. Еще в прошлом веке появились первые цифровые фотоаппараты и видеокамеры, а с помощью возрастающих возможностями вычислительной техники стало возможным и их обработка.

Однако вместе с внедрением информационных технологий во все аспекты жизнедеятельности человека возросли и потребности в более тщательной и детальной обработке изображения. Появились новые узкоспециализированные форматы графических файлов, росло качество отображения, совершенствовались и усложнялись алгоритмы обработки для получения наилучшего результата.

Огромное количество информации стало оцифровываться и представляться в электронном виде отчего возникла необходимость в специализированном программном обеспечении для отображения. И не смотря на изобилие подобных программных продуктов неизменно выходят новые и улучшаются старые с применением новых технологий отображения и обработки, с поддержкой более новых графических форматов и цветовых моделей.

Цель проекта – изучение особенностей современного цифрового изображения, исследование и анализ существующих способов преобразования и обработки графических файлов, а также разработка программного обеспечения для просмотра и редактирования графических файлов.

Постановка задачи

Объектом разработки является программное обеспечение для просмотра и редактирования графических файлов.

Программное обеспечение будет складываться из двух больших модулей, объединённых в одно приложение и предоставлять пользователю возможность просматривать графические файлы и, при необходимости, их редактировать. Редактирование будет включать все в себя все стандартные средства графического редактора, а также необходимый функционал для первичной обработки изображения. В приложении будут реализованы свыше трех десятков различных алгоритмов обработки и отображения изображения для предоставления максимальной функциональности пользователю.

Исследования

Программное обеспечение необходимо в первую очередь для простой обработки графических файлов при их непосредственном просмотре, без необходимости использовать различные сторонние программные продукты. В ходе проведения подготовительной работы и непосредственно разработки программного обеспечения были исследованы следующие аспекты:

- особенности построения цифрового изображения[1];
- классификация и типы цифровых изображений[2];
- алгоритмы масштабирования и изменения исходных размеров[1];
- общие алгоритмы обработки цифрового изображения[1];
- особенности преобразования цифровых изображений[3];
- цветовые модели[4].

В процессе анализа и исследования предметной области было выявлено, что огромную роль в обработке и отображении цифрового изображения выполняет тип графического файла и цветовая модель, с помощью которой он представляется. Каждый вид и тип представления графической информации обладает своими особенностями, преимуществами и недостатками что сильно сказывается на их обработке.

Кроме того важную роль выполняют и алгоритмы обработки изображения. Различные алгоритмы подходят для различных ситуаций, некоторые выполняют одну и ту же функцию, но с совершенно разным результатом, некоторые не работают с определенными форматами файлов или цветовыми моделями, а некоторые выполняются чрезвычайно медленно.

Все эти аспекты обработки необходимо учитывать при создании программного обеспечения. Приложение должно работать со всеми заявленными типами графических файлов и цветовыми моделями,

преобразовывать форматы данных и обрабатывать изображения без ошибок. Для этого каждый алгоритм обработки был проанализирован и исследован на совместимость и корректность выполнения с различными входными данными. Впоследствии, на основании полученных результатов были определены наилучшие алгоритмы обработки изображения для их дальнейшей реализации.

Программная реализация

Программное обеспечение будет предоставлять пользователю весь необходимый функционал для работы с графическими файлами. Программный продукт будет представлен в виде нативного приложения под операционные системы семейства Windows с использованием NETFramework версии не ниже 3.5 и состоять из следующих модулей:

- модуль перестройки размеров формы и ее элементов при изменении;
- модуль сохранения изображения и преобразования графических форматов;
- модуль масштабирования;
- модуль первичной обработки изображения;
- модуль фильтров, изменения яркости и контрастности;
- модуль вторичной обработки изображения.
- модуль добавления текста;
- модуль функциональных клавиш.

Приложение строится таким образом, что бы на любой стадии разработки можно было интегрировать новый функционал посредством других модулей или подмодулей, не приводя при этом к нестабильной работе всего приложения и всех его частей. Программная реализация выполнена в среде языка программирования C# и состоит из двух главных форм и нескольких дополнительных. На рисунке 1 показана стартовая форма приложения.

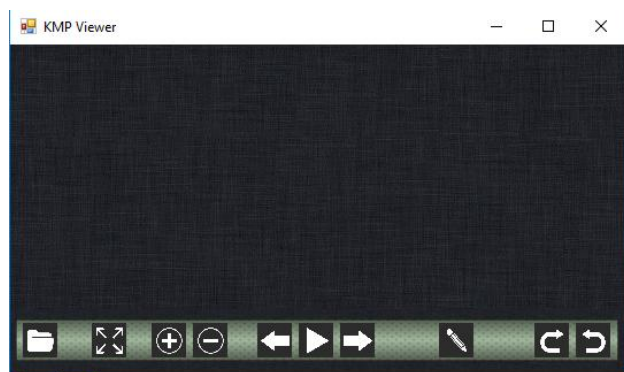


Рис. 1. Общая структура вызовов элементов программы и их взаимодействия

В стартовой форме размещены элементы интерфейса приложения для просмотра графических файлов, при переходе в режим редактирования вызывается форма графического редактора. Все модули и второстепенные формы связаны с двумя главными модулями и вызываются через их интерфейс. На рисунке 2 показана общая структура вызовов элементов программы и их взаимодействия.

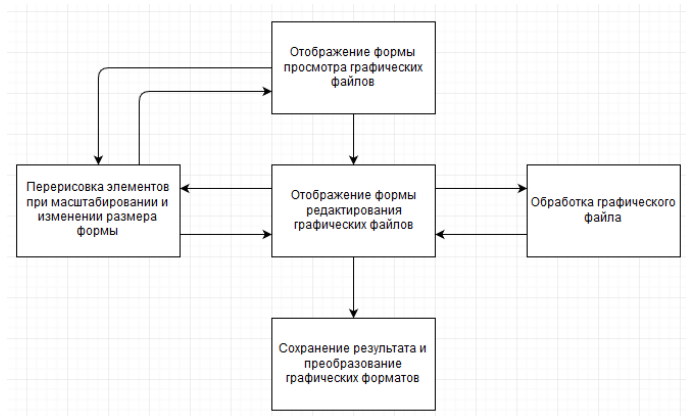


Рис. 2. Общая структура вызовов элементов программы и их взаимодействия

Выводы

В ходе исследования проблем современной обработки цифровых файлов были определены цели, достигаемые при разработке и использовании программного обеспечения, проведен анализ средств и возможностей современных графических форматов, их обработка и выполнено сравнение различных алгоритмов работы с растровой графикой. Также, были выделены основные модули, построена общая схема взаимодействия.

Список литературы

1. Херн Д. Г. «Компьютерная графика и ее обработка. Стандарт OpenGL» – М.: Лори, 2002 – 576с.
2. Порев В. С. «Компьютерная графика» – М.: Вильямс, 2008 – 1168 с.
3. Паринин Т. Н. «Цифровая обработка изображения» – М.: Наука, 2007 – 539 с.
4. Шашлов Б. А. «Теория цвета. Компьютерные цветовые модели» – СПб.: Питер, 2010 – 992 с.

УДК 004.023

**АЛГОРИТМЫ АНАЛИЗА СОЦИАЛЬНЫХ СЕТЕЙ ДЛЯ ВЫЯВЛЕНИЯ
СОЦИАЛЬНЫХ ХАРАКТЕРИСТИК ПОЛЬЗОВАТЕЛЕЙ****Лысюк Р.Р., Кравец Т.Н.**

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
radik93@outlook.com

Аннотация

Лысюк Р.Р., Кравец Т.Н. Алгоритмы анализа социальных сетей для выявления социальных характеристик пользователей. Все больше компаний начинают исследовать и разрабатывать собственные методы, для того, чтобы смоделировать поведение пользователей в различных ситуациях. Такие компании, как IBM, Microsoft, Google предлагают различные инструменты для анализа данных, чтобы помочь B2B компаниям лучше узнать своих потенциальных клиентов. Благодаря колоссальному объему информации оставленным пользователями – это дает возможность исследовать эту информацию и взаимосвязь между ней, открывает беспрецедентные возможности для решения исследовательских и бизнес-задач (многие из которых до этого невозможно было решать эффективно из-за недостатка данных), а также создания огромных вспомогательных сервисов и приложений для пользователей социальных сетей. В данной статье рассмотрены методы сбора и анализа доступных публичных данных пользователей социальной сетей для выявления характеристик пользователей.

Ключевые слова: социальная сеть, классификация текста, характеристики пользователей, нейронные сети, анализ социальных сетей.

Введение. Основными задачами информационно-аналитической поддержки пассивной и активной работы с социальными сетями являются: мониторинг и анализ социальных сетей (для достижения понимания происходящих в социальных сетях процессов), прогнозирование и управление (для перевода социальной сети в требуемое состояние) [1].

Мониторинг включает получение и структурирование первичных данных. Собираются тексты сообщений, связи между пользователями, ссылки на внешние ресурсы. Возможности системы во многом определяются богатством используемых данных и режимом их обработки. Системы, поддерживающие мониторинг в режиме реального времени, сложнее в реализации, чем использующие ретроспективный сбор данных [2].

Анализ подразумевает несколько этапов обработки первичных данных.

Во-первых, вычисление базовых показателей, которое позволяет отвечать на простые количественные вопросы типа «сколько сообщений написал пользователь А?». Далее выявление статистических и структурных закономерностей в данных дает понимание природы исследуемой сети.

Постановка задачи. Целью статьи является рассмотрение методов системы, которая будет собирать публичную информацию из социальной сети Вконтакте и анализировать ее для определения демографических данных пользователей и их политическую активность.

Основой для работы стали исследования, проведенные в области социальных сетей и существующих похожих системах. На данный момент существует множество систем анализа данных социальной сети Вконтакте, но зачастую возможности этих систем недостаточны для пользователей, так как они имеют ограничения на анализ данных либо доступны только государственным органам. Разработка системы для дипломной работы заключается в сборе и анализе данных пользователей для выявления правдивости данных представленных пользователями, а также для получения характеристик пользователей..

Входными данными приложения будут сообщения пользователей, комментарии к сообщениям, сообщения групп, комментарии к сообщениям в группах, профиль пользователей ВК, медиафайлы пользователей ВК.

Выходными данными являются выводы полученные при анализе данных и могут включать следующие сведения:

- распределение времени посещения сети (недельный, месячный, суточный циклы) на основе собранной базы временных меток посещений социальной сети конкретным пользователем;

- перечень интересов и уровень активности, (3 уровня активности: публикации у себя на стене, публикации в группах, «лайки» в группах и у друзей);

- индекс достоверности пользователя на основе проанализированных данных друзей;

- личностные характеристики пользователя, такие как – взгляды, личные планы на будущее, стиль текста, полученные на основе публикуемых сообщений на стене или в группах, включая «лайки», «репосты», медиафайлы.

Структурные свойства соиальных сетей. В настоящее время статические структурные свойства крупномасштабных саморазвивающихся сетей, таких как Интернет, сети цитирования ученых, онлайнные социальные сети, хорошо изучены. Например, степенной закон распределения, «тесный мир» (small world), т.е. малое среднее расстояние между узлами. Эти свойства наблюдались первоначально для социальных сетей еще до появления

Интернета. Затем было установлено, что структура Интернета обладает такими же характеристиками.

В 2011 году исследование сети Facebook показало, что ее топология имеет сходные свойства, но, например, распределение числа друзей описывается законом, отличным от степенного.

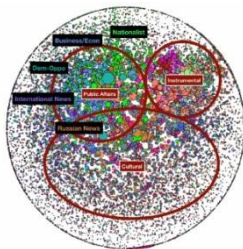


Рис. 1 – Карта политических предпочтений в русскоязычной блогосфере.

Рассмотрим русскоязычную блогосферу. Точки обозначают блогеров. Размер точки показывает популярность блогера; цветом обозначен кластер, к которому относится блогер [3].

На рис. 1 показана карта разбиения блогосферы на кластеры, группы блогеров, которые цитируют похожие ресурсы. Для визуализации использовался алгоритм Fruchterman–Reingold, имитирующий «физическое взаимодействие» блогеров. На всех блогеров действует одна общая сила, «сдувающая» их к краям карты. В то же время каждые два блога «притягиваются» друг к другу с силой, пропорциональной числу ссылок между ними. Получается, что ближе друг к другу расположены блогеры, активно цитирующие друг друга. Видно, что авторитетные блогеры, принадлежащие к разным кластерам, часто располагаются рядом. Нет тенденции к образованию замкнутых идеологически однородных сообществ, как в американской блогосфере.

В данном исследовании основное внимание было уделено шести кластерам в зоне политических и общественных дискуссий: 1) международные темы, 2) обсуждение российских новостей, 3) националисты, 4) новости экономики, бизнеса и финансов, 5) общественные и экологические активисты, 6) демократическая оппозиция.

Описание используемых методов анализа. В анализе социальных сетей можно выделить четыре основных направления: структурное, ресурсное, нормативное и динамическое. В каждом из них решается довольно большой круг задач и применяются методы из различных областей знаний.

Для получения данных о каких-либо изменениях в поведении пользователей возможен сбор временных меток о пользователе и

использование статистического метода для анализа и отклонений, на основе данных о том, как именно пользователи заходят в социальную сеть, с каких устройств и какие действия осуществляют там [4].

Для анализа сообщений пользователя используется рекуррентная нейронная сеть заранее обучения на наборах данных представляющих различные политические точки зрения. С этой целью обработанные файлы мультимедиа, такие как изображения и нейронные сети с помощью сверточной нейронной сети представляемой компанией Microsoft преобразовываются в текст и добавляются к сообщениям.

Для анализа реальности данных профиля пользователя проводится анализ статистической информации – ФИО, это же информация о его друзьях. На основе совпадений делается вывод о реальности данных и круга его друзей.

Отметим, что проблемы публичных систем состоят в невозможности применить весь арсенал математических методов анализа, зачастую применяется Ограниченный круг простейших приемов. Это объясняется тем, что системы закрыты, и исследователи не имеют доступа ко всем данным; не в полной мере учитывают специфику социальных сетей, не обладают возможностью сбора данных и обрабатывают относительно малые их объемы.

С возрастанием количества формируемых данных социальными сетями, а также мощностей вычислительных устройств появляется возможность использовать методы машинного обучения для анализа текста, изображений, а также связей между пользователями и групп пользователей [5].

При обработке текста, чтобы предложение можно было подать на вход нейронной сети, надо решить несколько проблем. Во-первых, необходимо преобразовать слова в цифры. Первый возможный вариант, который можно применить – сопоставить каждому слову из словаря свое число. Второй вариант – закодировать слова длинным вектором, в котором нужному слову соответствует 1, а всем остальным – 0. Здесь все слова равноудалены и не похожи друг на друга. Этот подход гораздо лучше и в ряде случаев работает хорошо [6].

Получив вектора слов, мы сталкиваемся с новой проблемой – как представить цельное предложение для нейронной сети (Рис. 2). Проблема в том, что обычные нейронные сети прямого распространения (feed-forward) должны иметь входные данные фиксированной длины.

Главный результат – нейронная сеть получает возможность строить иерархические модели, формируя в каждом следующем слое более абстрактные представления предложения.

Выводы. В данной статье рассмотрены структурные свойства социальной сети, а также существующие методы анализа социальных сетей.

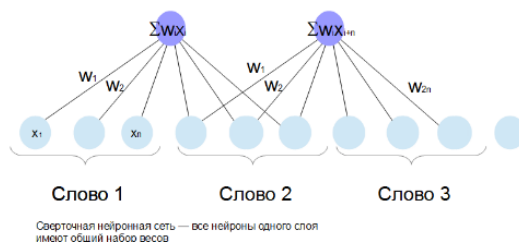


Рисунок 2 – Сверточная нейронная сеть

Для получения, как можно более детальной информации о пользователе необходим детальный сбор данных социальной сети Вконтакте, а также комбинация различных статистических алгоритмов для комплексного анализа данных, включающих различные действия пользователя, а также информацию о его подписках, друзьях и заполненном профиле.

Описанные методы были реализованы программно. Сбор данных осуществляется ботами в реальном времени, были реализованы следующие боты: бот для сбора времени посещений пользователей, бот для работы с пользовательской информацией, бот для работы с группами. Все данные сохраняются в базу, а затем анализируются описанными методами анализа. Расширение скорости сбора данных осуществляется добавлением IP адресов на сервер для дальнейших запросов от разработанных ботов системы. Были отобраны 100 пользователей г. Донецка, а также 100 пользователей из других городов. В результате были получены данные о посещениях пользователя, публикуемых ими сообщений, а также их активности. Был проведен анализ полученных данных с помощью метода анализа временных меток, метода достоверности пользователем. Выявлены интересы пользователей на основании их сообщений в группах и на стене.

Список литературы

1. Pinheiro Carlos A.R. Social Network Analysis in Telecommunications – Hoboken: John Wiley & Sons, 2011. – С.4. – 288 с.
2. Социальная сеть [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2015]. URL: https://en.wikipedia.org/wiki/Social_network_analysis
3. Чураков А. Н. Анализ социальных сетей: СоцИс 2001. № 1. с. 109–121.
4. Хайкин, Саймон. Нейронные сети: полный курс, 2-е изд., испр.: Пер. с англ. – М.: ООО "И.Д.Вильямс", 2006. - 1104 с.
5. Deep Learning [электронный ресурс] // Deep Learning глубокое обучения: [сайт]. [2017]. URL: <http://deeplearning.net>
6. Нейронные сети Facebook [электронный ресурс] // Империя нейросетей: как работает команда машинного обучения Facebook [сайт]. [2017]. URL: <https://rb.ru/story/aml-interview/>

АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ СТРУКТУРИРОВАННОЙ ТЕМАТИЧЕСКОЙ АННОТАЦИИ ДЛЯ РУССКОГО ЯЗЫКА

Тимошков А.В.

Государственное образовательное учреждение
высшего профессионального образования
«Донецкий национальный университет»
кафедра компьютерных технологий
E-mail: tov655kn@gmail.com

Аннотация

Тимошков А.В. Автоматическое построение структурированной тематической аннотации для русского языка. Рассмотрены основные подходы к автоматическому реферированию текста. Изучены популярные алгоритмы для определения веса предложения. Определен подход формирования генерирования аннотации текста.

Общая постановка проблемы

В современном мире электронная информация занимает большую роль во всех сферах жизни общества, а с развитием интернета появилась такая проблема, как информационная перегрузка. Поиск решения данной проблемы, а именно понимание машинами естественного языка, насчитывается уже более 50 лет и связана с именами таких исследователей, как Г.П. Лун, В.Е. Берзон, И.П. Севбо, Э.Ф. Скороходько, Д.Г. Лахути, Р.Г. Пиотровский и др.

Общая структура реферирования состоит из трех взаимосвязанных этапов: анализ входного текста, взвешивание (оценивание) его элементов (слов, словосочетаний, предложений и др.) и непосредственно генерация реферата. Самым трудоемким из них является этап автоматического анализа, который может состоять из нескольких стадий: лексического, лексико-грамматического, синтаксического и семантического анализа.

Исследования

Перед тем, как анализировать текст его сначала необходимо подвергнуть предварительной обработке, в ходе которой удаляются слова с помощью словаря стоп-слов, включающего 179 служебных частей речи, а также неинформативные слова и словосочетания. Дальше с помощью морфологического анализа собираем словарь лемм и разбиваем текст на предложения. Следующим шагом будет определения веса каждой леммы в словаре, а также определения веса предложения:

$$W = \sum(TF) \quad (1)$$

Тогда определения веса слова будет вычисляться по такой формуле:

$$TF = \frac{\text{wordfrequency}}{\text{countwordofdocument}} \quad (2)$$

В каждом из отобранных предложений удаляются примечания (слова в круглых, косых или угловых скобках) и некоторые обороты с помощью специального словаря. Отличие между словарем оборотов и словарем стоп-слов в том, что неинформативные слова целиком задаются списком, а в словаре оборотов имеется только начальная часть фразы, например, «Как сообщается», «Как стало известно». После распознавания в тексте удаляется из предложения не только эта часть, но и весь оборот до знака препинания вместе с ним. Далее определяем количество предложений в реферате и генерируем реферат в зависимости от веса предложения.

Для данной задачи был выбран язык программирования python из его достоинств можно выделить то что он направлен на продуктивность программиста. Благодаря чему на нем можно написать, что угодно. Его код лаконичен и понятен даже тому, кто не изучал данный язык. Также данный язык входит в топ мира по языкам программирования и им пользуется большинство крупных компаний. Для морфологического анализа будет использована `rumorph2`, который основан на словарном методе.

Выводы

В результате исследования было создано приложение на python, которое генерирует аннотацию текста. В качестве дополнительных возможностей программа может показать количество предложений в тексте, среднюю длину предложения, количество знаков в тексте, топ 10 лемм.

Список литературы

1. Абрамов В.Е. О методе автоматического реферирования, основанном на результатах рубрицирования документов / В.Е. Абрамов Н.Н. Абрамова [Электронный ресурс]. – Режим доступа: http://rcdl.ru/doc/2008/177_183_paper20.pdf (дата обращения: 27.04.2017 г.).

2. Большакова Е.И. автоматическая обработка текстов на естественном языке и компьютерная лингвистика / Е.И. Большакова, Э.С. Клышинский, Д.В. Ландэ, А.А. Носков, О.В. Пескова, Е.В. Ягунова // «Прикладная математика». МИЭМ, 2011. – 272 с.

3. Губин М.В. эффективный алгоритм формирования контекстно-зависимых аннотаций/ М.В.Губин, А.И. Меркулов// ИК «Кодекс», Санкт-Петербург

УДК 004.891.2

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ НАХОЖДЕНИЯ ОПТИМАЛЬНОГО РЕШЕНИЯ КОМБИНАТОРНЫХ ЗАДАЧ ПУТЁМ ПРИМЕНЕНИЯ ОПТИМАЛЬНОЙ КОМБИНАЦИИ ГЕНЕТИЧЕСКИХ ОПЕРАТОРОВ

А.А. Купоросов, Д.М. Бочаров

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
kupo_09@mail.ru

Аннотация

Купоросов А.А., Бочаров Д.М. Программное средство для нахождения оптимального решения комбинаторных задач путём применения оптимальной комбинации генетических операторов. Выполнен анализ операторов генетических алгоритмов. Определены достоинства и недостатки стандартных генетических алгоритмов.

***Ключевые слова:** генетический алгоритм, генетический оператор, комбинация, комбинаторные задачи.*

Постановка проблемы. В современном мире часто приходится решать различные задачи, которые относятся к группе оптимизационных задач. В связи с этим возникает необходимость выбора какую комбинацию генетических операторов необходимо использовать для достижения поставленной цели, и как провести анализ оптимального решения для комбинаторных задач.

Цель статьи – выделить основные генетические операторы для дальнейшей их реализации в программном продукте, а также привести примеры комбинаторных задач, на которых будет проходить дальнейшее тестирование программного средства.

Основные понятия и принципы генетических алгоритмов. Генетические Алгоритмы (ГА) – это адаптивные методы функциональной оптимизации, основанные на компьютерном имитационном моделировании биологической эволюции. Основные принципы ГА были сформулированы Голландом (Holland, 1975), и хорошо описаны во многих работах и на ряде сайтов в Internet.

Генетические алгоритмы основываются на теории Дарвина – существует популяция, состоящая из особей одного вида, которые способны давать потомство. В основе этой модели лежат случайные изменения генов при переходе от поколения к поколению. Целесообразные изменения, которые

облегчают выживание и производство потомков в данной конкретной внешней среде, сохраняются и передаются потомству, т.е. наследуются. Особи, не имеющие соответствующих приспособлений, погибают, не оставив потомства или оставив его меньше, чем приспособленные (считается, что количество потомства пропорционально степени приспособленности). Поэтому в результате естественного отбора возникает популяция из наиболее приспособленных особей, которая может стать основой нового вида [1].

На рисунке 1 представлен пример работы простого генетического алгоритма.

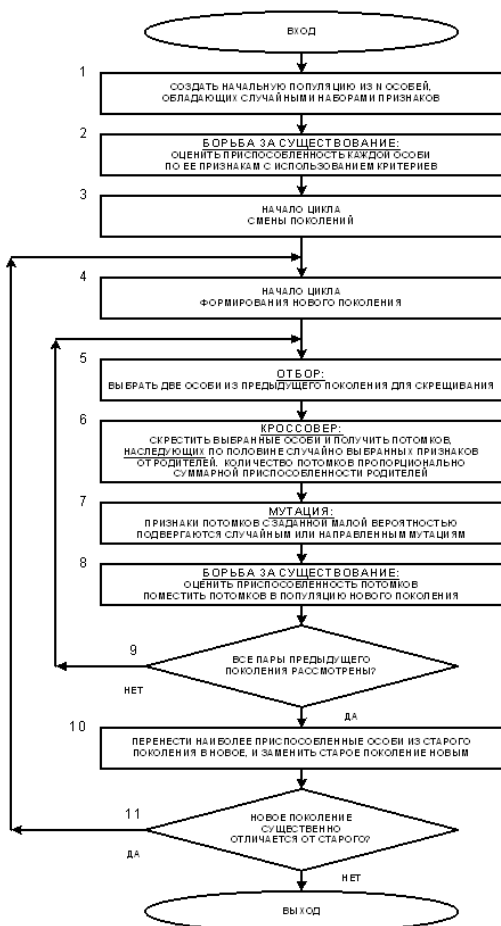


Рисунок 1 – Пример работы простого генетического алгоритма

Основные операторы генетического алгоритма. Основные операторы генетического алгоритма можно разбить на 3 группы:

- операторы выбора родителей (операторы отбора);
- операторы скрещивания (рекомбинации);
- мутация.

Операторы отбора служат для выбора родительской пары. Основные операторы отбора [3]:

1) панмиксия – каждому члену популяции сопоставляется случайное целое число на отрезке от $[1;n]$, где n – количество особей в популяции; происходит генерация случайного целых чисел на этом отрезке и полученные числа сопоставляются номерам членам популяции, именно они и будут родителями;

2) инбридинг – представляет собой метод, когда первый родитель выбирается случайным образом, а вторым родителем является член популяции ближайший к первому; метод можно охарактеризовать свойством концентрации поиска в локальных узлах, что приводит к разбиению популяции на отдельные локальные группы вокруг подозрительных на экстремум участков;

3) аутбридинг – так же используют понятие схожести особей, однако теперь пары формируют из максимально далёких особей; метод заставляет алгоритм просматривать новые, неисследованные области;

4) селекция – родителями могут стать только те особи, значение приспособленности которых не меньше пороговой величины; такой подход обеспечивает более быструю сходимости алгоритма; пороговая величина может быть вычислена разными способами, поэтому выделяют несколько вариантов селекции:

а) турнирный отбор – из популяции, содержащей n особей выбирают случайным образом t особей, и лучшая из них особь записывается в промежуточный массив; эта операция повторяется n раз, особи в полученном промежуточном массиве затем используют для скрещивания (случайным образом); размер группы строк часто равен 2, что говорит о двоичном турнире; преимущество данного способа является то, что он не требует дополнительных вычислений;

б) метод рулетки – особи отбираются с помощью n «запусков» рулетки; колесо рулетки содержит по одному сектору для каждого члена популяции, а размер сектора пропорционален вероятности попадания в новую популяцию; при таком отборе члены популяции с более высокой приспособленностью с большей вероятностью будут чаще выбираться, чем особи с низкой приспособленностью.

5) другие способы отбора можно получить на основе модификации выше приведённых.

Операторы рекомбинации применяют сразу же после оператора отбора родителей для получения новых особей-потомков. Смысл рекомбинации заключается в том, что созданные потомки должны наследовать генную информацию от обоих родителей. Различают несколько видов рекомбинации:

1) дискретная рекомбинация – в основном применяется к хромосомам с вещественными генами; основными способами дискретной рекомбинации является собственное дискретная рекомбинация, промежуточная, линейная и расширенно линейная рекомбинация; дискретная рекомбинация применима для любого типа генов (двоичные, вещественные и символьные);

2) промежуточная рекомбинация (дифференциальное скрещивание) – применима только к вещественным переменным, но не к бинарным; в данном методе предварительно определяется числовой интервал значений генов потомков, который должен содержать значения генов родителей;

3) линейная рекомбинация – отличается от промежуточной тем, что множитель выбирается для каждого потомка один раз;

4) кроссинговер (бинарная рекомбинация) – рекомбинацию бинарных строк принято называть кроссинговером (кроссовером) или скрещиванием; существует несколько видов кроссинговера:

а) одноточечный кроссинговер – случайным образом определяется точка внутри хромосомы (точка разрыва), в которой обе хромосомы делятся на две части и обмениваются ими;

б) двухточечный кроссинговер – хромосомы рассматриваются как циклы, которые формируются соединением концов линейной хромосомы вместе; для замены сегмента одного цикла сегментом другого цикла требуется выбор двух точек разреза;

в) многоточечный кроссинговер – точки разреза выбираются случайно и сортируются в порядке возрастания после чего происходит обмен участками хромосом, ограниченными точками разреза и таким образом получают двух потомков; участок особи с первым геном до первой точки разреза в обмене не участвует;

г) однородный кроссинговер – создаёт маску (схему) особи, в каждом локусе которой находится потенциальная точка кроссинговера, маска имеет ту же длину, что и скрещивающиеся особи.

После процесса воспроизводства происходят мутации. Данный оператор необходимым для «выбивания» популяции из локального экстремума и препятствует преждевременной сходимости. Это достигается за счёт того, что изменяется случайно выбранный ген в хромосоме. Так же как и кроссинговер, мутации могут проводиться не только по одной случайной точке, а можно выбирать для изменения несколько точек в хромосоме, причём из число также может быть случайным.

Примеры комбинаторных задач. Множество задач, имеющих одинаковую постановку и отличающихся только значениями параметров, называются – массовой задачей. Примеры таких задач приведены ниже.

1. Задача об одномерном ранце. Из заданного набора предметов, каждый из которых имеет вес и ценность (стоимость), нужно выбрать несколько и наполнить ими ранец таким образом, чтобы суммарная выгода по выбранным предметам была наибольшей, а их общий вес не превосходил вместимость ранца; задача легко реализуется с помощью бинарного представления хромосом;

2. Задача покрытия множества. Пусть задана булева матрица размером $m \times n$, с каждым столбцом которой связано положительное значение стоимости; построение покрытия минимальной стоимости может быть сформулировано как задача целочисленного линейного программирования, если в качестве решения принять вектор с бинарными переменными; как и в предыдущем случае допустимым решением является бинарный вектор, применяется кодировка генов в виде бинарной строки длиной n .

3. Задача о назначениях. Пусть имеется n работ и n кандидатов на их выполнении, причём назначение j -й работы i -му кандидату требует затрат; задача состоит в нахождении такого распределения кандидатов по работам, чтобы минимизировать суммарные затраты, причём каждый кандидат может быть назначен только в одну работу, и каждую работу может выполнить только один кандидат; каждое решение можно закодировать перестановкой, в которой позиция обозначает номер кандидата, а значение в этой позиции означает номер работы, назначенной кандидату.

4. Задача коммивояжера. Широко известная задача оптимизации комбинаторного типа. Дан полный взвешенный граф порядка n , в нём необходимо найти Гамильтонов цикл, имеющий наименьший суммарный вес входящих в него рёбер; очевидно, что решением данной задачи является перестановка из n вершин, количество возможных перестановок будет равно $n!$, что и делает её решения с помощью ГА оптимальным.

Достоинства и недостатки генетических алгоритмов. Генетические алгоритмы не гарантируют обнаружения глобального решения за приемлемое время, так же они не гарантируют того, что решение будет оптимальным. Тем не менее они применимы для поиска решения близкого к идеальному за достаточно короткое время. ГА имеют преимущество перед другими алгоритмами при очень больших размерностях задач и отсутствия упорядоченности в исходных данных, когда альтернативой им является метод полного перебора [4].

В случаях, когда задача может быть решена специально разработанным для неё методом, практически всегда такие методы будут эффективнее ГА как по быстроте действия, так и по точности найденных решений.

Главным же достоинство ГА является то, что они могут применяться для решения сложных неформализованных задач, для которых не разработано специальных методов.

Описание разрабатываемого программного средства. Программное средство для нахождения оптимального решения комбинаторных задач предназначено для использования на настольных компьютерах, так как целевая аудитория – это разработчики программного обеспечения, которые хотят использовать в своих задачах генетические алгоритмы, и нуждаются в их тестировании, а именно: нахождении оптимальной скорости работы или точности решения.

Программное средство будет включать в себя:

- выбор одного из типа вышеописанной задачи;
- задание начальных параметров выбранной задачи;
- конструктор генетических операторов, которые можно расположить в разном порядке для более глубокого тестирования;
- вывод результатов работы алгоритма и затраченное время;
- сравнение результатов работы различных пользовательских комбинаций;
- сохранение заранее созданных комбинаций генетических операторов;
- ведение логов в файл.

Выводы. Произведено исследование основных генетических операторов, а также приведён пример основных комбинаторных задач. Осуществлён анализ разрабатываемого программного средства для нахождения оптимального решения комбинаторных задач и определены его основные функции.

Список литературы

1. Стецюра Г.Г. Эволюционные методы в задачах управления, выбора, оптимизации / Стецюра Г.Г. – М.: ФИЗМАТЛИТ, 2006. – 320с.
2. Генетические алгоритмы [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: https://ru.wikipedia.org/wiki/Генетический_алгоритм
3. Генетический алгоритм. Просто о сложном [электронный ресурс] // Хабрахабр [сайт] [2011]. URL: <https://habrahabr.ru/post/128704>
4. Панченко Т.В. Генетические алгоритмы: учебно-методическое пособие / Панченко Т.В. – Астрахань : Издательский дом «Астраханский университет», 2007. – 83с.

УДК 519.854.2

ЭКОНОМИКО-МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ МЕТОДОМ ИМИТАЦИИ ОТЖИГА

Б.А. Шуляк, Ю.Е Харитонов

Донецкий национальный технический университет, г. Донецк
кафедра экономической кибернетики
bogdanbo@mail.ru

Аннотация

Шуляк Б.А., Харитонов Ю.Е. Экономико-математическое моделирование методом имитации отжига. Выполнен анализ метода имитации отжига для решения оптимизационных задач в экономике. На основании данного метода разработан алгоритм решения для типовой оптимизационной задачи и диаграмма классов для реализации данной задачи на объектно-ориентированном языке программирования.

***Ключевые слова:** экономико-математическая модель, оптимизационная задача, алгоритм имитации отжига, диаграмма классов.*

Постановка проблемы. В настоящее время существует большее количество экономических задач, которые требуют оптимизационного подхода при их решении. Такому типу задач необходима разработка отлаженного алгоритма решения, который позволит достичь максимальной прибыли (минимальных затрат) в результате их решения. Данный алгоритм целесообразно разработать на основании метода имитации отжига. Для этого необходимо выполнить следующее:

- проанализировать алгоритм имитации отжига, описать его основные функции и их назначения, выделить перспективы дальнейшего развития;
- разработать алгоритм решения типовой оптимизационной задачи на основании метода имитации отжига: переопределить исходные функции в зависимости от поставленной задачи;
- представить данную модель решения поставленной задачи в виде диаграммы классов для её последующей реализации на объектно-ориентированном языке программирования.

Цель статьи – провести анализ метода имитации отжига и на основании его разработать алгоритм решения типовой оптимизационной задачи.

Анализ алгоритма имитации отжига. Алгоритм имитации отжига(англ. Simulated annealing) — общий алгоритмический метод решения задачи глобальной оптимизации, предложенный Николасом Метрополисом.

Алгоритм имитирует физический процесс, который заключается в нагреве и последующем охлаждении субстанции. В результате данного процесса образуется прочная кристаллическая структура, которая отличается от дефектной структуры при быстром беспорядочном охлаждении. Структура представляет собой координированное решение, а температура используется для того, чтобы указать, как и когда будут приниматься новые решения[2].

Переход атома из одной ячейки в другую происходит с некоторой вероятностью, причём вероятность уменьшается с понижением температуры. Устойчивая кристаллическая решётка соответствует минимуму энергии атомов, поэтому атом либо переходит в состояние с меньшим уровнем энергии, либо остаётся на месте[3].

Сам алгоритм отжига может быть описан следующим образом. Для этого введём обозначения

S – множество всех состояний (решений) нашей задачи.

s_i – состояние на i -м шаге алгоритма, причём $s_i \in S$.

$t_i \in R$ – температура на i -м шаге алгоритма.

Для того, чтобы использовать имитацию отжига, нам понадобится определить три функции:

– функцию энергии $E: S \rightarrow R$

– функцию изменения температуры во времени $T: N \rightarrow R$

– функцию, порождающую новое состояние $F: S \rightarrow S$

Теперь остановимся подробнее на каждой из функций и их назначениями.

Функция E каждому решению по определённому правилу ставит в соответствие число. Зависит от конкретной задачи.

Функция T определяет температуру на шаге итерации i . Функция определяет, как долго будет работать наш алгоритм. Если T будет линейной функцией, то время работы будет относительно большим.

Функция F на основе предыдущего состояния порождает новое состояние – кандидат s_c , в которое система может перейти, а может и отбросить – способ получения кандидата полностью зависит от решаемой задачи.

Приведем в самом общем виде описание алгоритма метода имитации отжига [1]:

– На входе: задаётся минимальная температура t_{min} , начальная температура t_{max} .

– Задается произвольное первое состояние s_1 .

– Если $t_i > t_{min}$, то

– $s_c = F(s_{i-1})$,

– $\Delta E = E(s_c) - E(s_{i-1})$,

– если $\Delta E \leq 0$, тогда $s_i = s_c$,

– если $\Delta E > 0$, тогда переход осуществляется с вероятностью $P(\Delta E) = e^{-\Delta t_i}$,

- Понижаем температуру $t_{i+1} = T(i)$.
- Возвращаем последнее состояние s .

Отдельно стоит пояснить переход в новое состояние. Если энергия «кандидата» меньше, он становится новым состоянием, в противном случае, переход будет вероятностным (поэтому метод относят к классу стохастических).

Метод отжига имеет перспективу дальнейшего развития для оптимизации задач следующих классов:

- создание пути;
- реконструкция изображения;
- назначение задач и планирование;
- размещение сети;
- глобальная маршрутизация;
- обнаружение и распознавание визуальных объектов;
- разработка специальных цифровых фильтров.

Разработка алгоритма решения типовой оптимизационной задачи.

Задача коммивояжера является одной из знаменитых задач теории комбинаторики. В своей области (оптимизации дискретных задач) задача коммивояжера служит своеобразным полигоном, на котором испытываются все новые методы. Постановка задачи следующая [4]:

Коммивояжер должен выйти из первого города, посетить по разу в неизвестном порядке города 2,1,3...n и вернуться в первый город. Расстояния между городами известны. В каком порядке следует обходить города, чтобы замкнутый путь (тур) коммивояжера был кратчайшим?

Пусть города случайным образом разбросаны в квадрате 10×10 . Обозначим множество всех городов C , а их количество обозначается $|C|$ и равно 100. Каждый город можно представить как пару координат с соответствующим индексом.

По условию решением является кратчайший маршрут между всеми городами, значит множество состояний S – это все возможные маршруты, проходящие через каждый город. Другими словами множество всех упорядоченных последовательностей городов C , в которых всякий элемент встречается ровно один раз. Очевидно, что длина каждой такой последовательности $|C|$.

Для того, чтобы использовать метод имитации отжига, необходимо определить две функции, зависящие от каждой конкретной задачи. Это функция энергии E (или «целевая функция» в общепринятой терминологии) и функция F , порождающая новое состояние.

Так как цель алгоритма – минимизировать расстояние, оно и будет «энергией». Следовательно целевая функция будет иметь вид суммы евклидовых расстояний между парами городов в маршруте s_i :

$$E(s_i) = E_i = \left[\sum_1^{|C|-1} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \right] + \sqrt{(x_{|c|} - x_1)^2 + (y_{|c|} - y_1)^2}$$

Чтобы получить новое состояние необходимо местами два произвольных города в маршруте. Однако такое изменение непредсказуемо влияет на E_i и метод будет работать долго и не факт, что успешно.

Оптимальным вариантом в данном случае будет выбрать два произвольных города в маршруте и инвертировать путь между ними. Например имелся маршрут (1,2,3,4,5,6,7). Генератор случайных чисел выбрал города 2 и 7, выполнив процедуру получаем (1,7,6,5,4,3,2).

Теперь можно начать процесс оптимизации. Ниже представлена диаграмма классов, которая демонстрирует данный процесс, используя метод имитации отжига (рис. 1).

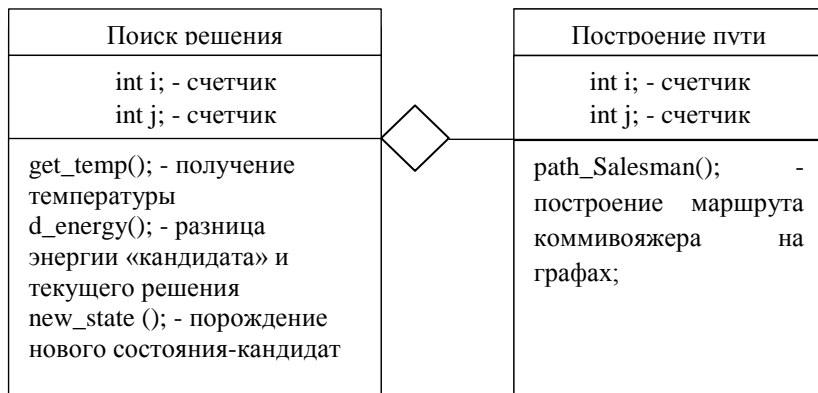


Рисунок 1 – Диаграмма классов

При написании данного алгоритма на каком-либо объектно-ориентированном языке он может быть оптимизирован в зависимости от сложности выполняемой задачи. Допускается переопределять следующие параметры: начальную и конечную температуры, функцию изменения температуры и количество итераций при одном значении температуры.

Приведенный выше алгоритм был реализован на объектно-ориентированном языке C++ с использованием консольной графики на основе разработанной диаграммы классов.

Пример функции вычисления энергии (E):

```
void computeEnergy()//вычисление энергии
{
    int i,j,x,y,tempx,tempy;
    char board[MAX_LENGTH][MAX_LENGTH];
    const int dx[4]={-1,1,-1,1};
    const int dy[4]={-1,1,1,-1};
    memset( board,0, MAX_LENGTH * MAX_LENGTH );//функция
очистки памяти
    for (i=0;i<MAX_LENGTH;i++){
        board[i][solution[i]]='Q';}
    /* Считает количество конфликтов */
    int conflicts=0;
    for (i=0;i<MAX_LENGTH;i++){
        x=i;y=solution[i];
        for (j=0;j<4;j++){//проверяем конфликты по диагонали
            tempx=x;tempy=y;
            while(1){
                tempx+=dx[j];tempy+=dy[j];
                if ((tempx<0) ||
(tempx>=MAX_LENGTH)||((tempy<0)||((tempy>=MAX_LENGTH))break;
                if(board[tempx][tempy]=='Q') conflicts++;}
            }
        }
        energy=(float)conflicts;
    }
}
```

Выводы. В данной работе проанализирован алгоритм имитации отжига, описаны его основные функции и их назначения. Разработан алгоритм решения типовой оптимизационной задачи на основании метода имитации отжига. В дальнейшем данный алгоритм может быть реализован на любом объектно–ориентированном языке программирования.

Список литературы

1. Алгоритм имитации отжига [электронный ресурс] URL: http://www.machinelearning.ru/wiki/index.php?title=Алгоритм_имитации_отжига
2. Бураков, М.В. Генетический алгоритм: теория и практика: учебное пособие / М. В. Бураков. – СПб.: ГУАП, 2008. – 164 с.URL: <http://window.edu.ru/catalog/pdf2txt/205/80205/60612>
3. Джонс М. Т. Программирование искусственного интеллекта в приложениях. –М.: ДМК Пресс, 2004. – 312 с. – ISBN 5-94074-275-0.. – С.25–42.
4. Задача коммивояжера | Метод имитации отжига [электронный ресурс] URL: <http://mech.math.msu.su/~shvets/54/inf/perl-problems/chCommisVoyageur>

УДК 004.932.2

МЕТОД НАХОЖДЕНИЯ ЦЕНТРА ЗРАЧКА НА ИЗОБРАЖЕНИИ ГЛАЗА

Потёмкин А.С., Миненко А.С.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: lihoj21@gmail.com

Аннотация

Потёмкин А.С., Миненко А.С. Метод нахождения центра зрачка на изображении глаза. В работе рассмотрены основные подходы для нахождения границ и центра зрачка. Подробно рассмотрен предлагаемый алгоритм нахождения центра зрачка. Проиллюстрировано выполнение основных шагов алгоритма. По результатам экспериментов можно сделать выводы о точности и быстродействии предложенного подхода. В качестве источника тестовых изображений использовалась открытая база фотографий CASIA-Iris-Distance.

Ключевые слова: поиск центра зрачка, преобразование Хафа, признаки Хаара, обработка изображений.

Общая постановка проблемы. На практике возникают ситуации, когда необходимо найти точки, которые соответствуют центрам зрачков. Например, расстояние между зрачками можно использовать для локализации основных фрагментов лица (таких как губы, брови, нос), чтобы сократить область поиска, тем самым снизив время на нахождения этих фрагментов. Также, центры зрачков можно использовать для определения наклона головы относительно линии горизонта, что позволяет снизить ошибку локализации основных фрагментов лица при значительных наклонах головы.

Большинство существующих методов нахождения зрачков нацелены на нахождение границы зрачка и радужной оболочки (так называемого «ириса»), а задача нахождения именно центра является второстепенной. Например, нахождение этой границы является промежуточным этапом в задаче идентификации человека по ирису. Но далеко не всегда необходима граница зрачка, иногда достаточно и центральной точки. Например, при определении эмоционального состояния человека на изображении, нет необходимости определять границу зрачка, так как сам зрачок не вносит вклада в эмоциональное состояние.

Исследование. Для решений задачи нахождения центра и границы зрачка существует большое количество методов [1,2,3]. В основном их можно разделить на следующие группы:

- методы, основанные на преобразовании Хафа;
- проекционные методы по яркости и градиенту яркости;
- морфологические методы;
- методы, основаны на построении оптимального контура.

Методы, которые используют преобразование Хафа [1], достаточно эффективно применяются на большинстве изображений глаз, но время поиска достаточно длительное. В этом методе определяются координаты центра зрачка и его радиус. Чтобы найти округлости на изображении используют преобразования Хафа:

$$H(x_0, y_0, r) = \sum_i h(x_i, y_i, x_0, y_0, r), \quad (1)$$
$$h(x_i, y_i, x_0, y_0, r) = \begin{cases} 1, & \text{если } (x_i - x_0)^2 + (y_i - y_0)^2 = r^2, \\ 0, & \text{иначе} \end{cases}$$

где (x_0, y_0) – определяемые координаты центра зрачка; (x_i, y_i) – текущий пиксель на изображении в некоторой области; $r \in [r_{min}, r_{max}]$ – предполагаемый радиусы зрачка.

На основании формулы (1) выполняется построение трехмерного аккумуляторного пространства, из которого и определяется округность. В процессе голосования каждой точке на изображении ставится в соответствие некоторое множество точек из аккумуляторного пространства. Результатом является точка из аккумуляторного пространства, которая попала в наибольшее число множеств.

Морфологические методы часто используется с подходами, которые основаны на вычислении проекции яркости по осям. Область зрачка определяется как минимум ярости на обеих осях, а с помощью морфологических операций выделяется сам зрачок. Однако такой подход часто находит брови или ресницы, как часть зрачка или вместо него. Это происходит, так как яркость зрачка может вовсе не отличаться от волосяного покрова на участке глаз.

Для построения оптимального контура [2] необходимо иметь начальную точку внутри зрачка, но для нахождения этой точки возникают подобные проблемы, что и для метода на основе проекции яркости.

Описание алгоритма. Так как существует ряд задач, для которых нет необходимости находить границу зрачка, был разработан алгоритм для нахождения только центра зрачка.

На вход поступает цветное или полутоновое изображение, на котором присутствует глаз и зрачок в частности. На рисунке 1 приведены основные шаги алгоритма.

На первом этапе, если изображение цветное, то необходимо перевести его в оттенки серого, однако на практике было установлено, что красный канал позволяет более четко отделить зрачок от ресниц. Поэтому имеет смысл сначала выделить красный канал.



Рисунок 1 – Основные шаги алгоритма

Для каждого пикселя изображения находится средняя яркость для окрестности по формуле (2).

$$I(x, y) = \frac{\sum_{i=x-\frac{N}{2}, j=y-\frac{M}{2}}^{x+\frac{N}{2}, y+\frac{M}{2}} I(i, j)}{N * M}, \quad (2)$$

где N и M это размер окрестности (в данном случае использовался 1/24 от высоты и ширины соответственно); $I(x, y)$ – это яркость пикселя с координатами (x,y).

После усреднения (своеобразное сглаживание) находим точку с наименьшей яркостью, которая предположительно находится на зрачке.

Производится бинаризация изображения в соответствии с формулой (3).

$$I(x, y) = \begin{cases} 1, & \text{если } I(x, y) < \min + \tau \\ 0, & \text{иначе} \end{cases}, \quad (3)$$

где min – это минимальная яркость на усредненном изображении, τ – некоторое пороговое значение (в данном случае использовалось $\tau = 12$).

После того как было получено бинарное изображение, необходимо «раскрасить» изображение. Каждая неокрашенная связанная область помечается новым цветом, пока все области не будут окрашены.

Зная точку, которая лежит на зрачке, получаем цвет связанной области, которая предположительно зрачок и удаляем области «не зрачок» используя формулу (4).

$$I(x, y) = \begin{cases} 1, & \text{если } I(x, y) = I(x_{min}, y_{min}), \\ 0, & \text{иначе} \end{cases}$$

где, $I(x_{min}, y_{min})$ – яркость точки, которая лежит на зрачке; $I(x, y)$ – яркость текущей точки.

В результате удаления областей «не зрачок», на бинарном изображении останется только область, которая является зрачком. На этом этапе, необходимо определить крайние точки связанной области. Для этого от точки, которая лежит на зрачке (x_{min}, y_{min}) будем двигаться в вверх, вниз, влево, вправо для нахождения верхней, нижней, левой и правой точки соответственно, анализируя цвет текущей точки. После того как точка достигнет перехода из 1 в 0, построчно или по столбцам вычисляется сумма пока не найдет нулевую строку/столбец. Он и будет краем. Так поочередно находятся 4 точки: top, bottom, left, right которые соответствуют верхнему, нижнему, левому и правому краю связанной области.

Имея граничные точки можно получить центр связанной области, как продемонстрировано формулой (5).

$$\begin{aligned} x_{center} &= \frac{bottom - top}{2}, \\ y_{center} &= \frac{right - left}{2}. \end{aligned} \quad (5)$$

На рисунке 2 приведены результаты выполнения описанных выше шагов для одного из изображений.

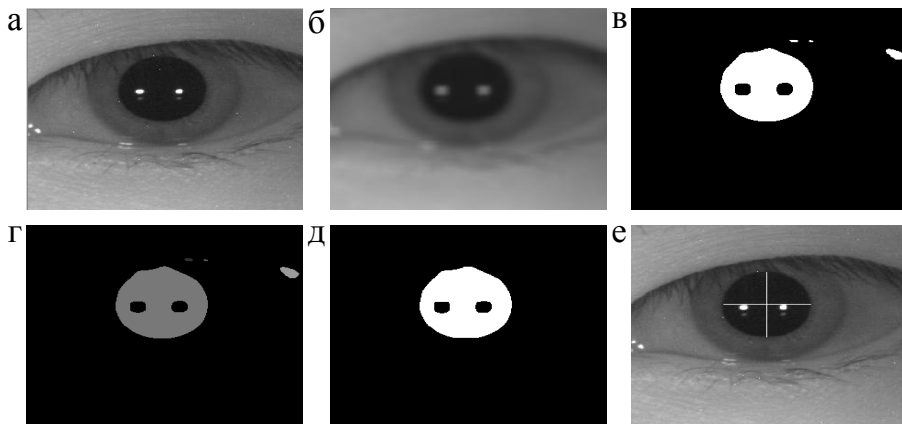


Рисунок 2 – Результаты выполнения шагов алгоритма. а – исходное изображение, б – изображение после усреднения, в – изображение после бинаризации, г – изображение после раскраски, д – изображение после удаления «не зрачков», е – найденный центр

Результаты экспериментов. В качестве источника изображений для экспериментов использовалась база CASIA-Iris-Distance (обработано 2567 изображений, размер каждого 2352x1728). В этой базе содержатся изображения лиц, а не глаз, поэтому изначально производился поиск области с глазами, а затем выполнялся поиск центра зрачка. Для поиска глаз на лице использовались признаки Хаара. На 2567 изображениях было выделено 4406 областей с глазами (размер каждого около 300x300). К сожалению, на 363 изображениях не удалось найти глаза. Результаты работы алгоритма для нахождения центра зрачка приведены в таблице 1. Большинство ошибочных определений происходили из-за очков, сильных отблесков. Также качества нахождения центра сильно снижалось за счет прищуренных глаз.

Таблица 1. Результаты работы алгоритма

Верных, %	Ошибок, %	Min время, мс	Max время, мс	Среднее время, мс
93,6	6,4	125	432	156

Выводы. Так как не всегда необходимо находить границу зрачка, а достаточно знать координаты его центра, предложенный метод выгодно смотрится на фоне своих конкурентов в плане быстрей действия, не сильно уступая в показателях точности полученных результатов. Однако в связи с некоторыми особенностями, качество результатов сильно ухудшается если зрачок перекрывается верхним веком. Также стоит отметить что фотографии из базы CASIA-Iris-Distance сделаны на инфракрасную камеру, за счет чего область минимума четко попадает в область зрачка. Поэтому возможно некоторое снижение качества определения центра зрачка на фотографиях с естественным освещением.

Список литературы

1. Wildes, R., Asmuth, J., Green, G., Hsu, S., Kolczynski, R., Matey, J., McBride, S. / A system for automated iris recognition. –Sarasota, FL: Proceedings IEEE Workshop on Applications of Computer Vision. – 1994. – P.121-128.
2. Ritter, N. Location of the pupil-iris boarder in slit-lamp images of the cornea [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/221356058_Location_of_the_Pupil-Iris_Border_in_Slit-Lamp_Images_of_the_Cornea.
3. Gui, F. Iris localization scheme based on morphology and gaussian filtering / F. Gui, L. Qiwei // Third Intern. IEEE Conf. on Signal-Image Technologies and Internet-Based System. Shanghai, China. – 2007. –P.798-803.
4. Харитонов А.В., Олейник А.Г. Метод Определения границы зрачка на изображении глаза. / А.В. Харитонов, А.Г. Олейник // Апатиты. Труды Кольского научного центра РАН. Информационные технологии V. – 2014. – С. 171-177.

УДК 004.932.72'1

ИССЛЕДОВАНИЕ И РАЗРАБОТКА АЛГОРИТМА РАСПОЗНАВАНИЯ СЛОЖНЫХ НАКЛАДЫВАЮЩИХСЯ ГЕОМЕТРИЧЕСКИХ ОБЪЕКТОВ

Минков О.В., Ручкин К.А.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа

Аннотация

Минков О.В., Ручкин К.А. Исследование и разработка алгоритма распознавания сложных накладывающихся геометрических объектов. Рассмотрен алгоритм бактериальной оптимизации для нахождения и распознавания геометрических фигур, в частности окружностей, а также различных его модификация. Проведен анализ классического алгоритма с его улучшенными версиями. Предложена своя модификация для одновременного нахождения окружностей.

Ключевые слова: распознавание, компьютерное зрение, эволюция, BFOA, геометрические фигуры.

Введение. В современном мире очень интенсивно развиваются различные системы компьютерного зрения, что в свою очередь позволяет эффективно взаимодействовать человеку и технике. Одной из самых важных и нужных целей которые ставит перед собой системы компьютерного зрения, является задача автоматического распознавания образов, решение которой позволит корректно оценивать окружающую среду, а затем выполнять в ней необходимые действия. Но данное направление является достаточно обширным и может применяться абсолютно в разных сферах жизнедеятельности. Например, это может пригодиться в строительной сфере, в сфере медицины, образования, машиностроения вплоть до военного дела. Соответственно меняется и сам объект распознавания, будь то человек, машина, животное, растение, строение и так далее.

Постановка задачи. В рамках данной работы мы будем рассматривать алгоритм бактериальной оптимизации для нахождения и распознавания геометрических фигур, в частности окружностей, а также различных его модификаций. Проведем небольшой анализ классического алгоритма с его улучшенными версиями. Затем рассмотрим модификацию для одновременного нахождения окружностей и приведем таблицу результатов по скорости и по точности.

Анализ предметной области. В настоящее время достаточно большое количество работ посвящено исследованию применения эволюционных методов для решения задач распознавания изображений. Рассмотрим эти работы более подробно.

Алгоритм BFO и различные его модификации. Одной из особенностей алгоритма BFO является нахождение накладывающихся объектов. Накладывающимся объектами выступают как пересекающиеся объекты, так и объекты, которые соприкасаются с друг другом.

В статье R. Panda, M. K. Naik [1] представлен модифицированный алгоритм бактериальной оптимизации, который называется кроссоверный алгоритм бактериальной оптимизации, который наследует кроссоверные техники от генетических алгоритмов. Это может быть использовано для импровизации значения оптимальной целевой функции.

Идея использования кроссовер механизма состоит в том, чтобы искать близлежащие места потомства (50 процентов бактерий), потому что они производятся в разных случайных местах. В традиционном алгоритме бактериальной оптимизации поиск начинается с некоторой позиции (50 процентов бактерий дублируются) что не желательно. Для оценки эффективности учитываются семь различных эталонных функций. Также приведено сравнение с результатами предыдущих методов, чтобы выявить эффективность предложенного алгоритма.

В статье Chin-Ling Lee, Cheng-Jian Lin [2] представлен алгоритм бактериальной оптимизации на основе адаптационной стратегии для решения сложных оптимизационных задач (SABFO). Предлагаемый алгоритм SABFO использует стратегический подход на этапе хемотаксиса традиционного алгоритма бактериальной оптимизации (BFO). Предлагаемый метод позволяет каждой бактерии плавать на разные расстояния, а также увеличивает бактериальное разнообразие. Для проверки производительности SABFO используются пять задач оптимизации нелинейных контрольных функций. Результаты моделирования показывают, что SABFO получает лучшие глобальные оптимальные решения, чем другие методы.

Алгоритм бактериальной оптимизации. Алгоритм бактериальной оптимизации (BFOA) предложенный Кевином Пассино (Kevin Passino, 2002), является новым пришествием в семействе оптимизационных алгоритмов, вдохновленных природой. Ключевой идеей этого нового алгоритма является применение групповой стратегии кормления роя бактерий *E.coli* при оптимизации функций. Бактерии ищут питательные вещества, чтобы максимизировать энергию, получаемую за единицу времени. Отдельная бактерия также общается с другими, отправляя сигналы. Бактерия принимает решения о кормлении после рассмотрения двух предыдущих факторов. Процесс, в котором бактерия движется, делая небольшие шаги при поиске питательных веществ, называется хемотаксисом. Ключевая идея BFOA -

имитировать хемотаксическое движение виртуальных бактерий в пространстве поиска проблем.

R: размер поискового пространства,

S: общее количество бактерий в популяции,

Ns: количество шагов хемотаксиса,

Ns: Длина плавания.

Nre: количество шагов воспроизведения,

Ned: количество событий выбывания-рассеивания,

Ped: вероятность выбывания-рассеивания,

C(i): размер шага, выполненного в случайном направлении, указанном кувырканием.

Теория поиска пищи, основывается на предположении, что животные ищут и получают питательные вещества, для того чтобы увеличить потребление энергии E, за единицу времени T, затраченного на поиск пищи. Следовательно, они пытаются максимизировать функцию типа E/T (или максимизировать свой долгосрочный средний уровень потребления энергии). Максимизация такой функции обеспечивает источники питательных веществ для выживания и дополнительного времени для других важных видов деятельности (например, борьба, бегство, спаривание, размножение, сон или строительство жилья). Такие виды деятельности, как строительство убежища и поиск партнера иногда очень схоже с тем, как это делает медведь. Травоядные животные, как правило, легко находят пищу, но должны есть много. Хищникам же обычно трудно найти пищу, но им не нужно столько есть, поскольку их пища имеет высокую энергетическую ценность. «Окружающая среда» устанавливает структуру питательных веществ, которые можно получить от других организмов которые пригодны для пищи, от геологических ограничений, таких как, реки, горы и погодные условия, и это создает ограничения для получения этой пищи (например, маленькие порции еды могут находиться на большом расстоянии друг от друга). Поиска пищи может быть рискован из-за хищников, жертва может быть подвижной, поэтому ее нужно преследовать, а физиологические характеристики ограничивают его возможности и конечный успех. Теория бактериальной оптимизации объясняется следующими шагами:

- хемотаксис;
- роение;
- воспроизведения;
- ликвидация –рассеивания.

Предлагаемая модификация. На каждом шаге в цикле выбывания-рассеивания, по окончанию самого цикла, мы считаем количество групп методом формального элемента. Каждая группа бактерий представляет собой некоторый объект на изображении, не обязательно окружность. Желаемое количество объектов задается на фазе инициализации. Как только количество

групп становится равным заданному, алгоритм прекращает свою работу. После этого высчитываются средние значения для X , Y и радиуса в каждой группе, эти значения и являются окружностями.

Метод формального элемента. Алгоритм Форель (ФОРмальный Элемент) является примером эвристического дивизимного алгоритма классификации. В основе работы алгоритма Форель лежит использование гипотезы компактности: близким в содержательном смысле объектам в геометрическом пространстве признаков соответствуют обособленные множества точек, так называемые «сгустки». Если расстояние между центром n -го таксона и точкой k этого таксона обозначить S_{nk} , то сумма расстояний между центром и всеми точками k этого таксона будет равна:

$$P_n = \sum_{k=1}^L S_{nk} \quad (1)$$

где, P_n – расстояние между центром n -го таксона и всеми точками этого таксона; S_{nk} – расстояние между центром n -го таксона и точкой k этого таксона. Сумма таких внутренних расстояний для всех n таксонов равна:

$$P = \sum_{n=1}^N P_n \quad (2)$$

Целью работы алгоритма Форель является найти такое разбиение множества объектов на n таксонов, чтобы величина P была минимальной. Работа алгоритма заключается в перемещении гиперсферы определенного радиуса в геометрическом пространстве до получения устойчивого центра тяжести наблюдений, попавших в эту гиперсферу. До начала работы алгоритма признаки объектов нормируются так, чтобы их значения находились между нулем и единицей.

Тесты. Проведем некоторые эксперименты. Будем запускать алгоритм по 10 раз на разных изображениях и высчитываем точность нахождения окружностей, а также среднюю скорость работы. Полученные данные представлены на рисунках 1-2.

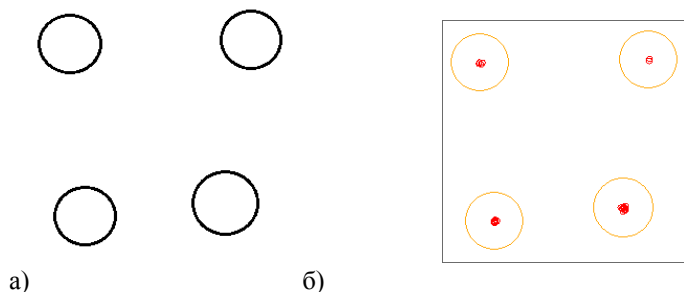


Рисунок 1 – Исходное (а) и результирующее (б) изображение с четырьмя кругами

Точность нахождения: 62.5%. Среднее время 4.45

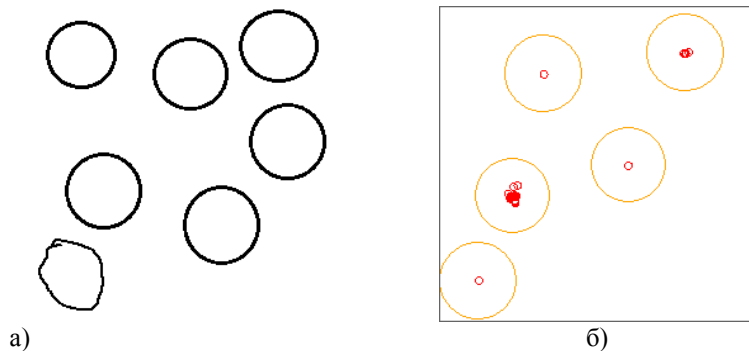


Рисунок 2 – Исходное (а) и результирующее (б) изображение с семью кругами

Выводы. Мы рассмотрели алгоритм бактериальной оптимизации для нахождения и распознавания окружностей, а также различных его модификаций. Провели анализ классического алгоритма с его улучшенными версиями. Затем предложили модификацию для одновременного нахождения окружностей и привели таблицу работы по скорости и по точности. Как видно из таблицы, наша модификация еще не ищет все сто процентов кругов, но делает это значительно быстрее классического алгоритма. Увеличение точности планируется за счет разброса бактерий, не на лучшие позиции, а по всей поверхности исключая уже найденные окружности, для того чтобы бактерии смогли найти другие локальные минимумы.

Список литературы

1. R. Panda, M. K. Naik – A crossover bacterial foraging optimization algorithm // Hindawi Publishing Corporation, vol. 2012, p. 7-14.
2. Chin-Ling Lee, Cheng-Jian Lin – A novel strategy adaptation based bacterial foraging optimization algorithm for numerical optimization// Int J Swarm Intel EvolComput – 2016. –P. 1151-1164.

УДК 004.021

**ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ ОФФЛАЙН-
РАСПОЗНАВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ РУКОПИСНЫХ
СИМВОЛОВ****Н.Г. Соколов, К.А. Ручкин**

Донецкий национальный технический университет, г. Донецк
Кафедра искусственного интеллекта и системного анализа
nick.sokolov@gmail.com

Аннотация

Соколов Н.Г., Ручкин К.А. Обзор существующих методов оффлайн-распознавания последовательностей рукописных символов. Выделена основная задача оффлайн-распознавания рукописных символов. Рассмотрены виды оффлайн-распознавания. Описаны существующие методы и алгоритмы, выполнено их сравнение, выделены достоинства и недостатки.

Ключевые слова: *рукописные символы, методы оффлайн-распознавания.*

Введение. В настоящее время сканирование и сохранение в памяти компьютера текста с твердого носителя является решенной задачей. Это существенно облегчает задачу хранения рукописных и печатных текстов и предоставления доступа к ним различных пользователей. Однако полученный в результате сканирования текст хранится в памяти компьютера в виде изображения, чаще всего растрового, что делает работу с ним весьма сложной: затруднено ориентирование, практически невозможны редактирование, форматирование и поиск по тексту. Перевод изображений рукописного, машинописного или печатного текста в текстовые данные называется распознаванием текста.

В настоящее время существуют высокоточные системы для распознавания машинописных и рукопечатных текстов (например, ABBYY FineReader). Задача распознавания рукописных цифр является одной из классических задач распознавания образов и имеет значительную практическую ценность. Одним из первых практических применений методов распознавания рукописных цифр стало создание системы чтения ZIP-кодов. Методы распознавания цифр применяются также для решения таких практических задач, как чтение банковских чеков, автоматизированное чтение анкет и др.

Постановка задачи. В данной работе рассматриваются и анализируются существующие методы и алгоритмы для распознавания рукописных цифр. Выделяются преимущества и недостатки подходов, а также проводится оценка по результатам анализа, чтобы определить наиболее эффективный метод распознавания.

Этапы оффлайн-распознавания. Распознавание рукописного текста проходит в несколько этапов[1]:

1. Предварительная обработка изображения (preprocessing): на этом этапе происходит обработка изображения с целью повышения его качества и приведения его к виду, удобному для сегментации.

2. Сегментация (segmentation): на этом этапе происходит выделение текста на изображении и его разделение на составные части. Обычно текст обрабатывается иерархически: сначала выделяются отдельные строки, затем отдельные слова, затем символы или части символов.

3. Извлечение признаков (feature extraction): на этом этапе формируются признаковые описания выделенных на этапе сегментации частей.

4. Классификация (classification): на этом этапе по признаковым описаниям, построенным на этапе извлечения признаков, система принимает решение о том, к какому заранее известному классу отнести выделенный на этапе сегментации элемент.

5. Обработка результатов (postprocessing): на этом этапе происходит построение итогового текста по результатам классификации выделенных частей текста.

Методы оффлайн-распознавания. Рассмотрим наиболее популярные методы и алгоритмы для оффлайн-распознавания рукописных текстов.

Алгоритм AdaBoost (сокр. от adaptive boosting) – алгоритм машинного обучения, предложенный Йоавом Фройндом (Yoav Freund) и Робертом Шапиром (Robert Schapire). Является мета-алгоритмом, в процессе обучения строит композицию из базовых алгоритмов обучения для улучшения их эффективности. AdaBoost является алгоритмом адаптивного бустинга в том смысле, что каждый следующий классификатор строится по объектам, которые плохо классифицируются предыдущими классификаторами. AdaBoost вызывает слабый классификатор в цикле. После каждого вызова обновляется распределение весов, которые отвечают важности каждого из объектов обучающего множества для классификации. На каждой итерации веса каждого неверно классифицированного объекта возрастают, таким образом новый классификатор «фокусирует своё внимание» на этих объектах. Экспоненциальная функция потерь слишком сильно увеличивает веса наиболее трудных объектов, на которых ошибаются многие базовые алгоритмы. Однако именно эти объекты чаще всего оказываются шумовыми выбросами. В результате AdaBoost начинает настраиваться на шум, что ведёт к переобучению [2].

Сверточная сеть LeNet. Одним из вариантов специализированной нейронной сети для распознавания изображений является архитектура сверточной нейронной сети. Основные принципы построения этой архитектуры заключаются в следующем:

1) Сеть в целом является многослойным персептроном с прямыми связями.

2) Нейроны в нескольких первых слоях имеют 2-мерные локальные рецептивные поля.

3) Нейроны-детекторы признаков формируют 2-мерные карты признаков способом, подобным математической операции свертки (convolution). Над картами признаков выполняется процедура уменьшения пространственной разрешающей способности.

4) Входом первого слоя нейронов-детекторов признаков является входное изображение. Входом каждого из следующих слоев детекторов признаков являются карты признаков с уменьшенным пространственным разрешением, сформированные предыдущим слоем детекторов признаков [3].

В сети LeNet5 первый слой насчитывает 6 нейронов с квадратным рецептивным полем 5×5 . Выходы нейронов первого слоя формируют первую карту признаков с размерностью $6 \times 28 \times 28$. Выходом первого слоя группировки признаков является карта признаков с размерностью $6 \times 14 \times 14$. Второй слой детекторов признаков содержит 16 нейронов и формирует карту признаков с размерностью $16 \times 10 \times 10$. Нейроны этого слоя имеют рецептивные поля 5×5 и получают входные данные с первой карты признаков. Второй слой группировки признаков также уменьшает вдвое размер карты признаков, соответственно на его выходе вторая карта признаков имеет размерность $16 \times 5 \times 5$. Эта карта признаков является входом 100 нейронов следующего слоя. Последний слой имеет 10 нейронов – по одному на каждый класс, который должен распознаваться.

Метод опорных векторов (англ. SVM, support vector machine) – набор схожих алгоритмов обучения с учителем, использующихся для задач классификации и регрессионного анализа. Принадлежит семейству линейных классификаторов и может также рассматриваться как специальный случай регуляризации по Тихонову. Особым свойством метода опорных векторов является непрерывное уменьшение эмпирической ошибки классификации и увеличение зазора, поэтому метод также известен как метод классификатора с максимальным зазором[4].

Основная идея метода – перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей 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[5].

Сравнение программных средств. Сравнение приведенных и наиболее популярных алгоритмов представлено в табл. 1. Эта таблица показывает основные реализованные алгоритмы, которые используются для оффлайн-распознавания рукописных цифр.

Таблица 1. Итоговые данные по частоте ошибок

	Алгоритм AdaBoost	Сверточная сеть LeNet	Метод опорных векторов (SVM)
Частота ошибок (проценты)	1,6	0,9	0.56
Время выполнения (мс/цифра)	10	30	200
Потребность в памяти (Мбайт)	0,49	0,012	11
Время обучения (суток)	7	14	10

Алгоритмы, которые имеют наиболее меньший процент ошибок требуют больше времени на обработку цифры, также требуют больше времени на обучение и большую потребность в необходимой памяти. Каждый из алгоритмов, приведенных в табл. 1. имеет свои достоинства и недостатки перед аналогами. Однако главным преимуществом является работа в оффлайн-режиме распознавания.

Выводы. В данной статье была рассмотрена основная задача оффлайн-распознавания рукописных символов. Выделены основные проблемы при распознавании, также рассмотрена разница между распознаванием рукописных символов и рукописных цифр. Были описаны существующие методы оффлайн-распознавания и выделены 3 самых известных алгоритма. У каждого из алгоритмов описана основная структура работы, выделены достоинства и недостатки. Были выведены итоговые данные по частоте ошибок описанные методов распознавания.

Список литературы

1. Mansi Shah и Gordhan B Jethava. «A literature review on hand written character recognition». В: Indian Streams Research Journal 3.2 (2013), с. 1 19.
2. Yoav Freund и Robert E Schapire. «A desicion-theoretic generalization of on-line learning and an application to boosting». В: Computational learning theory. Springer. 1995, с. 23 37.
3. Лукович В.В. Простая архитектура сверточной нейронной сети для распознавания рукописных цифр / В.В. Лукович // Кибернетика и вычислительная техника. – 2013.
4. Машина опорных векторов [электронный ресурс] // MachineLearning.ru: [сайт]. URL: <http://www.machinelearning.ru/wiki/index.php?title=SVM>
5. Chih-Chung Chang и Chih-Jen Lin. «LIBSVM: A library for support vector machines». В: ACM Transactions on Intelligent Systems and Technology 2 (3 2011).

УДК 004.021

ИССЛЕДОВАНИЕ И РАЗРАБОТКА АЛГОРИТМА ПОСТРОЕНИЯ ДЕРЕВЬЕВ ШТЕЙНЕРА ДЛЯ СЛУЧАЯ ОРТОГОНАЛЬНОЙ МЕТРИКИ В ТРЁХМЕРНОМ ПРОСТРАНСТВЕ

М.С. Нечепуренко, А.И. Ольшевский

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
nechepurenko@mail.ua

Аннотация

Нечепуренко М.С., Ольшевский А.И. Исследование и разработка алгоритма построения деревьев Штейнера для случая ортогональной метрики в трёхмерном пространстве. Выполнен анализ распространенных форматов изображений растровой и векторной графики. Определены правила выбора наилучшего графического формата при разработке изображения.

Ключевые слова: дерево Штейнера, точка Штейнера, терминал.

Постановка проблемы. Задачи отыскания минимального остовного дерева и кратчайшей сети решались в применении к планированию топологии телефонных сетей, трубопроводов и шоссежных дорог. Решения, приближённые или точные, помогают спланировать геометрию сети и подсчитать необходимые количества материалов.

В более сложных формулировках задачи Штейнера можно учитывать такие факторы, как необходимость избегания определённых географических свойств местности, а также отыскивать кратчайшие соединения между узлами уже существующих сетей.

Возможно, наиболее важным практическим применением задачи Штейнера является конструирование интегральных электронных схем. Более короткая сеть проводящих линий на интегральной схеме требует меньшего времени зарядки-разрядки по сравнению с более длинной сетью и повышает, таким образом, быстродействие схемы. Однако задача отыскания кратчайшей сети на интегральной схеме имеет другую геометрию, так как проводники на ней обычно проходят лишь в двух направлениях — горизонтальном и вертикальном.

Для решения этих проблем необходимо выполнить следующее:

- проанализировать существующие методы построений деревьев Штейнера в евклидовой метрику;
- сравнить результаты различных методов по различным параметрам;

– на основании существующих методов разработать алгоритм для построения деревьев Штейнера в ортогональной метрике.

Цель статьи– провести анализ существующих методов построения деревьев Штейнера и определить их эффективность по различным параметрам.

Общая постановка задачи. В общем случае задача строится следующим образом. Пусть дан граф $G = (V, E)$, где множество вершин V состоит из двух непересекающихся множеств, множества терминальных вершин A и множества точек Штейнера S . Всюду в дальнейшем будем считать, что $|V| = p = n + s$, $|A| = n$, $|S| = s$, $|E| = m$. Ребрам графа приписаны неотрицательные веса, а под длиной дерева понимается сумма весов, входящих в это дерево ребер. Задача состоит в нахождении такого подграфа G , покрывающим все терминальные вершины, и имеет минимальную длину среди всех подобных подграфов.

Задачу Штейнера невозможно решить, просто рисуя линии между заданными точками. Для решения необходимо добавить новые точки, называемые точками Штейнера и служащие в качестве узлов искомой кратчайшей сети. Задача с тремя точками и задача Штейнера для многих точек имеют много общих свойств. Их решения, имеющие вид дерева, характерны тем, что при удалении любого отрезка из кратчайшей сети мы должны будем исключить одну из заданных точек. По этой причине графические решения задачи с тремя точками и задачи со многими точками называются деревьями Штейнера. Отрезки прямых называются рёбрами, а точки называются точками Штейнера.

Задача Штейнера для трёх точек даёт также некоторую информацию о геометрии кратчайших деревьев Штейнера. Во-первых, каждый угол равен 120° или больше, а это означает, что каждая точка соединяется с остальным деревом не более чем тремя рёбрами. Во-вторых, в каждой точке Штейнера сходятся ровно три ребра, образуя друг с другом углы, в точности равные 120° . В-третьих, число рёбер дерева всегда на единицу меньше суммарного числа заданных исходных точек и точек Штейнера. И наконец, последнее свойство: поскольку в каждой точке Штейнера сходятся ровно три ребра и по крайней мере одно ребро должно касаться каждой из заданного множества точек, максимальное число точек Штейнера для любой задачи на две меньше, чем число заданных исходных точек.

Анализ алгоритмов. Задача впервые рассмотрена в работе Прима, в которой предложен точный алгоритм синтеза сети минимальной стоимости.

Первоначально рассмотрим исходное множество несвязанных узлов (вершин) $X = \{x_i\}$.

1. Выбираем произвольный узел (подграф) x_i и отыскиваем стоимость ввода ребра (i, j) , связывающего x_i с некоторым подграфом $x_j c_{ij}$. Если подграфы x_i и x_j состоят из нескольких узлов, то отыскиваем ребро, связывающее ближайшую пару узлов x_i и x_{i1} , принадлежащих к разным подграфам.

2. Среди всех пар (i, j) находим такую (i^*, j^*) , что $c_{i^*j^*} = \min c_{ij}$.

3. Объединяем подграфы в один.

На этом одна итерация метода заканчивается. Таким образом, на каждой итерации число изолированных подграфов сокращается. Как только их число N станет равно 1, работа алгоритма заканчивается.

Доказано, что данный алгоритм позволяет получить оптимальное решение. Заметим, что в задаче Прима не вводятся ограничения на пропускные способности сети, и поэтому отсутствует ограничение на суммарный поток f_{Π} , передаваемый по произвольной ветви (r, l) .

Таким образом, алгоритм Прима позволяет синтезировать кратчайшее связывающее дерево (КСД) без ограничений. Практически гораздо более важной является задача синтеза КСД с ограничениями на суммарный поток f_{Π} , определяемый пропускными способностями кратчайшей сети d_{Π} .

Одним из наиболее известных и распространенных алгоритмов является алгоритм Исау–Вильямса, известный под названием CNDP. Предположим, что центр обработки данных расположен в пункте x_1 , а пропускная способность всех ветвей одинакова и равны d .

Пусть первоначально имеется некоторое множество изолированных узлов $X = \{x_1, \dots, x_n\}$. Для каждого узла i вычисляем стоимость его подключения к центру $c_{i1} = v_i$, а также стоимость связи c_{ij} двух узлов i и j между собой. В общем случае $c_{ij} = c_{ij}(h_i)$, где h_i – поток информации в узле i .

Вычисляем экономию от подключения узла i к j вместо подключения его к РЦ: $E_{ij} = c_{ij} - c_{i1} = c_{ij} - v_i$. Находим такую пару (i^*, j^*) , для которой $E_{i^*j^*} = \min E_{ij}$ при условии, что $h_{i^*} - h_{j^*} \leq d$. Если $E_{i^*j^*} < 0$, то вводим ветвь (i^*, j^*) и объединяем два узла x_{i^*} и x_{j^*} в один: $X_{i^*N} = x_{i^*} \cup x_{j^*}$, при этом определяем новое значение потока информации $N_{i^*} = h_{i^*}$.

Пусть проведено k итераций и построено k обобщенных узлов (подграфов) X_1, X_2, \dots, X_k и пусть $N_k = N(X_k)$ – суммарный информационный поток всех узлов, входящих в X_k .

Опишем $(k+1)$ -ю итерацию. Выбираем произвольный изолированный подграф X_i . Находим произвольный подграф X_j и проверяем возможность ввода ребра (i, j) : $N_i + N_j \leq d$.

Если условие выполняется, то вычисляем $E_{ij} = c_{ij}(N_i) - v_i$.

Находим такую пару (i^*, j^*) , для которой $E_{i^*j^*} = \min E_{ij}$.

Если $E_{i^*j^*} < 0$, то объединяем пару подграфов X_i и X_j в один: $X_{i^*} = X_{j^*} = X_{i^*} \cup X_{j^*}$, в противном случае подключаем все оставшиеся изолированные подграфы напрямую к центру, и конец работы алгоритма.

Алгоритм Краскала отдельно рассматривать не стоит, так как он аналогичен уже описанному. Отличительной особенностью является то, что в начале все узлы изолированы, на каждом шаге отыскивают наименьшую по стоимости допустимую линию связи различных узлов.

В алгоритме Шарма определяют полярные координаты (a_i, r_i) каждого терминала относительно центра. Терминалы (узлы) рассматривают в последовательности, соответствующей возрастанию угла полярных координат a_i , так что $a_1 \leq a_2 \leq \dots \leq a_n$. Строят минимальную по стоимости древовидную сеть терминалов x_1, x_2, \dots, x_k и РЦ, где x_1, x_2, \dots, x_k удовлетворяют ограничениям, но при добавлении x_{k+1} к многопунктовой (многоточечной) сети ограничения нарушаются. Вышеуказанную процедуру, начиная с x_{k+1} повторяют до тех пор, пока все терминалы не будут включены в дерево.

Как следует из описания, все перечисленные алгоритмы близки друг к другу и различаются только очередностью объединения компонент, которая обеспечивается назначением соответствующих весов значимости этим компонентам (узлам).

В Алгоритм Фогеля для каждого терминала (узла) x_i подсчитывают выигрыш E_i как разность $E_i = c'_{ij} - c_{ij}$, где X_j – ближайший к x_i подграф; X'_j – следующий по порядку ближайший к x_i подграф.

Текущая итерация алгоритма заключается в том, что находится i^* такой, что $E_{i^*} = \max E_i$. Проводится проверка по ограничению: если оно выполняется, то x_i подключается к ближайшему узлу x_j . В результате образуется некоторый подграф $X_i = x_i \cup x_j$. Стоимость связи между двумя сегментами определяется как стоимость самой дешевой связи между двумя терминалами (узлами), принадлежащими разным подграфам. Если связь (i, j) нарушает ограничение по ПС, то $c_{ij} = \infty$. Когда все терминалы оказываются подключенными к РЦ, конец работы алгоритма.

Унифицированный алгоритм. В результате анализа известных алгоритмов (алгоритмы Прима, Исау – Вильямса, Краскала, Шарма и т. д.) предложен так называемый унифицированный алгоритм, из которого можно получить как частный случай любой из известных алгоритмов построения деревьев Штейнера, приписав определенные значения некоторым формальным параметрам.

Для формального описания алгоритма введем следующие обозначения: v_i – вес терминала i ; X_i – подграф (набор терминалов), содержащий терминал i ; (i, j) – линия, соединяющая терминалы i и j ; E_{ij} – экономия, соответствующая линии (i, j) ; c_{ij} – стоимость связи (i, j) ; N – число терминалов.

Шаг 0.

1. Задаем начальные значения величины v_i для $i = 1, 2, \dots, N$, используя соответствующее правило из таблицы 2.1.

2. Устанавливаем $X_i = \{i\}$, $i = 1, 2, \dots, N$.

3. Определяем $E_{ij} = c_{ij} - v_i$ для (i, j) , если c_{ij} существует и терминалы (узлы) i и j не объединены.

Шаг 1.

Определяем $E_{i^*j^*} = \min E_{ij}$. Если $E_{i^*j^*} = \infty$, то заканчиваем поиск, в противном случае переходим к шагу 2.

Шаг 2.

Оцениваем выполнение ограничений для $X_{i^*} \cup X_{j^*}$ (объединения подграфов). Если любое из них нарушено, то устанавливаем $E_{i^*j^*} = \infty$ и возвращаемся к шагу 1. В противном случае переходим к шагу 3.

Шаг 3.

1. Вводим ветвь (i, j) .

2. Изменяем величины v_i для $i = 1, 2, \dots, N$.

3. $E_{ij} = c_{ij} - v_i$ для тех i , для которых v_i изменено.

4. Сформируем новый подграф $X_{i^*} \cup X_{j^*}$. Повторно пересчитываем ограничения и возвращаемся к шагу 1.

Меняя правило задания и коррекции весов v_i , можно получить любой из алгоритмов, причем одни правила во многих случаях дают лучшие результаты, чем другие. Однако пока не найдено такое универсальное правило, которое бы давало наилучшие результаты во всех случаях. Поэтому применяют параметрический способ определения весов.

Рассмотрим один из таких способов. Запишем веса узлов в виде

$$v_t = a(bc_{i_0} + (1 - b)c_{i_2}), \quad (1)$$

где: c_{i_0} , c_{i_2} – стоимости соединения терминала i с центральным узлом и с его ближайшим соседним терминалом соответственно;

a и b – некоторые константы, причем, $a \geq 0$, $0 \leq b \leq 1$.

При $a = 0$ получаем алгоритм Краскала; при $a = b = 1$ – алгоритм Исау – Вильямса; при $a = 1$, $b = 0$ – алгоритм Фогеля. Наконец, придавая a и b некоторые промежуточные значения, можем получить одновременно свойства всех вышеперечисленных алгоритмов.

Рассмотрим эффективность реализации унифицированного алгоритма построения деревьев Штейнера. Сложность алгоритма – время вычислений и необходимый объем памяти – зависит от размерности графа N .

В общем случае граф должен быть полным, что требует просмотра всех пар узлов. Однако при большом количестве терминалов можно без значительного снижения точности результатов решать задачу с помощью разреженного графа, в котором каждый терминал связан лишь с некоторым ограниченным числом соседних терминалов.

Например, в сети с 40 узлами достаточно рассматривать соединения каждого терминала с пятью соседними. Дальнейшее увеличение числа терминалов K лишь незначительно улучшает результаты. Для сети терминалов, однородно распределенных вокруг центрального узла, можно игнорировать около 75 % возможных соединений.

Для быстрой оценки приближенного значения K всю область, содержащую терминалы, разбивают на прямоугольники, в пределах каждого из которых находят K ближайших соседей заданного терминала. Благодаря такому правилу число операций для повторного вычисления E_{ij} при включении новой ветви в дерево можно снизить с $N(N-1)/2$ до $N \times K$. Кроме того, учитывая, что $E_{ij} = v_i - c_{ij}$, где c_{ij} – константа, пересчет E_{ij} можно свести к пересчету v_i , т. е. вместо $N \times K$ повторных вычислений E_{ij} достаточно пересчитать N раз значение v_i .

Выводы

Результаты экспериментов показали, что зависимость логарифма времени работы ЭВМ от логарифма числа терминалов имеет линейный характер с наклоном кривой, который свидетельствует о квадратичной зависимости вычислительной сложности унифицированного алгоритма от размерности задачи N . В результате исследований выявлено, что время решения задачи почти не изменяется при переходе от одного правила v к другому и при изменении ограничений, в то же время оно сильно зависит от числа соседних терминалов, подключаемых к данному (т.е. числа новых линий, подключаемых к дереву).

Список литературы

1. Дюк В., Самойленко А. Data Mining: учебный курс. – СПб: Питер, 2001. – 386с.
2. Калашников Р.С. Построение дерева Штейнера методом генетического поиска // Перспективные информационные технологии и интеллектуальные системы. – 2005. – № 2 (22).
3. Курейчик В.М. Генетические алгоритмы. – Таганрог: изд-во ТРТУ, 1998. – 242 с.
4. Маршалл У. Берн, Рональд Л. Грэм Поиск кратчайших сетей. // Scientific American (издание на русском языке). – 1989. – № 3. – С. 64–70.
5. Панченко Т.В. Генетические алгоритмы: Учебно-методическое пособие / под ред. Ю.Ю. Тарасевича. – Астрахань: АГУ, 2007. – 87 с.
6. Рыженко Н.В. Алгоритм построения минимальных связывающих деревьев с дополнительными вершинами (деревьев Штейнера) для случая прямоугольной метрики. Труды ИМВС РАН, 2002.

УДК 004.93'14

ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ ЯЗЫКА ЖЕСТОВ

Потёмкин А.С., Миненко А.С.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
E-mail: airgunner6@gmail.com

Аннотация:

Потёмкин А.С., Миненко А.С. Использование компьютерного зрения для распознавания языка жестов. В работе рассмотрены использование компьютерного зрения для распознавания языка жестов используемого глухими людьми. Подробно рассмотрена основная классификация методов и подходов распознавания жестов, в которую входят такие виды как захват движения на основе маркеров, отслеживание рук с помощью оборудованной перчатки, визуальные методы распознавания жестов.

Ключевые слова: сурдоперевод, язык жестов, компьютерное зрение, визуальные методы распознавания жестов.

Постановка проблемы. Во время общения людям свойственно произвольное жестикулирование, однако есть такая категория людей лишенных слуха, для которых язык жестов является единственным способом общения [1]. Для начала, разделим два понятия поза руки и жест. Под позой руки понимается ее статическое положение. Жестом же называют телодвижение, преимущественно движение рукой, сопровождающее речь или имеющее значение какого-либо сигнала, знака. Язык жестов в широком смысле бесконечно разнообразен как по форме жестов, так и по их семантике, поэтому полномасштабное распознавание языка жестов представляет собой сложную задачу даже для человеческого интеллекта. На современном уровне развития систем компьютерного зрения задача автоматического распознавания обычно ставится применительно к небольшому набору заранее определенных жестов [2].

Методы распознавания жестов. Все методы и подходы распознавания жестов обычно классифицируют по типу используемых входных данных и сенсоров для восприятия руки [3]. В основном используют такую классификацию распознаванию жестов:

1. захват движения на основе маркеров;
2. отслеживание рук с помощью оборудованной перчатки;
3. визуальные методы распознавания жестов.

Захват движения на основе маркеров. В основе маркерной системы лежит специальное оборудование. На руку человека прикрепляются датчики;

данные движения рук с датчиков фиксируются камерами и поступают в компьютер, где сводятся в единую трёхмерную модель, точно воспроизводящую жесты человека, позже (или в режиме реального времени) на основе модели создаётся анимация жестов человека. Подобные системы, основанные на захвате движения по маркерам, ориентированы на точность за счет простоты использования и установки. В свою очередь недостатком систем, основанных на захвате движения по маркерам, является использование навязчивых светоотражающих или цветных маркеров, или светодиодов и дорогих многокамерных установок.

Отслеживание рук с помощью оборудованной перчатки. В основе такого метода отслеживания рук лежат оборудованные перчатки, которые основаны на экзоскелетах, позволяющих повторять биомеханику человеческой руки, и оснащены более чем десятком датчиков (например, акселерометром, сенсорами, передающими тактильные, температурные и вибрационные ощущения, датчиками давления и др.). Информация о движении рук считывается с датчиков и передается на обработку в ПК. Полученная информация затем анализируется и преобразовывается в символы. Данные о характеристиках отдельных жестов используется системой в качестве шаблонов.

Не смотря на то что системы, имеющие в основе оборудованные перчатки, обеспечивают высокую точность отслеживания рук и распознавания жестов, имеется также существенные недостатки. Во-первых, системы подобного рода являются дорогостоящими, во-вторых применяющиеся перчатки очень громоздки, что ограничивает в движении.

Визуальные методы распознавания жестов. Визуальные методы распознавания жестов являются перспективной областью исследований. Отслеживание рук основано на методах, работающих исключительно с визуальной информацией и не предъявляющих специальных требований к оснащению пользователя дополнительным оборудованием. Выделения контуров и силуэтов являются наиболее распространенными характеристиками, используемыми для идентификации позы руки. Пока эти сигналы являются общими и устойчивы к различным условиям освещения и, исходя из этого, требуют больших вычислительно сложных определяющих алгоритмов, которые ищут многомерные позиции пределов руки [4]. В свою очередь визуальные методы распознавания жестов можно разделить на три большие категории.

К первой относятся методы, которые основаны на восстановлении полной модели кисти с 27 степенями свободы по входному изображению. В данном подходе, рука представляется как соединение составляющих частей, и каждое состояние называется «степенью свободы». Степени свободы определяют позиции и ориентации реального объекта. Моделирование руки с учетом всех ее степеней свободы представляет собой сложную задачу. Поэтому

используется упрощенная модель руки с 27 степенями свободы, ставшая неофициальным стандартом, пример такой модели изображен на рисунке 1. Системы отслеживания руки в пространстве на основе 3D модели руки используют заранее созданные базы данных всех известных конфигураций руки и их визуализации. Каждая конфигурация содержит множество углов наклона суставов и параметры позы руки для описания состояния руки.

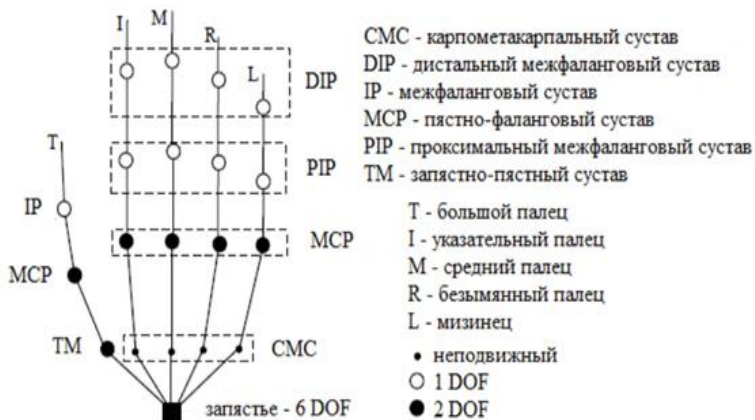


Рисунок 1 – Модель руки с 27 степенями свободы

Теоретически методы основанные на модели руки с 27 степенями свободы являются наиболее перспективными, так как они подразумевают полное оценивание позы и динамики руки. Основными ограничениями подобных методов являются большая вычислительная сложность и ограниченная точность восстановления модели руки из-за наличия окклюзий, что делает невозможным их применение на практике.

Ко второй категории относятся статистические методы, которые вместо восстановления полной модели руки предлагают построение признакового описания входного изображения и дальнейшую классификацию жестов именно по этому описанию [5]. Решающую роль в методах подобного рода играет выбор подходящих характеристических признаков. Так для статических поз руки, можно использовать геометрические признаки, такие как кончики пальцев, направление пальцев, контур руки, а также негеометрические признаки (цвет кожи, форма, текстура и др.). Однако эти признаки не всегда доступны или адекватны для распознавания из-за взаимного перекрытия элементов руки и/или из-за условий освещения.

Важным моментом в этих подходах является классификатор, который предоставляет возможность разделить (по категориям) различные признаки

или информацию, выделенные из «грубых данных», например, изображения или видеопотока.

Рассмотрим пример распознавания статических жестов языка глухих. Особенность этой работы заключается в том, что для распознавания жестов используется только одна видеопоследовательность, полученная с веб-камеры. Полученное с камеры RGB изображение конвертируется в двоичное изображение «binary image». Затем к двоичному изображению применяется техника обнаружения краев «edge detection» полученный результат продемонстрирован на рисунке 2.



Рисунок 2 – Результат обнаружения краев

После того, как края обнаружены, необходимо определить кончики пальцев. Для этого предлагается использовать Вейвлет преобразование, позволяющее определить пики. Жест сравнивается с эталонными жестами в базе данных, имея координаты кончиков пальцев и углы между ними как показано на рисунке 3.



Рисунок 3 – Определенные пики пальцев

К третьей категории относятся метрические методы распознавания жестов. Подобные методы предполагают построение некоторой метрики на множестве входных изображений и выполнение классификации за счет сравнения входного изображения с набором эталонов. Так, например,

предлагается метрика, характеризующая степень сходства скелетов силуэтов ладони, и выполняется классификация жестов с помощью метода ближайшего соседа.

Выводы. Несмотря на то, что в литературе достаточно подробно описаны различные алгоритмы и методы распознавания жестов, в целом данная технология находится на начальном уровне и представляет широкую область для дальнейшей работы и проведения исследования. Можно привести несколько сценариев, где технология распознавания жестов будет перспективна:

- управление развлекательными приложениями и системами;
- управление медицинским оборудованием при требованиях стерильности, которые можно удовлетворить за счет минимизации тактильного контакта между оператором и устройством;
- визуализация и моделирование сложных трехмерных данных, удобство работы с которыми ограничено при применении стандартных двухмерных манипуляторов, таких как компьютерная мышь;
- компьютерное распознавание языка глухонемых.

В настоящее время технологии отслеживания рук используются в современных телевизорах. Такие телевизоры улавливают движения человека при помощи инфракрасных датчиков, расположенных около экрана. Они фиксируют движения рук и преобразуют их в понятный для телевизора сигнал. Пока телевизоры понимают лишь несколько жестов. В частности, подъем руки вверх означает включение и выключение телевизора. Для переключения каналов достаточно провести рукой из стороны в сторону. Громкость регулируется круговыми движениями.

Список использованных источников

1. Popovic J., Wang R. Y. Real-time hand-tracking with a color glove //ACM Transactions on Graphics (TOG). ACM, 2009. – Т. 28. – №. 3. – С. 63.
2. Понс Ж., Форсайт Д. Компьютерное зрение: современный подход. М.: – С.- П.; – К: Вильямс, 2004. – 926 с.
3. Aran O. Vision based sign language recognition: modeling and recognizing isolated signs with manual and non-manual components : дис. Bogazici University, 2008.
4. Куракин А.В. Распознавание динамических жестов в системе компьютерного зрения на основе медиального представления формы изображений: автореф. дис. на соискан. учен. степ. канд. технич. наук. Москва: Московский государственный университет имени М. В. Ломоносова, 2012.
5. Нагапетян В. Э. Методы распознавания жестов руки на основе анализа дальностных изображений: РУДН, 2013.

УДК 004.4'275

ОБЗОР РАСПРОСТРАНЕННЫХ ИГРОВЫХ ДВИЖКОВ ДЛЯ СОЗДАНИЯ 2D ПРИЛОЖЕНИЙ

В.С. Вивденко, Д.М. Бочаров

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
hectovlad@yandex.ru

Аннотация

Вивденко В.С., Бочаров Д.М. Обзор распространенных игровых движков для создания 2D приложений. Рассмотрены наиболее распространенные игровые движки, ориентированные на создание 2D приложений. Выделены их особенности. Определены движки, наиболее подходящие для создания игрового Windows-приложения в жанре 2D-платформер «Space One».

Ключевые слова: программа, игра, игровое приложение, игровой движок, 2D.

Постановка проблемы. Задействованный игровой движок является определяющим фактором при разработке игры. Однако, выбор подходящей платформы для разработки среди множества вариантов является сложным вопросом. Различные игровые движки предлагают разные возможности, которые лучше всего подойдут для выполнения конкретных задач, но могут быть не востребованы или стать препятствием для каких-либо других. Таким образом, выбор оптимального игрового движка требует учета множества факторов, решающими среди которых являются:

- цели и задачи проекта, его особенности;
- приоритеты, на которые будет опираться процесс разработки;
- целевая платформа;
- целевая аудитория;
- знания и опыт персонала, их возможности;
- доступный бюджет.

Цель статьи – провести анализ распространенных игровых движков, наиболее подходящих для создания дебютного игрового приложения, в жанре 2D-платформер для операционной системы Windows разработчиком-одиночкой без опыта разработки игр и с ограниченными финансовыми возможностями.

Актуальность исследования. На сегодняшний день для начинающего разработчика компьютерных игр предоставлен широкий выбор инструментов, которые способны значительно упростить процесс реализации всевозможных идей для игровых приложений, а также повысить качество конечного

продукта. Однако, выбор нужного инструментария требует серьезного подхода, так как он оказывает большое влияние на весь ход разработки и способен как значительно упростить и сократить процесс, так и сделать его гораздо более трудоемким. В то же время, активное развитие игровой индустрии приводит к повышению заинтересованности в разработке и появлении множества небольших независимых команд-разработчиков или даже разработчиков-одиночек, которые хотели бы попробовать себя в этой отрасли. Именно для них очень остро стоит вопрос о выборе игрового движка для дебютного проекта, по причине нехватки опыта, ограниченного финансирования и широкого ассортимента программ, среди которых сложно сразу определить наиболее подходящую для поставленных задач. В качестве идеи для такого первого проекта зачастую берется простая 2D-игра, например, в жанре «платформер», для самой распространенной игровой платформы Windows. Так что рассматриваемые игровые движки можно оценить по тому, как они подходят для этого концепта [1].

Анализ распространенных игровых движков для создания 2D-игр.

«RPGMaker». Этот движок изначально задумывался как конструктор ролевых 2D-игр в японском стиле (так называемые jRPG). Несмотря на это, его последние версии являются весьма гибкими и позволяют создавать игры и других жанров, которые ориентированы на двухмерное представление. Программа является очень простой в освоении и позволяет создавать простые игры тем, у кого нет опыта в программировании вообще, т.к. в ней можно создать приложение, не написав ни одной строчки кода. В то же время, она обладает гибкими возможностями, позволяя более опытным разработчикам дополнять ее функционал для реализации любых идей. Для этого требуется опыт работы с JavaScript или Ruby [2].

В программе уже заложены наборы ресурсов для создания первой игры, такие как шаблоны карт, стандартные типы интерфейса, генератор персонажей и многое другое. Предусмотрено переключение перспективы с классического вида сверху на вид сбоку. Для создания неклассических jRPG, платформа позволяет подключать ранее созданные расширения. С ее помощью разрабатываются приложения как для Windows, так и MacOS, Android, IOS, а также, в виде кода HTML 5, для Web-браузеров.

«RPGMaker» является коммерческим проектом, цена последней версии продукта составляет 80\$. Кроме этого, существует бесплатная пробная версия, которая имеет ограниченный набор инструментов. Разработчик имеет полные права на игры, созданные на основе «RPGMaker», и может использовать их в коммерческих целях.

«GameMakerStudio». Этот игровой движок рассчитано на разработку 2D приложений, но не привязан к какому-либо конкретному жанру игр. Он обладает набором решений для стандартных игровых задач, но при этом не устанавливает готовые игровые механики – их составляет уже разработчик.

Программа может служить удобным инструментом для быстрого создания прототипов игры или каких-либо отдельных механик, однако в последствии способна самостоятельно создать полностью готовый продукт. Для этого в ней предусмотрен, к примеру, графический редактор, ориентированный именно на создание игровых текстур, спрайтов и анимации [3].

Движок использует свой собственный язык программирования GML, который основан на принципах JavaScript и C++, так что знание какого-либо из этих языков поможет в освоении, но не является обязательным. Работать с программой можно и без опыта программирования.

«GameMakerStudio» распространяется бесплатно для некоммерческого использования. Для получения права продавать созданные с ее помощью игры, необходимо приобрести лицензию, которая стоит 150\$. Кроме того, она позволит создавать кроссплатформенные приложения, и откроет некоторые возможности для удобной командной работы.

Среди успешных проектов, созданных с помощью Game Maker Studio, можно выделить «Hotline Miami», «Hyper Light Drifter», «Gods Will Be Watching».

«Construct 2». Легкий в освоении движок, который позволяет создавать игры на HTML5 без написания кода. Интуитивно понятный интерфейс позволяет вести разработку без опыта в программировании, путем перетаскивания и расположения объектов и событий в рабочей области.

Программа рассчитана на создание 2D приложений и имеет встроенный набор инструментов, стандартных для жанра «платформер», как например принципы перемещения персонажа и его взаимодействия с объектами. Особенностью является возможность в реальном времени следить за ходом выполнения игры, одновременно внося в нее правки[4].

Присутствует возможность расширять функционал программы с помощью расширений и плагинов. Движок позволяет разрабатывать игры для всех популярных платформ, среди которых есть Windows, Android, IOS, MacOS, а также имеет встроенную интеграцию с крупными сайтами для публикации игр, таких как Facebook, Newground, ChromeWebStore.

Бесплатная версия «Construct 2» позволяет создавать приложения только для платформы Windows, а также имеет ограничение на коммерческое использование. Полная лицензия продукта со всеми возможностями стоит 130\$.

«LÖVE». Это простой в использовании, кроссплатформенный фреймворк для разработки 2D игр любого жанра. В качестве языка программирования используется lua. Обладает базовым набором функций, которые могут понадобиться при разработке 2D игры, а именно: поддержка устройств ввода (в том числе контроллеров), физика (на основе Box2D), работа с таймерами, поддержка многопоточности, загрузка аудио файлов и изображений, поддержка шрифтов и многое другое. Движок не имеет собственной среды

разработки (IDE), так что для работы с ним потребуется какой-либо текстовый редактор (например, Notepad++) [5].

В отличие от предыдущих рассмотренных примеров, этот игровой движок не является «конструктором» для создания игр, и требует понимания основ программирования и знания языка lua.

Особенностью «LÖVE» является его открытость и доступность. Исходный код программы находится в свободном доступе, его можно найти на официальном сайте. А это означает, что он открыт для пользовательских модификаций, который могут помочь разработчику в выполнении задач, которые уже были решены кем-либо ранее, либо подобным им. Также эта особенность собрала вокруг движка крупное сообщество, которое готово делиться советами и наработками, что очень помогает новичкам. И, как следствие, «LÖVE» полностью бесплатен, начать разрабатывать с его помощью может любой желающий в любой момент, как для личного, так и для коммерческого использования.

Выводы. В ходе исследования был проведен обзор нескольких наиболее распространенных игровых движков, которые подходят для создания 2D-платформера для ОС Windows. Были выделены их отличительные черты, особенности и достоинства, которые как можно точнее подходят под заданные требования проекта. По полученным результатам можно выбрать игровой движок, который будет наиболее эффективен при разработке игрового приложения с учетом его особенностей и возможностей команды разработчиков, а также финансирования проекта.

Результаты исследования планируется использовать при выборе игрового движка для создания игрового Windows-приложения в жанре 2D-платформер «SpaceOne».

Список литературы

1. Все программы для создания 2D игр на компьютер[электронный ресурс] // Gameisart.ru: [сайт]. [2009-2017]. URL: http://gameisart.ru/game_dev_programms_2d.html
2. Make your own game with RPGMaker[электронныйресурс] // RPGMaker: [сайт]. [2017]. URL: <https://www.rpgmakerweb.com/>
3. Сравнительный обзор Game Maker Studio 1.4 [электронныйресурс] // RPGMaker.ru: [сайт]. [2015]. URL: <http://rpgmaker.ru/forum/Game-Maker/60617-sravnitelnyj-obzor-game-maker-studio-1-4>
4. Make your own 2d games with Construct 2[электронныйресурс] // Scirra: [сайт]. [2017]. URL: <https://www.scirra.com/construct2>
5. LÖVE – Free 2DGameEngine[электронныйресурс] // LÖVE: [сайт]. [2017]. URL: <https://love2d.org/>

УДК 004.931

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

В.В. Ткаченко, К.А. Ручкин

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
sam_dntu@mail.ru, vlad.os.011235@gmail.com

Аннотация

Ткаченко В.В., Ручкин К.А. Исследование алгоритмов обнаружения объектов на изображении. Выполнен анализ распространенных алгоритмов обнаружения объектов на изображении. Проведено сравнение алгоритмов между собой.

Ключевые слова: базовая симметрия, преобразование Хафа, случайная выборка, генетические алгоритмы.

Введение. Алгоритмы обработки изображений сталкиваются с большими трудностями, если изображение содержит шум или искажения. Автоматическое обнаружение объектов на изображении до сих пор остается нерешенной проблемой в области обработки изображений.

Данная работа представляет анализ распространенных алгоритмов для автоматического обнаружения различных объектов, которые находятся на изображении. Существует несколько распространенных подходов, которые традиционно делятся на три категории: базовая симметрия, преобразования Хафа и случайная выборка. Одним из новых направлений в обработке изображений является применение генетических алгоритмов. Рассмотрим алгоритмы обнаружения объектов заданного класса на изображении на примере обнаружения эллипса.

Базовая симметрия. При обнаружении симметрии учитывается геометрия эллипса. Самые распространенные элементы, используемые в геометрии эллипса – центр и ось. С их помощью можно найти эллипс на изображении. Этот метод работает, представляя геометрическую форму своим набором параметров. Пики в контейнерах обеспечивают индикацию, где могут быть эллипсы [1]. Очевидно, что так как параметры квантуются в дискретных бункерах, интервалы бункеров непосредственно влияют на точность результатов и вычислительные затраты. Таким образом, для точного квантования пространства, алгоритм возвращает более точные результаты, в то время как страдает от больших нагрузок памяти и вычислений.

Случайная выборка. Для преодоления проблемы потребления больших объемов памяти и высокой вычислительной потребности, некоторые другие исследователи предложили другие детекторы эллипсов с помощью случайной выборки. В случайной выборке бункер представляет собой форму кандидата, а не набор квантованных параметров. Тем не менее, случайные подходы для отбора кандидатов прошли через процесс накопления признаков в бункерах. Бункер с наибольшим количеством признаков представляет собой наилучшее приближение фактического эллипса в изображении цели [2]. Исследования показывают, что случайная выборка дает улучшение точности и вычислительной сложности, а также сокращает число ложных срабатываний по сравнению с алгоритмом базовой симметрии.

Преобразование Хафа. Данный алгоритм предназначен для обнаружения прямых и кривых линий на полутоновых или цветных изображениях. Метод позволяет указать параметры семейства кривых и обеспечивает поиск на изображении множества кривых заданного семейства. В алгоритме преобразования Хафа используется аккумуляторный массив, размерность которого соответствует количеству неизвестных параметров в уравнении семейства искомых кривых. Аккумуляторный массив позволяет определить параметры бесконечно протяжённых прямых или кривых, но с его помощью невозможно определить, где именно начинаются и заканчиваются отрезки этих линий [3]. В преобразовании Хафа используется процедура голосования, которая применяется к пространству параметров, из которого и получают объекты определённого класса фигур по локальному максимуму аккумуляторном массиве. Классический алгоритм преобразования Хафа связан с идентификацией прямых в изображении, но позже алгоритм был расширен возможностью идентификации позиции произвольной фигуры, чаще всего эллипсов и окружностей.

Преобразование Хафа эффективно только при значительном количестве «попаданий» в соответствующий элемент пространства Хафа, только тогда можно с уверенностью определить фигуру, пренебрегая фоновым шумом. Это значит, что размер элемента не должен быть очень маленьким, иначе некоторые значения попадут в соседние элементы, уменьшая видимость нужного элемента. Эффективность алгоритма в большой степени обусловлена качеством входных данных: границы фигур на этапе предобработки изображения должны быть четко определены. Использование преобразования Хафа на зашумленных изображениях затруднено. Для зашумленных изображений необходим этап предобработки с целью подавления шума.

Генетические алгоритмы. В качестве альтернативы традиционным методам, проблема обнаружения объектов заданного класса на изображении также была реализована с помощью методов оптимизации. Они

продемонстрировали лучшие результаты, чем те, которые основаны на преобразовании Хафа и случайной выборки по отношению к точности и надежности [4]. Такие подходы произвели несколько надежных детекторов объектов заданного класса с использованием различных алгоритмов оптимизации такие как генетические алгоритмы. Генетический алгоритм – это алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искоемых параметров с использованием механизмов, аналогичных естественному отбору в природе, таких как наследование, мутации, отбор и кроссинговер. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов [5].

У алгоритмов обнаружения, основанных на генетических алгоритмах, присутствует несколько преимуществ по сравнению с традиционными подходами. Функция оценки каждого решения подсчитывает количество пикселей, содержащихся внутри объекта, метод склонен производить ложные обнаружения, особенно для изображений, содержащих шум или искажения.

Выводы. В ходе анализа различных алгоритмов обнаружения объектов заданного класса, были рассмотрены такие алгоритмы как: случайная выборка, базовая симметрия, преобразование Хафа, генетические алгоритмы. Перспективным направлением в области обработки изображений можно считать генетические алгоритмы, так как данное направление все еще не дает высокой точности обнаружения объектов. Целесообразно проводить исследования по оптимизации и улучшению существующих, а также созданию качественно новых генетических алгоритмов для обнаружения объектов заданного класса на изображениях.

Список литературы

1. Van Aken, J. R. (2005). Efficient ellipse-drawing algorithm. *IEEE Computer Graphics and Applications*, 4(9), 24 – 35.
2. Gonzalez, R. C., & Woods, R. E. (1992). *Digital image processing*. Reading, MA: Addison Wesley.
3. Muammar, H., & Nixon, M. (1989). Approaches to extending the Hough transform. In *Proceedings International Conference on Acoustics, Speech and Signal Processing ICASSP- 89, Glasgow (Vol. 3, pp. 23 – 26, 1556 – 1559)*, May 1989.
4. Ayala-Ramirez, V., Garcia-Capulin, C. H., Perez-Garcia, A., & Sanchez-Yanez, R. E. (2006). Circle detection on images using genetic algorithms. *Pattern Recognition Letters*, 27, 652–657.
5. Storn, R., & Price, K. (1995). Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report No. TR-95-012, International Computer Science Institute.

УДК 004.42

ПРИМЕНЕНИЕ АЛГОРИТМОВ ПРОГРАММНОГО РАСПОЗНАВАНИЯ АВТОМОБИЛЬНЫХ НОМЕРОВ С ИСПОЛЬЗОВАНИЕМ ANDROID- УСТРОЙСТВА

Яременко П.А.

Донецкий национальный технический университет
кафедра компьютерной инженерии
e-mail: pavel.yaremenko@gmail.com

Аннотация

Яременко П.А. Применение алгоритмов программного распознавания автомобильных номеров с использованием Android-устройства. Рассмотрена проблема подбора оптимального алгоритма распознавания текста в контексте специализированного ПО для мобильных устройств на базе Android. Рассмотрено сравнение различных алгоритмов распознавания, в частности уделено внимание библиотеке OpenCV. Предложена модель клиент-серверной работы приложения как способ решения проблемы ограниченной памяти устройства.

Ключевые слова: Android-устройства, клиент-сервер, автотранспорт, распознавания автомобильных номеров.

Общая постановка проблемы

В условиях роста составляющей частного автотранспорта в дорожном трафике крупных городов всё чаще и чаще возникает необходимость контроля правомерного поведения участников дорожного движения, а также периодического контроля и проверки используемого участниками дорожного движения автотранспорта на предмет угона или каких-либо нарушений.

Сотрудниками МВД для подобного вида мониторинга может быть использовано специализированное программное обеспечение для Android-устройств, что обеспечило бы не только простоту в эксплуатации, но и универсальность, удобство в обслуживании клиентской системы. Подобной системой могла бы послужить связка из утилиты для мобильных устройств на базе Android и сервера, обрабатывающего запросы от данной утилиты. При этом, разумеется, подразумеваем наличие мобильного интернет-подключения, достаточного для передачи данных с устройства на сервер и наоборот.

Решая данную задачу, в первую очередь необходимо продумать алгоритмы и способы организации распознавания текста автомобильного номера на первом этапе – на уровне пользователя. Здесь и должно быть сосредоточено основное внимание разработчика данного программного обеспечения.

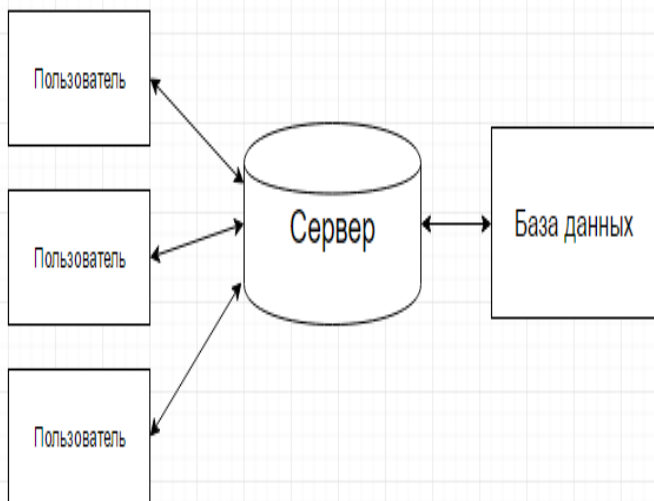


Рис. 1. Общий принцип простого клиент-серверного взаимодействия.

Помимо алгоритмов работы с первоисточником входных данных (в данном случае таковым выступает автомобильный номер) также следует рассмотреть принципы организации работы сервера и структуру базы данных, с которой серверная машина должна осуществлять взаимодействие для максимально быстрого ответа устройствам-пользователям. Данная задача будет решаться при дальнейшем развитии темы в последующих работах.

Средства для выполнения поставленной задачи

Разбиение общей задачи на три глобальных составляющих позволяет более эффективно проработать структуру проекта. Одной из наиболее интересных и важных является составляющая, скрытая под пользовательской частью. Она и будет наиболее подробно рассмотрена в данной статье.

Итак, дабы осуществить распознавание автомобильного номера необходимо пройти несколько этапов алгоритма: предварительный поиск области для распознавания, нормализация распознаваемого участка изображения, непосредственно распознавание текста.

Алгоритмы предварительного поиска

Выделение границ и их анализ может осуществляться различным образом. Самый простой и очевидный – поиск прямоугольного контура.

Реализуется это, как правило, посредством выделения точек цифрового изображения, в которых резко изменяется яркость. Слабое место данного способа – наличие любых помех (номер загорожен чем-либо и т.д.). В случае наличия помех алгоритм не является надежным.

Более продвинутый способ захвата области распознавания связан с гистограммой направленных градиентов, а точнее – с ее несколько упрощенной формой. Следуя этому принципу, анализируется только часть номерной рамки. Выделяются все контуры на изображении, после чего по алгоритму ведется поиск всех вертикальных прямых, расположенных друг от друга на условном расстоянии N . Для любых таких вертикальных прямых с заданным отношением N к расстоянию между ними можно условно полагать, что между данными прямыми и находится целевой для распознавания текст.

Также популярным при поиске решения аналогичных задач является метод, связанный с анализом гистограмм изображения. Основанием методу служит допущение, что частотная характеристика участка изображения с номерным знаком отлична от частотной характеристики окрестности. Однако данный метод имеет очень «болезненный» изъян: кадр для анализа должен быть подобран таким образом, чтобы размеры машины полностью совпадали с размером кадра. Причина тому – вероятность содержания на фоне различных сторонних надписей, табличек и т.д., что может также восприниматься данным алгоритмом как потенциальный номерной знак.

Все описанные выше методы имеют ряд недостатков, которые, очевидно, совершенно не сопоставимы с работой в реальных условиях: с наличием грязи, пыли, помех, плохой освещенности во время съемки. В такой ситуации оптимальным может стать использование обученного каскада Хаара. Каскад Хаара – один из самых простых способов распознавания объектов с относительно высокой скоростью работы. К целевым объектам можно отнести не только автомобильные номера, но и более сложные элементы, такие как фигуры людей, части тела, лица и так далее. Процесс обучения каскада занимает также относительно немного времени, если использовать ряд имеющихся в открытом доступе наборов, в частности, средства библиотеки OpenCV.

Однако последний метод также имеет некоторые недостатки: малое количество положительных или отрицательных примеров для обучения могут негативно сказаться на качестве работы алгоритма. Поэтому дабы приобрести уверенность в его корректной работе необходимо создать или использовать готовую большую базу фотографий с автомобильными номерами.

Алгоритмы нормализации изображения

Предположим, область номерного знака обнаружена. Однако для корректной ее обработки необходимо создать некоторые дополнительные

условия: номер может быть сфотографирован под углом, допускается наличие загрязнений, помех и т.д.

Первичная задача – разворот и обрезка изображения с номерным знаком по границам последнего. Осуществить это можно с помощью преобразования Хафа [3], активно используемого в компьютерном зрении и при решении ряда схожих с поставленной в данной работе задач.

Следующий шаг – увеличение контрастности целевого участка изображения. Необходимость данной меры очевидна: черный текст на белом фоне становится явно более удобным для восприятия при большем контрасте.

Последним этапом в процедурах нормализации изображения будет разбиение номерного знака на буквы. Это реализуется, как правило, с помощью некоторых процедур бинаризации изображения.

На этапе нормализации есть вероятность также столкнуться с некоторыми уязвимостями алгоритма: ряд помех и специфическое расположение автомобильного номера (к примеру, номер может быть изогнут для некоторых моделей автотранспорта или «утоплен» немного вглубь кузова) может оказать существенное влияние на процедуру разбиения на буквы. На рисунке 2 приведен пример проблемного для процедуры нормализации изображения.

Однако в условиях использования предлагаемого программного обеспечения сотрудниками МВД при исполнении, несомненно, есть возможность очистить номерной знак при остановке автотранспорта и обеспечить тем самым корректные входные данные для использования одного из предложенных выше алгоритмов.

Условия быстрой съемки, очевидно, диктуют необходимость использования более сложных алгоритмов нормализации и распознавания.

Алгоритмы распознавания символов номерного знака

Существующие ныне алгоритмы распознавания текста можно условно разбить на две большие группы: структурные и растровые. Первые основаны на анализе контура букв, морфологии и имеют дело с бинаризацией входных данных (входного изображения). Растровые, в свою очередь, основаны на анализе непосредственно входной картинке в чистом виде.



Рис.2. Снимок со сложными для нормализации данными

На практике зачастую используется некий гибрид растровых и структурных методов распознавания текстовых символов. Конкретизируя вопрос выбора системы распознавания символов, можем выделить несколько интересных алгоритмов.

Tesseract OCR [2] для выполнения данной задачи может быть применен только в случае абсолютной чистоты номерного знака. Грязь и любые схожие помехи приблизительно на 80% снижают эффективность данного алгоритма. Из его очевидных преимуществ можно выделить наличие открытого (бесплатного) программного обеспечения, работающего под разными операционными системами.

Также допустимо применение для решения подобных задач различных нейросетей [4]. Условно их можно разделить на классические (2-3-слойные нейросети) и сверточные нейросети. Последние уже порядка 10-ти лет показывают наилучшие результаты по распознаванию текстов и символов в различных условиях при различных вариациях помех. Однако есть и ряд существенных недостатков, среди которых стоит выделить невозможность диагностики аномального поведения, а также банальную сложность метода.

Выводы

В данной работе предложена модель вспомогательного программного обеспечения для быстрой идентификации автотранспортных средств. В частности, детально рассмотрены алгоритмы предварительного поиска, нормализации изображения, а также распознавания символов номерного знака. Приведена некоторая сравнительная характеристика, достоинства и недостатки различных алгоритмов.

Литература

1. В. Р. Крашенинников. Основа теории обработки изображений / Крашенинников В. Р. . - У.: УлГТУ, 2005.
2. Tesseract - [Электронный ресурс].- Режим доступа: <https://ru.wikipedia.org/wiki/Tesseract>
3. OpenCV. Преобразование Хафа - [Электронный ресурс].- Режим доступа: <http://robocraft.ru/blog/computervision/502.html>
4. Искусственная нейронная сеть - [Электронный ресурс].- Режим доступа: https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C

УДК 004.037

ОБ ОДНОМ АЛГОРИТМЕ АДАПТИВНОГО ТЕСТИРОВАНИЯ

Иванов А.А., Копытова О.М.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
alexey.iva94@gmail.com

Аннотация

Иванов А.А., Копытова О.М. Об одном алгоритме адаптивного тестирования. В статье проведен анализ модели комплексного адаптивного тестирования Х.Ф. Данга. Представлена модель адаптивного тестирования, в которой явно выделены компоненты, подходящие к определенному способу подготовки адаптивного теста, и компоненты, присущие всем методам адаптивного тестирования.

***Ключевые слова:** тестирование, модель адаптивного тестирования, алгоритм адаптивного тестирования*

Постановка проблемы. Тестирование считается одной из форм оценки знаний и значимым компонентом образовательного процесса [1]. Тестирование имеет большое количество конкретных недочетов, но считается главным научно-техническим средством для измерения степени знаний и незаменимо как механизм, помогающий надлежащим способом осуществить регулирование учебного процесса и гарантировать продуктивное педагогическое контролирование.

Существует огромное число технологий адаптивного тестирования, отличающихся между собой методом представления модели учащегося, методом подбора следующего теста, аспектом окончания теста и иными компонентами [2, 3, 4]. Все указанные технологии обладают единой особенностью – они выполняют итеративный метод адаптивного тестирования.

Цель статьи – провести анализ модели комплексного адаптивного тестирования, разработанной Х.Ф. Дангом, и создать модель адаптивного теста, где будут явно выделены компоненты, подходящие к определенному способу подготовки адаптивного теста, и компоненты, присущие всем методам адаптивного тестирования.

Понятие адаптивного тестирования. Адаптивное тестирование – система контроля познаний и оценки итогов учебы, имеющая высокую

эффективность из-за оптимизации операций генерации, предъявления и оценки итогов исполнения адаптационных заданий.

Результативность контрольно-оценивающих операций увеличивается при применении многостадийной стратегии отбора и предъявления тестов, основанной на методах с абсолютной контекстуальной связью. В них каждый следующий этап происходит исключительно после оценки итогов исполнения предшествующего шага. После прохождения обучаемым очередного теста появляется необходимость в принятии решения о выборе трудности следующего теста на основании того, правильным или неправильным был прошлый результат. Метод отбора и предъявления тестов основывается на принципе обратной связи, если испытуемый отвечает на текущий тест правильно, то следующий тест для него будет подобран более высокой сложности, а при неправильном ответе следующий тест будет выбран более лёгким, относительно текущего. Кроме того, существует возможность задать наводящие вопросы, если по выбранной теме испытуемый отвечает стабильно не очень хорошо, для более конкретного определения уровня его знаний.

Анализ модели адаптивного тестирования. Для организации процесса комплексного адаптивного тестирования (АТ) Х. Ф. Данг разработал собственную модель адаптивного тестирования [5].

Она имеет вид: $ATM = \langle KD, LM, QM, AlgAT \rangle$, где: ATM — модель комплексного АТ; KD (Knowledge Domain) — модель предметной области; LM (Learner Model) — модель обучаемого; QM (Question Model) — модель тестового задания; AlgAT — структура алгоритма АТ.

Модель предметной области (МПО) показывает содержание теоретической части, которая используется при тестировании. Данная модель демонстрирует, какие присутствуют взаимосвязи между тестами и как тесты привязаны к содержанию учебного материала. Взаимосвязи показывают взаимоотношения для пары тестов согласно допустимым переводам. МПО представляет собой конечное непустое множество тестов с разрешенными переходами W на Test: $KD = \langle Test, W \rangle$, где: KD — модель предметной области; Test — множество тестов, представляющих собой конечное непустое множество тестовых заданий; W — множество возможных переходов между тестами.

Модель обучаемого (МО) применяется с целью расчета и сохранения оценок итогов тестирования. Модель хранит данные об обучаемом и его итогах тестирования.

Алгоритм АТ показывает, как адаптируется модель тестирования в зависимости от модели обучаемого и модели предметной области. Тестирование на основании метода АТ осуществляет следующие функции:

- исследование действий тестируемого;

– регулирование хода тестирования на основании подобранных методов проведения линии тестирования;

– установка итогов тестирования, выставление оценки тестируемому.

На рисунке 1 представлены связи между атрибутами и свойствами модели комплексного адаптивного тестирования.

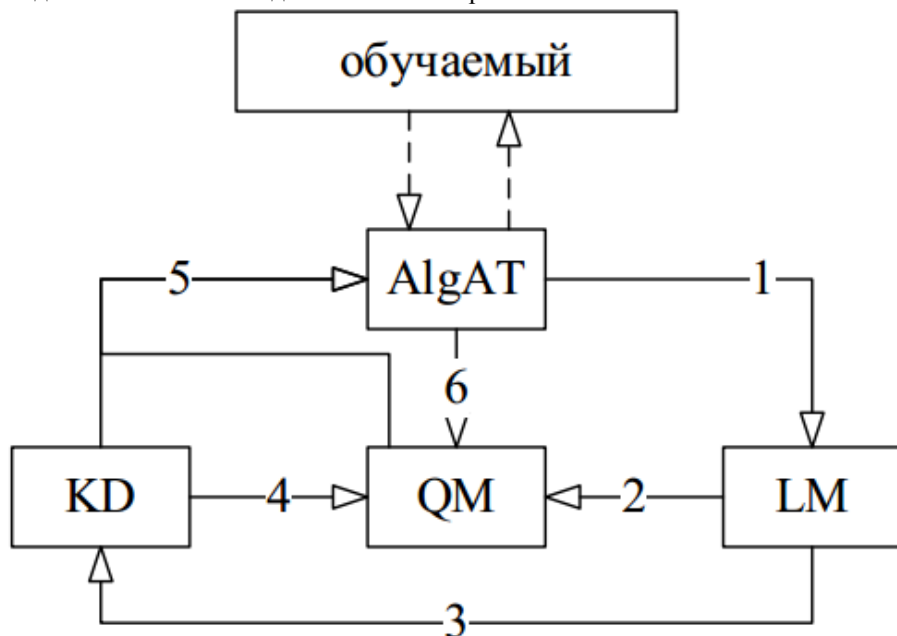


Рис. 1. Взаимосвязи между компонентами модели комплексного АТ

АТ выполняется последующим способом: тестируемый проходит предложенный тест, после чего его результат вносится в структуру алгоритма АТ.

Алгоритм АТ на основании анализа результата тестируемого применяет способ оценивания для выведения данных об обучаемом. Модель обучаемого (МО) на основании результата испытуемого обновляется и отправляет информацию в модель предметной области (МПО) и модель тестового задания (МТЗ). Процедура обучения состоит в следующем: из МПО извлекается и предлагается тестируемому часть теории по избранной теме. Далее на основании текущей степени знания обучаемого формируется представление, как подготавливаемый закончил освоение данного учебного материала, и данные сведения учитываются в подборе последующего учебного материала. Тест формируется на основании структуры метода АТ (результат выполнения

тестового задания), МО (степень способностей тестируемого) и МПО (избранный теоретический материал); информация о параметрах последующего теста выбирается из МТЗ; сформированный тест предоставляется тестируемому.

Подобным способом АТ дает возможность обучаемому какой угодно категории подобрать себе собственную очередность прохождения тестов, т.е. собственную линию движения освоения МПО. Линия движения – освоение обучаемым учебных материалов МПО на основании оценки, полученной при тестировании на прошлом этапе.

Подготовка адаптивного теста. Для подготовки тестирования и дальнейшего его проведения необходимы следующие входные данные: база данных вопросов и стратегии адаптивного тестирования. База данных вопросов включает в себя задания из разных предметных областей, которые не зависят от способов тестирования. Стратегия АТ объединяет в себе все характерные черты определенного способа адаптивного тестирования.

Подготовка адаптивного теста состоит в подборе стратегии тестирования, выборе из базы данных вопросов задания. Впоследствии создается тест, дающий возможность производить оценку знания в некой предметной области и с установленным способом адаптации. В соответствии с полученным тестом АТ производится проверка знаний и умений определенного тестируемого. Процедура тестирования итеративна: на итерации сначала осуществляется подбор и представление следующего вопроса, контроль выбранного ответа, после чего модели обучаемого и стратегий тестирования обновляются, а в завершении итерации проверяется условие завершения теста.

Алгоритм адаптивного тестирования. Подготовленный тест используется в качестве входных данных вместе с идентификатором тестируемого. Идентификатором может быть имя аккаунта в системе либо ФИО.

На рисунке 2 представлен алгоритм проведения тестирования.

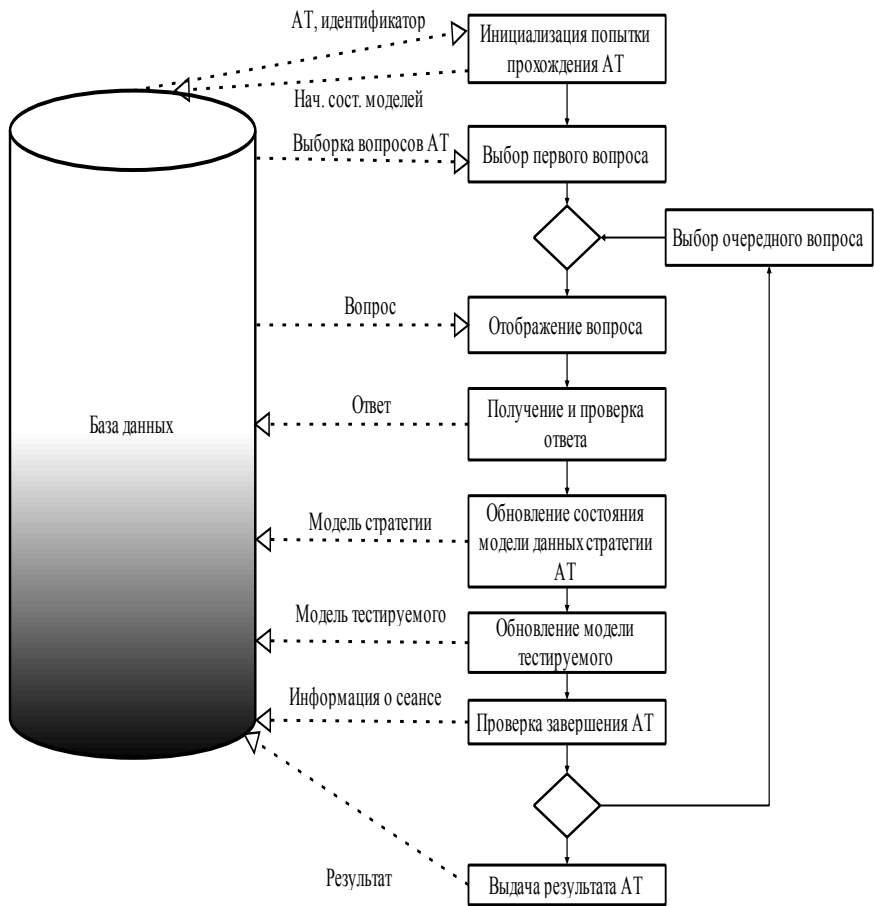


Рис.2 – Алгоритм адаптивного теста

На каждом шаге алгоритма применяются следующие операции, характеризующие выбранной стратегией тестирования:

- формирование начального состояния моделей тестируемого и данных стратегии происходит при инициализации попытки прохождения АТ. При этом применяются операции извлечения начального состояния модели данных тестируемого и операция извлечения начального состояния модели данных стратегии;
- операция подбора первого вопроса;

- операция подбора очередного вопроса;
- операция обновления модели тестируемого;
- операция обновления модели данных стратегии;
- операция проверки завершения теста.

В результате исполнения всех итераций фиксируется попытка прохождения теста, содержащая конечные состояния моделей тестируемого и данных стратегии. На основе указанных данных создается и выводится итог прохождения теста.

Выводы. В статье был проведен анализ модели адаптивного тестирования Х. Ф. Данга с точки зрения взаимосвязей между компонентами модели АТ и тестами. В результате установлено, что модель тестирования адаптируется в зависимости от модели обучаемого и модели предметной области. Предложена модель адаптивного тестирования, которая явно разделена на элементы (модель стратегии, модель тестируемого), зависящие от стратегии тестирования, и элементы, общие для всех методов АТ (база данных заданий, стратегии). На базе приведённой модели предложен алгоритм проведения АТ.

Список литературы

1. Ким, В. С. Тестирование учебных достижений: монография / В. С. Ким. – Уссурийск: Издательство УГПИ, 2007. – 214 с.
2. Зайцева, Л. В. Модели и методы адаптивного контроля знаний / Л. В. Зайцева, Н. О. Прокофьева. – Educational Technology & Society, 2004.
3. Luecht R. M. A Review of Models for Computer-Based Testing / Richard M. Luecht, Stephen G. Sireci // College Board, 2011.
4. Карпенко, А. П. Модельное обеспечение автоматизированных обучающих систем. Обзор / А. П. Карпенко // Наука и образование: электронное научно-техническое издание. – 2011. – № 7.
5. Данг, Х. Ф. Метод разработки алгоритмов адаптивного тестирования / Х. Ф. Данг, В. А. Камаев, О. А. Шабалина // Известия Волгоградского государственного технического университета: межвуз. сб. науч. ст. № 4 (91) / ВолгГТУ. – Волгоград: ИУНЛ ВолгГТУ, 2012.

УДК 004.8

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЗАГРУЗКИ ТРАНСПОРТНОГО СРЕДСТВА С ПОМОЩЬЮ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

Семенова А.П., Стрюковская А.Д.

Донецкий национальный технический университет

E-mail: saniastry95@gmail.com

Аннотация

Семенова А.П., Стрюковская А.Д. Разработка программного обеспечения для загрузки транспортного средства с помощью эволюционных алгоритмов. Проанализированы методы решения оптимизационных комбинаторных задач (задача о загрузке транспортного средства, задача о рюкзаке, задача о раскрое). Рассмотрены основные понятия технологии генетических алгоритмов. Описан метод решения и программная реализация задачи загрузки транспортного средства с помощью генетического алгоритма.

Ключевые слова: задача о рюкзаке, задача раскроя, генетические алгоритмы, задача о загрузке транспортного средства.

Общая постановка проблемы. В настоящее время многие производственные предприятия сталкиваются с задачей загрузки транспортного средства [1]. При решении данной задачи возникает проблема поиска оптимального распределения некоторого ресурса при наличии ряда ограничивающих факторов. Таким образом, задачу о загрузке можно свести к задаче о рюкзаке [4] и задаче о раскрое [3]. Задача о рюкзаке, как и многие оптимизационные комбинаторные задачи, принадлежит к классу NP-полных задач. Ее можно решить полным перебором всех допустимых вариантов заполнения рюкзака имеющимися предметами, однако при больших массивах входных данных такой переборный алгоритм практически неприемлем, поскольку имеет экспоненциальную сложность относительно длины входа. Если же применить идеи «метода ветвей и границ» [5], то в большинстве случаев объем перебираемых вариантов можно сократить. Подобные алгоритмы с ограниченным перебором могут иметь неплохую сложность в среднем, но в худшем случае все равно остаются экспоненциальными. Поэтому на практике часто используют различные модификации «жадного» алгоритма [5], который имеет полиномиальную сложность, что и является его основным достоинством. В данной работе будет представлен метод загрузки транспортного средства на основе генетического алгоритма.

Цель статьи – изучение технологии генетических алгоритмов для решения задачи загрузки транспортного средства, исследование и анализ существующих аналогов данной задачи, а так же разработка программного обеспечения для поиска оптимального варианта размещения груза.

Постановка задачи. Задачу о загрузке транспортного средства можно сформулировать следующим образом: из заданного множества объектов с определенными свойствами необходимо выбрать и разместить в транспортном средстве такое количество объектов, чтобы количество объектов было максимальным, площадь свободного места в кузове была минимальна. При этом необходимо учитывать все ограничения, как груза, так и транспортного средства.

Объектом разработки является программное обеспечение для загрузки транспортного средства. Программная реализация будет складываться из пяти модулей, объединенные в одну программу, которая будет размещать груз по заданным параметрам на определенную поверхность. В программе будут реализованы задачи рюкзака и о раскрое, которые с помощью генетических алгоритмов позволят ближе к оптимальному разместить груз на поверхность.

Исследования. Программное обеспечение реализовано с целью уменьшения затрат времени на поиск оптимального размещения груза и загрузку транспортного средства.

В ходе разработки программного обеспечения были исследованы следующие понятия:

- виды генетических алгоритмов [1];
- виды генетических операторов и их принадлежности [5];
- поиск наиболее оптимального способа заполнения блоками пространства;
- типичные ошибки при компоновке груза [1];
- особенности и значимость задач рюкзака и раскроя [4];
- анализ рассмотренных аналогов программного обеспечения.

Каждый груз обладает своими параметрами (вес, высота, ширина, содержимое груза), которые необходимо учитывать при загрузке. Также необходимо помнить, что и транспортное средство имеет свои ограничения (грузоподъемность, высота и ширина кузова, площадь поверхности). Таким образом, в процессе анализа предметной области было выявлено, что решение задачи о загрузке можно свести к решению задач о рюкзаке и раскрое, с помощью которых решение сводиться к оптимальному. Все аспекты, которые приведены выше, нужно учитывать при создании программного обеспечения.

Метод решения. Для решения классических задач о рюкзаке и раскрое

существуют уже известные генетические алгоритмы [1]. Приведем описание генетического алгоритма, позволяющего решить задачу о загрузке транспортного средства. В данной задаче генетический алгоритм, позволяет получить оптимальные решения, сочетающие в себе случайные методы. Размер начальной популяции составляет 20 особей. Хромосома представлена в виде матрицы. В качестве способа кодирования хромосом было выбрано бинарное кодирование. Формирование новой популяции осуществляется с помощью таких генетических операторов:

- отбор родителей – заключается в турнирном отборе;
- скрещивание – происходит обмен одним геном;
- мутация – является генной.

Поскольку в задаче о загрузке транспортного средства суммарная стоимость загруженного груза должна стремиться к максимуму. Предположим, что количество типов груза равно n . Обозначим стоимость i -го предмета как p_i , а i -й ген хромосомы как h_i . Тогда фитнес-функция примет вид:

$$FF = h_1p_1 + h_2p_2 + \dots + h_np_n \rightarrow \max \quad (1)$$

Но на значение фитнес-функции влияют некоторые условия задачи:

- грузоподъемность транспортного средства;
- площадь кузова транспортного средства;
- габариты груза.

Эти ограничения учтены в штрафной функции.

В качестве критерия остановки работы генетического алгоритма были выбраны следующие критерии:

- генетический алгоритм работает до тех пор, пока число поколений не достигло заданной величины;
- значение фитнес-функции не изменилось за последние десять поколений.

Программная реализация. Программное обеспечение позволит пользователю вводить необходимые параметры груза и полотно для загрузки транспортного средства. Программный продукт будет реализован в виде нативного приложения под операционную систему семейства Windows с использованием Net.Framework не ниже 4.1 версии и состоять из следующих модулей:

Разрабатываемый программный продукт представлен программы и состоит из нескольких модулей:

- модуль списка ограничений по грузу;
- модуль параметров груза;
- модуль размеров поверхности;
- модуль отображения заставки, отвечающий за загрузку основной формы;
- модуль отображения справочной информации;
- модуль сохранения результатов наилучшей компоновки.

Программная реализация создана на языке C# и состоит из главного меню и последовательно вызываемых форм.

В формах, в свою очередь, размещены элементы интерфейса, взаимодействие с которыми вызывает другие элементы или формы. На рисунке 1 показана общая структура вызовов элементов программы и их взаимодействия.



Рис. 1. Общая структура вызовов элементов программы и их взаимодействия

Выводы

Была исследована и проанализирована задача загрузки транспортного средства, а также ее аналоги. Описан алгоритм решения данной задачи с помощью генетического алгоритма. Приведено описание основных модулей

разрабатываемого программного обеспечения и построена общая схема взаимодействия этих модулей.

Список литературы

1. Панченко Т. Генетические алгоритмы в примерах / Панченко Т. – М.: Питер, 2007. – 89 с.
2. Падлазова И. Генетические алгоритмы на примерах задачи раскроя / Падлазова И. – М. : Питер, 2015. – 50 с.
3. Задача раскроя [Электронный ресурс]. – Режим доступа : <http://splanet.ru/referat9/referat-1491102.php>
4. Задача раскроя и рюкзака [Электронный ресурс]. – Режим доступа : <http://splanet.ru/referat6/referat-1491102.php>
5. Основы генетические алгоритмы. Задача о загрузке рюкзака [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/post/209738/>
6. Павловская Т. А. С/С#. Программирование на языке высокого уровня / Павловская Т. А. – СПб.: Питер, 2010.
7. Давыдов В. Г. Технологии программирования. С#: [руководство]/ В. Г. Давыдов. –СПб.: БХВ-Петербург, 2005.
8. Шилд Г. С#. Объектно-ориентированное программирование. Практикум. СПб.: Питер, 2008.

УДК 004.588

РАЗРАБОТКА ИНФОРМАЦИОННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ ПО КУРСУ «ОСНОВЫ ПРОГРАММИРОВАНИЯ И АЛГОРИТМИЗАЦИИ»

Струков В.А., Сорокин Р.А.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
E-mail: yourmonitor@mail.ru

Аннотация

Струков В.А., Сорокин Р.А. Разработка автоматизированной обучающей системы по курсу «Основы алгоритмизации и программирования». Рассмотрены основные причины актуальности автоматизированных обучающих систем в настоящее время. Выделены принципы построения, а так же функциональные составляющие проектируемой системы. Описаны основные способы применения и внедрения в эксплуатацию разрабатываемой системы.

Ключевые слова: АОС, компьютерное обучение, дистанционное обучение.

Постановка проблемы

В настоящее время дистанционное обучение приобретает все большее распространение [6]. Это связано, в первую очередь, с необходимостью оптимизации учебного процесса. Нередко в учебных заведениях высшего и среднего уровня возникают ситуации, когда объем учебного материала велик настолько, что на изложение данного материала в лекционной форме не хватает времени.

Возможна и иная ситуация – материал скомпонован и распределен, но количество обучаемых настолько велико, что проведение столь массовых лекционных или практических занятий требует слишком большого количества ресурсов. При возникновении непредвиденных обстоятельств, при которых студент и преподаватель не могут контактировать, процесс обучения так же не должен стоять на месте.

Для решения подобного рода проблем разрабатываются автоматизированные обучающие системы [1], позволяющие производить параллельное обучение большого количества студентов, при этом, не требуя личного контакта обучающегося и преподавателя.

Цель проекта – разработка автоматизированной обучающей системы, введение в эксплуатацию которой позволит оптимизировать учебный процесс в учебных заведениях высшего и среднего уровня. Использование

разрабатываемой системы способствует приобретению обучающимися основных теоретических сведений по курсу «Основы алгоритмизации и программирования», а так же позволяет закрепить полученные знания путем выполнения практических заданий и типовых задач [2].

Постановка задачи

Объектом разработки является автоматизированная обучающая система, осуществляющая обучение по программе «Основы алгоритмизации и программирования».

Разрабатываемая система будет осуществлять обучение студентов без прямого взаимодействия с преподавателем. Информация в программе изложена в виде теоретического материала, структурированного на лекции, что способствует улучшению восприятия путем создания ощущения «пошагового обучения». Для закрепления приобретенных знаний будет реализована система контроля успеваемости [4], столпами которой являются тесты и типовые задачи для выполнения.

Для улучшения восприятия материала обучаемыми, предусмотрено применение мультимедийных технологий. В процессе обучения информация предоставляется не только в виде текстового лекционного материала, но и различного рода изображений, а так же аудио и видеозаписей. Предусмотрена статистическая составляющая, позволяющая как преподавателю, так и обучаемому отслеживать прогресс обучения по курсу и знакомиться с результатами [5].

Назначение разрабатываемой системы

Разрабатываемая автоматизированная обучающая система предназначена для введения в эксплуатацию в учебных заведениях высшего и среднего уровней. Целевой аудиторией для использования являются студенты и преподавательский состав заведений. Программа актуальна так же для людей, не получающих техническое образование, но желающих ознакомиться с данной областью знаний [3].

Система проводит обучение следующим аспектам:

- основы алгоритмизации и понятие алгоритма;
- основные принципы построения блок-схем;
- введение в основы языка C++;
- типы данных в программировании C++;
- переменные и арифметические операции;
- условные и циклические операторы;
- массивы данных в C++;
- работа с файлами;
- введение в объектно-ориентированное программирование.

Программа предусматривает построение «диалога» с пользователем. Данный психологический аспект реализуется при помощи многочисленных подсказок, справочного материала, а так же простого и доступного для понимания интерфейса. Файлы мультимедиа способствуют упрощению восприятия информации обучаемым. Изображения и видео материалы воспринимаются более доступно и более информативны для некоторых пользователей. Таким образом, в определенных разделах программы обучающийся сможет самостоятельно выбрать, в каком виде учебный материал для него более понятен [5].

Программная реализация

Программный продукт разрабатывается средствами языка C# с использованием технологии Windows Forms. Для реализации многочисленных функций подключаются дополнительные программные библиотеки. Так же используется стандартный набор операторов и функций, а так же дополнительные возможности, предоставляемые программным продуктом Microsoft Visual Studio 2012/

Основой программы является главная форма, содержащая функциональные элементы, позволяющие пользователю переходить к работе с дополнительными формами. Главная форма содержит в себе навигационную панель и лекционный материал.

Дополнительные формы открывают доступ к дополнительным материалам, таким как мультимедийные файлы, материалы для тестирования, а так же статистическим данным обучающегося. Переход к дополнительным формам осуществляется при помощи управляющих элементов, расположенных на главной форме программы. Часть данных форм содержит побочную информацию, ознакомление с которой способствует освоению изученного материала. Переход к ряду форм обязателен для дальнейшего продвижения по учебному курсу программы.

Разрабатываемый программный продукт состоит из описанных ниже модулей.

1. Модуль авторизации. Данный модуль обеспечивает разделение пользователей путем создание личной учетной записи, к которой будет привязана статистическая информация. Предусмотрена авторизация при наличии уже готовой записи, или создание новой.

2. Лекционный модуль. Отвечает за отображение лекций, их структурирование а так же последовательность подачи. Ограничивает пользователя при попытке просмотра более поздних лекций, без прохождения тестового контроля предыдущих.

3. Модуль тестирования. Модуль реализован в виде дополнительных форм, содержащих в себе тесты и типовые задачи для выполнения.

Выполнение задач происходит после ознакомления с лекционным материалом и обязательно для дальнейшего продвижения по обучающей программе.

4. Статистический модуль. Обеспечивает синхронизацию файлов учета статистики пользователей с программой. Так же служит для отображения статистических данных пользователя в специальном окне.

5. Мультимедийный модуль. Предоставляет доступ пользователю к дополнительным видео, аудио или другим материалам. Реализован в виде отдельных форм, переход к которым предоставляется в процессе обучения.

6. Модуль защиты данных. Данный модуль реализован в виде алгоритма шифрования личных данных пользователя. При создании учетной записи введенный пароль шифруется и сохраняется в отдельный файл.

7. Справочный модуль. Реализован в виде текстовых подсказок и справочных разделов, помогающих пользователю в использовании программы, а так же навигации по ней.

Выводы

Была рассмотрена проблема оптимизации учебного процесса в учебных заведениях высшего и среднего уровня.

Так же, в данном контексте, рассмотрен вопрос актуальности разработки автоматизированных обучающих систем для решения подобного рода проблем. Определено назначение и цели эксплуатации разрабатываемого программного продукта. Выделена целевая аудитория для использования программы.

Проведено разделение разрабатываемой автоматизированной обучающей системы на функциональные модули, а так же описание основных принципов взаимодействия с пользователем.

Список литературы

1. Бухаров М.Н., Практическая теория компьютерного обучения, деловые игры и экспертные системы / М.Н. Бухаров. – СПб. : Питер, 2001. – 640 с.
2. Бычков С.Н., О тождестве фундаментальности и гуманитарности в общем образовании / С.Н. Бычков, Е.В. Шишкин – СПб. : Питер, 1985. – 360 с.
3. Гусева А.И. Оценка качества распределенных обучающих систем / А.И. Гусева – СПб. : Питер, 2001. – 500 с.
4. Клебанов И.И. Групповой анализ дифференциальных уравнений в базовом математическом образовании / И.И. Клебанов – Москва : Бино, 2001. – 255 с.
5. Мельников А.В., Основные принципы автоматизации проектирования обучающей системы / А.В Мельников, П.Л. Цитович, Б.Ф. Цитович. – Москва : Бино, 1990. – 320 с.
6. Полат Е.С. Дистанционное обучение / Е.С. Полат – СПб : Питер, 2003. – 300с.

УДК 004.4'236

ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ ДЛЯ СОЗДАНИЯ WEB-САЙТА

А.Д. Гетманова, П.Д. Михневич, К.А. Ручкин

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа

E-mail: anyutkagetmanova@gmail.com

Аннотация

Гетманова А.Д., Михневич П.Д., Ручкин К.А. Выбор программного обеспечения с открытым исходным кодом для создания Web-сайта. Проанализированы WordPress и самописная системы управления сайтами, выявлены преимущества и недостатки обеих CMS на основе которых была определена более приемлемая система управления.

Ключевые слова: *WordPress, CMS, сайт, самописная система*

Введение

В настоящее время Web-сайты являются, пожалуй, самым популярным способом опубликовать или найти необходимую информацию. Отчасти благодаря тому, что инструментальные средства для их создания становятся более удобными в своем использовании.

CMS (Content Management System) – это система управления содержимым сайта, которая представляет собой информационную систему или компьютерную программу, используемую для обеспечения и организации совместного процесса создания, редактирования и управления содержимым сайта.

Системы управления сайтами предоставляют инструменты для создания и организации совместной работы над содержимым, позволяют управлять и публиковать содержимое, а также представлять информацию в виде, удобном для навигации или поиска.

Подобные системы часто используются для хранения, пересмотра и публикации различных данных: документов, фотографий, фильмов, номеров телефонов и так далее.

Целью исследования является определение наиболее приемлемой системы управления сайтом.

Задача данной работы состоит в рассмотрении систем управления сайтами WordPress и самописной системы, а также анализе преимуществ и недостатков данных CMS.

Актуальность данного исследования заключается в необходимости выбрать подходящую систему, способную удовлетворить требования,

касающиеся создания, управления и редактирования сайта, а также имеющую меньшее количество недостатков.

Система управления WordPress

WordPress – самая популярная в настоящий момент времени бесплатная CMS с открытым исходным кодом, которая была разработана в 2001 году Мэттом Мюллервегом. Система применяется как для создания блогов, так и для создания достаточно сложных новостных ресурсов, интернет-магазинов[1].

Система написана на языке PHP, сервер базы данных –MySQL, выпущена под лицензией GNU GPL версии 2 и занимает первое место среди других систем управлений.

Преимущества WordPress

Система управления WordPress является одной из самых простых CMS, для освоения которой не требуются знания программирования, а для работы понадобятся базовое руководство и минимальный набор инструментов. Она имеет множество преимуществ, среди которых:

- большой выбор бесплатных шаблонов, который обеспечивает возможность создания любых проектов;
- быстрая установка и настройка системы;
- понятная панель управления администратора;
- наличие возможности изменять код шаблона.

Помимо этого система постоянно обновляется, появляются новые версии, которые имеют большее количество интересных тем, возможностей и интерфейсов по сравнению с предыдущими версиями, а подробную документацию и последнюю версию можно найти на официальном сайте.

Особенности

WordPress может быть установлен не только непосредственно на хостинг, но и на сам компьютер. Система поддерживает большое количество языков и позволяет публиковать записи с помощью стороннего программного обеспечения и сервисов. Имеется специальный режим – gzip, который позволяет сжимать определенные ресурсы с сервера, – картинки, скрипты, контент, стили оформления перед их отправкой. Активировать данный режим можно в настройках в любое удобное для пользователя время.

Существенной особенностью WordPress является возможность обмена информацией посредством методов Pingback и TrackBack, которые представляют собой уведомления о ссылке на какой-либо документ [3].

Система является удобной в использовании. Помимо этого статьи можно защищать паролем, заносить в архив с возможностью будущей публикации в определенный день, ограничивать их комментирование посредством модерации. Кроме того, администратор может ограничить использование ряда

HTML-тегов. Данные возможности способны защитить сайт от спамеров, потенциальных взломов и мошенничества.

Среди дополнительных возможностей WordPress следует отметить: встроенный инструмент поиска, плагины, существенно расширяющие функциональность ядра блога, а также широкие возможности работы со статьями – наличие разбивки на страницы, автоматической публикации, функции предпросмотра и черновика.

Недостатки

Среди недостатков системы WordPress можно выделить большую нагрузку. Загрузка страниц сайта может занимать некоторое время из-за большого количества плагинов, также, при большом количестве посетителей может возникнуть серьезная нагрузка на хостинг.

WordPress имеет малую функциональность базового движка. Для получения большей функциональности системы необходимо устанавливать отдельные плагины, так как набор базовых функций ограничен.

Неполная документация также является недостатком системы. Несмотря на большое количество справочных материалов в интернете, только малая часть переведена на русский язык, в связи с чем возникает проблема нахождения толкового руководства по WordPress.

Похожая проблема возникает с плагинами, темами оформлений и виджетами, которых в интернете достаточное количество, что может осложнять поиск качественных доработок.

Еще одним недостатком является слабая безопасность системы, однако эта проблема значительно обработана, а сама система постоянно совершенствуется разработчиками.

Преимущества самописной CMS

Самописные системы управления сайтом – это индивидуальные системы управления, которые создаются самой веб-студией [4]. Основными ее преимуществами являются:

- масштабируемость и гибкость управления;
- безопасность, так как программный код имеется только у владельца сайта и разработчика системы;
- простота управления;
- высокая производительность. За счет облегченного кода система управления не будет слишком большой, что позволит экономить на сервере и обслуживании системы.

Недостатки

Самописная система управления сайтами имеет несколько недостатков:

- стоимость разработки;

– привязанность к разработчику, от которого зависит развитие системы, изменения CMS;

– отсутствие аудита безопасности, который включает детальную проверку сайта на наличие уязвимых элементов;

– малая распространенность системы;

– отсутствие API – набора готовых функций, структур и классов, предоставляемых приложением для ее использования во внешних программных продуктах;

– отсутствие профессиональных тестировщиков.

Помимо этого самописные системы имеют несколько недостатков, которые зависят от разработчика самой системы. К ним может относиться отсутствие полноценной пользовательской документации, без наличия которой достаточно сложно разобраться в системе. Также отсутствие профессионального сообщества и службы поддержки, в которых можно быстро найти ответы на типовые задачи относительно системы.

Выводы

Таким образом, в процессе исследования были выявлены существенные недостатки и преимущества WordPress и самописной систем управления сайтами, на основе которых можно определить систему более удобную в использовании. Такой системой является бесплатная CMS WordPress, так как, несмотря на свои недостатки, которые постепенно устраняются разработчиками, сама система совершенствуется с новыми версиями и обновлениями. Также немаловажными достоинствами системы являются простота в освоении и доступность данной CMS.

Самописные системы имеют высокую стоимость разработки, и зависят от разработчиков, не только в выходе новых обновлений, но и в качестве написания самой системы.

Список литературы

1. Скотт Адам Д. WordPress для изучения / Адам Д. Скотт. – Бирмингем: Профессиональное издательство ООО, 2012. – 144 с.

2. Официальный сайт WordPress [Электронный ресурс]. – Режим доступа: <https://wordpress.org>

3. Трекбеки в WordPress, что это такое [Электронный ресурс]. – Режим доступа: <https://wpcafe.org/hacks/trekbeki-trackbacks-v-wordpress-что-это-такое/>

4. Почему нужно выбирать самописную CMS [Электронный ресурс]. – Режим доступа: <http://incon.solutions/ru/blog/news/pochemu-nuzhno-vybirat-samopisnuju-CMS-что-такое-admin-panel-CMS.html>

УДК 004.912

РАЗРАБОТКА СИСТЕМЫ МАШИННОГО ПЕРЕВОДА ТЕКСТОВ В АНГЛО-РУССКОМ НАПРАВЛЕНИИ

Сорокин Р.А., Оверченко Я.Ю.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: overchenko96@mail.ru

Аннотация

Сорокин Р.А., Оверченко Я.Ю. Разработка системы машинного перевода текстов в англо-русском направлении. Рассмотрены основные подходы к построению систем машинного перевода. Реализована система машинного перевода с английского языка на русский, основанная на правилах. Описаны некоторые особенности лексического анализа и синтеза в системах машинного перевода текстов.

***Ключевые слова:** машинный перевод, системы машинного перевода, правила лексического анализа, лексический синтез.*

Общая постановка проблемы

С развитием информационных технологий и международных связей с каждым годом возрастает количество различной информации, требующей перевода на родной язык людей, работающих с ней. Таковой информацией является техническая документация, художественная литература, множество всевозможных деловых и официальных документов и так далее. Требуется также перевод при взаимодействии друг с другом представителей различных стран. Для обработки столь больших объёмов данных порой недостаточно работы человека, поэтому автоматизация набирает обороты и в сфере переводов.

С развитием технологий возрастает необходимость в программном обеспечении, которое способно выполнить качественный и осмысленный перевод больших объёмов информации за короткое время. Эта потребность способствует развитию систем машинного перевода различных типов, способных выполнять работу человека-переводчика.

Цель проекта – изучение морфологических особенностей русского и английского языков, исследование существующих подходов к машинному переводу, их особенностей и основных этапов, а также разработка на основе

полученных данных программного продукта, позволяющего произвести перевод с английского языка на русский.

Постановка задачи

Объектом разработки является система машинного перевода текстов с английского языка на русский, основанная на правилах.

Разрабатываемый программный продукт будет предоставлять пользователю возможность перевода различных текстов с английского на русский язык, основываясь на данных морфологического анализа и правилах словоизменения.

Простота и удобство в использовании играют важную роль в процессе работы пользователей с программным продуктом. Поэтому разрабатываемая система машинного перевода будет обладать простым дизайном, реализующим основной функционал данного программного продукта – непосредственно перевод.

В основе систем машинного перевода часто лежит словарь пословных переводов или двуязычные корпуса текста. От их полноты напрямую зависит качество получаемого системой перевода. Работа разрабатываемого программного продукта основана на словаре пословных переводов, содержащем 50 000 записей.

Исследования

Программный продукт будет полезен в случае необходимости перевода слов и текста с английского языка на русский. В ходе проведения подготовительной работы и непосредственно разработки системы машинного перевода был исследован ряд следующих теоретических и практических аспектов:

- теория машинного перевода;
- история развития систем машинного перевода;
- анализ и синтез в системах машинного перевода;
- морфологический анализ;
- основные подходы к реализации машинного перевода;
- классификация систем машинного перевода.

В ходе исследования было определено, что работа большинства систем машинного перевода базируется на использовании словарей пословного перевода или двуязычных корпусов текстов [1].

После обзора имеющихся систем и анализа существующих подходов к машинному переводу для разрабатываемого программного продукта был

выбран словарь пословных переводов объемом в 50 000 единиц, а также выбран тип – система машинного перевода, основанная на правилах [5].

Кроме того в процессе работы над программным продуктом были исследованы основные морфологические особенности русского и английского языков, на основе которых были созданы правила словоизменения, регулирующие процедуру конечного синтеза словоформ [2]. Также был разработан алгоритм морфологического анализа найденных переводов английских слов для их дальнейшей коррекции и согласования [3].

Программная реализация

Программа предоставляет пользователям возможность перевода слов и текстов с английского языка на русский.

Разрабатываемый программный продукт представлен в виде нативного приложения и состоит из нескольких модулей:

- модуль загрузки словарной базы, считывающий данные из БД во внутреннюю структуру программы для дальнейшей работы во время отображения загрузочного экрана;

- модуль считывания текста и его разделения на структурные единицы для дальнейшей обработки;

- модуль поиска перевода слов в словарной базе;

- модуль определения морфологических характеристик найденных переводов, на которых основан конечный синтез перевода;

- модуль отображения справочной информации о программе и её авторе, который можно вызвать нажав специальную клавишу на главном окне программы;

- модуль конечного синтеза перевода согласно сформированным ранее правилам словоизменения.

Программная реализация выполнена в среде языка программирования C# и состоит из загрузочного экрана и основной рабочей формы [4].

В форме, в свою очередь, размещены элементы интерфейса, взаимодействие с которыми активирует те или иные модули программы. На рисунке 1 показана общая структура вызовов элементов программы и их взаимодействия.

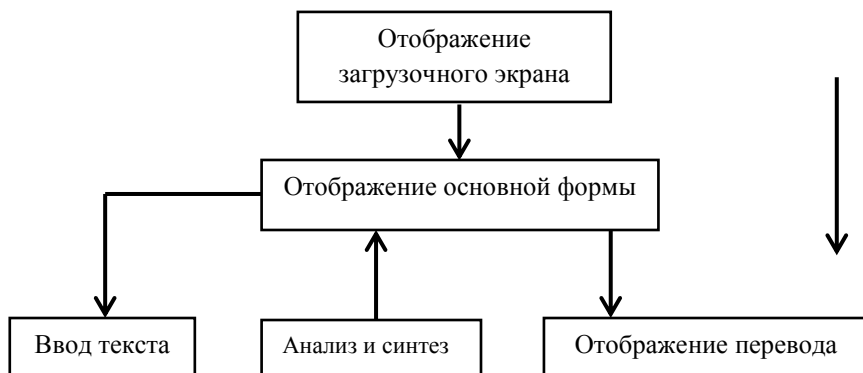


Рис. 1. Общая структура вызовов элементов программы и их взаимодействия

Выводы

Была исследована проблема необходимости использования и развития систем машинного перевода. Также определены цели, достигаемые при разработке и использовании программного продукта. Проведен анализ теоретических аспектов построения систем машинного перевода, а также определен конкретный тип и подход к разработке программного продукта. Выделены основные модули, составляющие систему машинного перевода, построена общая схема их взаимодействия.

Список литературы

1. Белоногов Г. Г. «Компьютерная лингвистика и перспективные информационные технологии» – М.: Русский мир, 2004 – 300 с. литературы на иностранных языках, 1958 – 459 с.
2. Марчук Ю. Н. «Основы компьютерной лингвистики. Учебное пособие. Издание 2-е дополненное» – М.: Изд-во МПУ «Народный учитель», 2000 – 226 с.
3. Марчук Ю. Н. «Проблемы машинного перевода» – М.: Наука, 1983 – 232 с.
4. Павловская Т. А. Щупак Ю.А. С#. Объектно-ориентированное программирование. Практикум. СПб.: Питер, 2008.
5. Хроменков П. Н. «Анализ и оценка эффективности современных систем машинного перевода» Диссертация на соискание учёной степени кандидата филологических наук. – М.: РГБ, 2000 – 265 с.

УДК 004.921

ОБЗОР СИСТЕМ ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ

Ручкин К.А., Егурнов Н.О.

Донецкий национальный технический университет

E-mail: ygurnovn@gmail.com

Аннотация

Ручкин К.А., Егурнов Н.О. Обзор систем процедурной генерации. Существует множество различных методов процедурной генерации позволяющих создавать различные игровые элементы. Естественным образом разнообразие контента в игре провоцирует ее реиграбельность тем самым продлевая жизнь данной игры. Так как процедурная генерация отличный инструмент для создания множества объектов в реальном времени что в своей сути упрощает разработку и уменьшает затраты так же является хорошим способом разнообразить игровой опыт. Представленные генераторы позволяют создавать достаточно много разнообразных вещей

Ключевые слова: процедурная генерация, текстурирование, генераторы объектов, ландшафт, карта высот.

Введение. Процедурная генерация – алгоритм исходя из которого во время выполнения игрового приложения производится генерация определенной части контента: текстуры для объектов, анимация, игровая карта, предметы, звуки, события и т.д. Процедурная генерация контента является одним из наиболее актуальных и активно развивающихся направлений исследований в сфере мультимедиа, в частности в индустрии видеоигр. Главное назначение упростить создание огромного количества объектов вручную, уменьшая тем самым затраты на разработку и упрощая задачу малым командам разработки. Использование ПГК позволяет не только значительно понизить стоимость создания контента, но и решает проблему персонализации, приобретающую большую значимость в связи с увеличением количества потенциальных игроков. В исследованиях алгоритмов ПГК особый интерес представляет проблема процедурной генерации игровых правил, которые лежат в основе всех, в том числе и неэлектронных, игр. Помимо этого внести определенное разнообразие в существующую концепцию игры, разнообразить геймплей. Так же позволяет создавать неограниченные по своей площади игровые миры, не затрачивая на это много времени, что не возможно было бы осуществить, проводя разработку вручную. Решение данной задачи обладает практической значимостью, поскольку процедурная генерация позволит не только существенно разнообразить видеоигры за счет динамического изменения правил в процессе игры (в том числе, адаптируя игровой процесс под желания игрока), но и, возможно, создать совершенно новые игровые механики и жанры. Кроме того, системы

генерации игровых правил могут быть полезны исследователям алгоритмов искусственного интеллекта в сфере универсальных игровых программ, а также игровым дизайнерам для быстрого создания и тестирования различных игровых прототипов. Существует множество различных алгоритмов процедурной генерации часто используемых для конкретных целей, генерации анимации текстур ландшафтов звуков.

Генератор анимации. Хорошим примером использования генерации анимации является «euphoria». Программное средство, предназначенное для создания процедурной анимации, работающее в режиме реального времени. Разработанное компанией Natural Motion на основе технологии «Dynamic Motion Synthesis». Euphoria применяется для процедурной анимации трёхмерных персонажей и базируется на полной симуляции трёхмерного персонажа, включая его тело, мускулы и моторную нервную систему. Вместо того, чтобы использовать предопределённые анимации, действия и реакции персонажей синтезируются в реальном времени. Эти анимации каждый раз являются особенными, неидентичными к предыдущим, даже если повторяется одна и та же сцена. В то время, когда традиционные физические движки используют физику ragdoll (физику «тряпичной куклы») для анимаций, сгенерированных на лету, данное программное средство использует более комплексный метод для физически связанных объектов во всей полноте внутри игрового окружения.

Процедурное текстурирование. Процедурное текстурирование – метод создания текстур, при котором изображение текстуры создается с помощью процедурного алгоритма. Тем самым создавая уникальные текстуры камня, дерева, металла обеспечивающие разнообразие. Лучше всего процесс процедурного текстурирования представить в виде блоков (операторов). Существует три типа блоков: генераторы, фильтры, вспомогательные.

Каждый генератор и фильтр реализует какой-либо процедурный алгоритм. Каждый блок имеет совокупность параметров. Даже если не использовать такую схему все равно она сводится к этому общему случаю. Для создания «природных» текстур, таких как дерево, гранит, металл, камни, лава в качестве фильтров используются фрактальный шум и ячеистые текстуры. На данный момент генерация процедурных текстур доступна как стандартный инструмент в большинстве современных игровых движках.

Генератор объектов. Существуют так же генераторы объектов. Процедурно созданные элементы ландшафта, здания, предметы. Примером является Plant Factory набор инструментов предназначенных для генерации моделей растительности полностью готовыми для использования в игровом движке. Растения, созданные с помощью Plant Factory могут быть экспортированы в любой стандартный формат трёхмерных объектов, такие как Alembic, FBX, 3DS, OBJ, C4D, LWO и т.д.

Растения полностью текстурированы, так же поддерживается создание анимации, которую в последствии можно настраивать в 3D редакторе или игровом движке.

Plant Factory позволяет генерировать растений по множеству параметров в основных условиях таких как вид размер и более углублённые возраст растения, время года, здоровье. Генератор достаточно гибки для создания любого вида растительности травы, кустарников, деревьев.

Фактические виды растений загружается с конкретными установками, которые изменяются определенным образом. Тем самым позволяя создать несколько вариантов одного типа растения молодые, пожилые, зимнее, летнее, процветающие, умирающие. Все параметры при желании можно изменить и вручную. Так же возможно динамическое изменение растения, к примеру, внешний вид модели может, меняться в зависимости от времени произвольно с конца зимы до начала лета. Позволяя создавать динамическое окружение.

Генератор ландшафтов. Генераторы ландшафтов используются для создания игровой карты. Уже давно в картографии для описания земной поверхности применяется понятие "карта высот". Суть метода состоит в использовании концентрических областей разного цвета, где каждому оттенку соответствует какая-либо высота относительно некоего базового уровня, обычно уровня моря. На этом базируется большинство генераторов ландшафтов, карты высот генерируются автоматически, а в зависимости от первоначальных настроек. Другой подход к представлению высот на местности - векторный. Он состоит в следующем: каркас будущей местности строится из набора векторных "хребтов" - линий, которые могут быть как замкнутыми, так и в виде кривой линии. Затем полученная система зашумляется путем добавления в нее случайных элементов по некоему фрактальному закону и просчитывается положение всех элементов относительно уровня моря. В результате этих действий получается пространственный каркас моделируемой местности, который в дальнейшем при желании можно модифицировать.

Примером генератора ландшафтов может послужить Vue. Создателям Vue удалось сделать программ настолько универсальной, что по своим функциональным возможностям этот генератор ландшафтов плотную приблизился к обычному трехмерному редактору. Vue интегрируется с популярными пакетами для моделирования и анимации. В 3D-редакторах Maya, Lightwave, 3ds Max, Cinema4D выбор визуализатора остается на усмотрение пользователя. Интеграция Vue редакторами дала большие преимущества программе. Помимо расширенных возможностей визуализации, программа приобрела очень мощный инструментарий для моделирования, которым не может похвастаться ни один генератор трехмерных ландшафтов. Собственный инструментарий программы довольно широк. VueXStream напоминает конструктор Вселенной. Среди объектов, которые можно

создавать в программе, есть как стандартные объекты-примитивы (куб, сфера, тор и пр.) так и специальные объекты, без которых создание ландшафтов было бы невозможным. Среди специфических объектов программы - скала, вода, планета, растение и облачность. Всеми этими объектами можно управлять в программе, настраивая их форму и внешний вид. Для редактирования свойств объектов в программе используется специальный менеджер. Этот инструмент позволяет быстро перейти к свойствам той или иной модели в трехмерном пространстве. При нехватке системных и аппаратных ресурсов, программа меняет вариант отображения объектов в окнах проекций на упрощенный, вплоть до условных параллелепипедов.

Одно из неоспоримых удобств программы – возможность использования огромного количества библиотек. В этих библиотеках можно найти множество различных моделей растений, наборы материалов для имитации горных пород любых типов, а также различные модели освещения – от анимированного восхода солнца до фантастической неестественной дымки вечерних сумерек. Анимация программы может быть настроена в автоматическом режиме или при помощи специального мастера, который помогает описать характер движения анимированного объекта.

Выводы. Процедурная генерация является самым перспективным направлением в сфере разработки игр. Благодаря огромным возможностям по созданию контента и обширной реиграбельности, игры в большинстве своем, основанные на процедурной генерации имеют довольно большой жизненный цикл. Приведенные в данной статье программные решения позволяют внедрить в разрабатываемую игру множество элементов процедурной генерации.

Список литературы

1. Браун, С. Эволюционный игровой дизайн / С. Браун – Берлин: Спрингер, 2011. – 122 с.
2. Кук, М. Эстетические соображения для автоматизации проектирования платформ / М. Кук, С. Колтон, А. Pease // Конференция AAAI по искусственному интеллекту и интерактивным цифровым развлечениям – 2012. – С. 56-63.
3. Кук, М. Автоматизация игрового дизайна в трех измерениях / М. Кук, С. Колтон, Дж. Гоу // Материалы симпозиума AISB по искусственному интеллекту и играм. – 2014. – С. 20-24.
4. Официальный сайт компании NaturalMotion. — Сборник ЧаВо по продуктам NaturalMotion. <http://www.naturalmotion.com/faq.htm>
5. Современная терминология 3D графики. Раздел №4: Procedural Textures (Процедурные Текстуры) <http://www.ixbt.com/video2/terms2k5.shtml#pt>

УДК 004.02

ОСНОВНЫЕ АЛГОРИТМЫ РОЕВОГО ИНТЕЛЛЕКТА, ИХ СРАВНЕНИЕ И СФЕРЫ ПРИМЕНЕНИЯ

Лукашов С.А.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
serlukas94@gmail.com

Аннотация

Лукашов С.А. Основные алгоритмы роевого интеллекта, их сравнение и сферы применения. Представлена описание основных алгоритмов роевого интеллекта и практические примеры их реализации.

Ключевые слова: роевой интеллект, муравьиный алгоритм, метод роя частиц, пчелиный алгоритм, групповое поведение животных

Общая постановка проблемы. В статье рассматриваются алгоритмы роевого интеллекта. Особый акцент сделан на муравьином, пчелином алгоритмах и методе роя частиц. Рассмотрены их особенности, сферы применения и математический аппарат, их реализующий. На основании полученных данных в конце делается вывод о преимуществах перед другими способами реализации, даётся краткое заключение о перспективах развития, а также подведён некий итог разнообразным методам роевого интеллекта.

Исследования. Роевой интеллект описывает коллективное поведение децентрализованной самоорганизующейся системы. Рассматривается в теории искусственного интеллекта как метод оптимизации.

Существуют различные алгоритмы роевого интеллекта, однако наиболее распространенными являются метод частиц роя, муравьиный алгоритм и пчелиный алгоритм (рис. 1).



Рис.1 Общий график методов роевого интеллекта

Метод роя частиц (МРЧ) – алгоритм роевого интеллекта, для использования которого не нужно знать точный градиент оптимизируемой функции. МРЧ оптимизирует функцию, поддерживая популяцию возможных

решений, называемых частицами, и перемещая эти частицы в пространстве решений. Перемещения подчиняются принципу наилучшего найденного в этом пространстве положения, которое постоянно изменяется при нахождении частицами более выгодных положений.

Сам алгоритм выглядит так:

Пусть $f: \mathbb{R}^n \rightarrow \mathbb{R}$ — целевая функция, которую требуется минимизировать, S — количество частиц в рою, каждой из которых сопоставлена координата $x_i \in \mathbb{R}^n$ в пространстве решений и скорость $v_i \in \mathbb{R}^n$. Пусть также p_i — лучшее из известных положений частицы i , а g — наилучшее известное состояние роя в целом. Тогда общий вид метода роя частиц таков.

Для каждой частицы $i = 1, \dots, S$ сделать:

Сгенерировать начальное положение частицы с помощью случайного вектора $x_i \sim U(b_{lo}, b_{up})$, имеющего многомерное равномерное распределение. b_{lo} и b_{up} — нижняя и верхняя границы пространства решений соответственно.

Присвоить лучшему известному положению частицы его начальное значение: $p_i \leftarrow x_i$.

Если $(f(p_i) < f(g))$, то обновить наилучшее известное состояние роя: $g \leftarrow p_i$.

Присвоить значение скорости частицы: $v_i \sim U(-(b_{up}-b_{lo}), (b_{up}-b_{lo}))$.

Пока не выполнен критерий остановки (например, достижение заданного числа итераций или необходимого значения целевой функции), повторять:

Для каждой частицы $i = 1, \dots, S$ сделать:

Сгенерировать случайные векторы $r_p, r_g \sim U(0,1)$.

Обновить скорость частицы: $v_i \leftarrow \omega v_i + \varphi_p r_p \times (p_i - x_i) + \varphi_g r_g \times (g - x_i)$, где операция \times означает покомпонентное умножение.

Обновить положение частицы переносом x_i на вектор скорости: $x_i \leftarrow x_i + v_i$. Заметим, что этот шаг выполняется вне зависимости от улучшения значения целевой функции.

Если $(f(x_i) < f(p_i))$, то делать:

Обновить лучшее известное положение частицы: $p_i \leftarrow x_i$.

Если $(f(p_i) < f(g))$, то обновить лучшее известное состояние роя в целом: $g \leftarrow p_i$.

Теперь g содержит лучшее из найденных решений.

Параметры ω , φ_p , и φ_g выбираются вычислителем и определяют поведение и эффективность метода в целом.

Среди преимуществ стоит отметить довольно низкую алгоритмическую сложность в реализации, а также достаточно высокую эффективность для глобальной оптимизации.

На данный момент метод роя частиц применяется для задач машинного обучения, задачи оптимизации функций многих параметров, форм, размеров и топологий, а также в области проектирования, биоинженерии, биомеханике, биохимии. Развивать МРЧ можно, представив его в качестве многоагентной вычислительной системы. Также существует возможности включения в него других, более сложных методов РИ.

Муравьиный алгоритм (алгоритм оптимизации подражанием муравьиной колонии) – один из эффективных полиномиальных алгоритмов роевого интеллекта, использующийся для решения задач поиска маршрутов на графах, в частности для нахождения приближённых решений задачи коммивояжёра. Суть подхода заключается в анализе и использовании модели поведения муравьёв, ищущих пути от колонии к источнику питания и представляет собой метаэвристическую оптимизацию.

Муравьиный алгоритм моделирует многоагентную систему. Ее агентов условно называют муравьями. Каждый муравей хранит в памяти список пройденных им узлов. Этот список называют списком запретов или просто памятью муравья. Выбирая узел для следующего шага, муравей «помнит» об уже пройденных узлах и не рассматривает их в качестве возможных для перехода. На каждом шаге список запретов пополняется новым узлом, а перед новой итерацией алгоритма он опустошается. Кроме списка запретов, при выборе узла для перехода муравей руководствуется «привлекательностью» ребер, которые он может пройти. Она зависит, во-первых, от расстояния между узлами (то есть от веса ребра), а во-вторых, от следов феромонов, оставленных на ребре прошедшими по нему ранее муравьями.

Алгоритм описывается следующей формулой:

$$P_i = \frac{l_i^q * f_i^p}{\sum_{k=0}^N l_k^q * f_k^p},$$

где:

P_i – вероятность перехода по пути i ;

l_i – величина, обратная весу (длине) i -го перехода;

f_i – количество феромона на i -м переходе;

q – величина, определяющая «жадность» алгоритма;

p – величина, определяющая «стадность» алгоритма и $q+p=1$.

Одно решение не является точным и даже может быть одним из худших, однако, в силу вероятности решения, повторение алгоритма может выдавать (достаточно) точный результат.

Среди преимуществ муравьиного алгоритма стоит отметить высокую эффективность для задачи коммивояжера, возможность использования в

задачах, подверженных изменениям и низкую подверженность неудачным первоначальным результатам.

Помимо задачи коммивояжера, на данный момент муравьиный алгоритм используется для расчетов компьютерных и телекоммуникационных сетей, задачи раскраски графа и задач оптимизации сетевых трафиков. Развивать алгоритм можно, используя базы нечетких правил, а также гибридируя его с генетическими алгоритмами.

Алгоритм пчелиной колонии (алгоритм оптимизации подражанием пчелиной колонии) – один из полиномиальных эвристических алгоритмов роевого интеллекта для решения оптимизационных задач в области информатики и исследования операций. Относится к категории стохастических бионических алгоритмов, основан на имитации поведения колонии медоносных пчел при сборе нектара в природе. Предложен Д. Карабога в 2005 г.

Каждая пчела в рое рассматривается как частица или агент. Все частицы роя действуют индивидуально в соответствии с одним управляющим принципом: ускоряться в направлении наилучшей персональной и наилучшей общей позиции, постоянно проверяя значение текущей позиции. Аналогично местоположению пчелы на поле представляется координатами на плоскости $x-y$. Однако, в общем случае можно расширить эту идею в любое N -мерное пространство в соответствии с поставленной задачей. Это N -мерное пространство является областью решений для оптимизируемой задачи, где каждый набор координат представляет решение.

По аналогии с примером пчелиного роя функция пригодности будет плотностью цветов: чем больше плотность, тем лучше позиция. Функция пригодности служит средством связи между физической проблемой и алгоритмом оптимизации.

Каждая пчела помнит позицию, где она сама обнаружила наибольшее количество цветов. Эта позиция с наибольшим значением пригодности обнаруженная пчелой известна как персональная наилучшая позиция (ПНП). Каждая пчела имеет собственное ПНП, определяемое путем, который она пролетела. В каждой точке вдоль пути движения пчела сравнивает значение пригодности текущей позиции со значением ПНП. Если текущая позиция имеет значение пригодности выше, значение ПНП заменяется на значение текущей позиции.

Каждая пчела также каким-то образом узнает область наибольшей концентрации цветов, определенную всем роем. Эта позиция наибольшей пригодности известна как глобальная наилучшая позиция (ГНП). Для всего роя это одна ГНП, к которой стремится каждая пчела. В каждой точке на

протяжении всего пути каждая пчела сравнивает пригодность ее текущей позиции с ГНП. В случае если какая-либо пчела обнаружит позицию с более высокой пригодностью, ГНП заменяется текущей позицией этой пчелы.

Алгоритм может быть записан следующим уравнением:

$$v_n^{i+1} = w * v_n^i + c_1 rand() (p_n - x_n) + c_2 rand() * (g_n - x_n),$$

где: v_n^i – это скорость частицы в n-том измерении на предыдущем шаге;

x_n – это координата частицы в n-том измерении;

p_n – ПНП;

q_n – ГНП.

Среди преимуществ пчелиного алгоритма стоит отметить высокую скорость работы, а также возможность эффективного разделения на параллельные процессы.

Алгоритм роя пчел применяется для оптимизации управления и оптимизации классификаторов. Развитие алгоритма возможно с помощью снижения зависимости от устанавливаемых параметров и объединения с генетическими алгоритмами.

Выводы. Из трех рассмотренных алгоритмов каждый имеет свои преимущества над другими. Так алгоритм роя частиц является наиболее простым, алгоритм роя пчел – наиболее быстрым, а муравьиный алгоритм меньше всех страдает от неудачных первоначальных результатов. Кроме того, особую важность имеет поставленная задача. Так, например, для задачи коммивояжера муравьиный алгоритм будет наиболее эффективен, для задачи машинного обучения – метод роя частиц, а для оптимизации управления – пчелиный алгоритм.

Список литературы

1. Роевой интеллект и его наиболее распространённые методы реализации [электронный ресурс] // Молодой ученый: [сайт]. [2017]. URL: <https://moluch.ru/archive/138/38900/>

2. Роевой интеллект [электронный ресурс] // Академик: [сайт]. [2017]. URL: <http://dic.academic.ru/dic.nsf/ruwiki/1614280>

3. Муравьиный алгоритм [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: https://ru.wikipedia.org/wiki/Муравьиный_алгоритм

4. Естественные алгоритмы. Алгоритм поведения роя пчёл [электронный ресурс] // Хабрахабр: [сайт]. [2010]. URL: <https://habrahabr.ru/post/104055>

УДК 004.02

**ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ ДЛЯ СОЗДАНИЯ
ОНЛАЙН-СЛОВАРЯ****Лукашов С.А.**

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
serlukas94@gmail.com

Аннотация

Лукашов С.А. Использование онтологий для создания онлайн-словаря. Рассмотрена задача создания онлайн-словаря на основе онтологий. Описаны необходимые компоненты для создания словаря.

Ключевые слова: онтологии, онлайн-словарь, семантическая паутина, RDF-схема, OWL, XML, SPARQL.

Общая постановка проблемы. Электронные словари нужны чтобы иметь возможность быстро найти нужное слово, часто с учётом морфологии и возможностью поиска словосочетаний (примеров употребления), а также с возможностью изменения направления перевода (например, англо-русский или русско-английский). Эффективным методом создания такого словаря является использование онтологий в качестве базы данных.

Исследования. Онтология в информатике – это попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы.

Онтологии используются в процессе программирования как форма представления знаний о реальном мире или его части. Основные сферы применения – моделирование бизнес-процессов, семантическая паутина (англ. Semantic Web), искусственный интеллект.

Для создания словаря требуется использование семантической паутины и следующих ее компонентов:

- расширяемый язык разметки XML;
- общая схема описания ресурсов RDF;
- метаданные и схема RDF Schema (RDFS);
- онтология и языки ее описания OWL;
- язык запросов SPARQL к RDF-хранилищам.

Первый этап создания такого словаря – создание онтологии, которая будет выполнять роль базы слов. Для этого используется язык OWL.

OWL – Web Ontology Language, стандарт W3C, язык для семантических утверждений, разработанный как расширение RDF и RDFS. В основе языка – представление действительности в модели данных «объект – свойство». OWL пригоден для описания не только веб-страниц, но и любых объектов

действительности. Каждому элементу описания в этом языке (в том числе свойствам, связывающим объекты) ставится в соответствие URI. На данный момент актуальной считается вторая версия языка OWL, в которой определяются следующие разновидности. Для удобного создания онтологии на языке OWL используется программа Protege 5.0.

Для дальнейшего подключения и использования онтологии используются язык RDF, схемы RDFS и расширяемый язык разметки XML.

Язык Resource Description Framework (RDF) - система описания ресурсов Web. Разработан для того, чтобы описывать содержимое Web. В Semantic Web, когда говорят о каких-то сущностях Web, называют эти сущности ресурсами. RDF представляет собой язык для описания таких ресурсов. Ввиду того что описания семантики документов должны быть понятны компьютерам, необходимо разработать специальные программы-агенты, которые производили бы такое чтение. Также необходимо обеспечить возможность обмена информацией между различными программными агентами. Таким образом, под RDF подразумевается не только сам язык, но также и различные дополнительные программные модули, необходимые для обеспечения полноценного чтения и обмена информацией, записанной на этом языке. RDF сам по себе является не форматом файла, а только лишь абстрактной моделью данных, то есть описывает предлагаемую структуру, способы обработки и интерпретации данных. Для хранения и передачи информации, уложенной в модель RDF, существует целый ряд форматов записи. RDF обеспечивает семантический поиск в онтологии, используя формальные описания. А там, где есть формальные описания поисковый агент может искать факты и знания. RDF используется в качестве хранилища, к которому подключена онтология.

Схема RDF (RDF Schema, RDFS) представляет собой расширение языка RDF, позволяющее описывать простые онтологии данных, находящихся в хранилищах RDF. Так же, как схема базы данных описывает структуру базы данных в виде заголовков таблиц и связей между ними, схема RDF позволяет описывать структуру RDF-хранилища. Структура описывает хранилище в терминах типов и отношений между ними.

XML – расширяемый язык разметки, описывающий XML-документы и частично описывающий поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. XML используется для подключения RDF-хранилища к HTML-странице онлайн-словаря.

SPARQL используется для представления запросов к разнообразным источникам данных, независимо от того, хранятся эти данные непосредственно в RDF либо представляются в виде RDF с помощью промежуточного программного обеспечения (middleware). SPARQL обладает возможностями формирования запросов к обязательным и необязательным графовым шаблонам вместе с их конъюнкциями и дизъюнкциями. SPARQL также поддерживает тестирование расширенного значения и ограничение запросов посредством исходного RDF-графа. Результаты запросов SPARQL могут быть представлены результирующими наборами или RDF-графами. SPARQL используется в словаре для реализации функции вывода данных по запросу.

В плане внешнего устройства онлайн-словарь является HTML-страницей. Главными компонентами являются поле для ввода слова и поле для вывода результата по запросу. Также необходима кнопка для реализации запроса. Альтернативно можно сделать реализацию перевода сразу после ввода слова или же по нажатию клавиши. Дополнительными компонентами могут быть поле, дающее дополнительное описание введенному слову или же перечень синонимов. Для словаря-переводчика еще одной дополнительной функцией может быть возможность пользователю предложить улучшенный перевод.

Выводы. Была рассмотрена возможность использования онтологий для создания онлайн-словаря. Также были описаны средства, использующиеся при разработке, и задействованные языки программирования.

Онлайн-словари, в частности словари-переводчики остаются достаточно популярными и достаточно используемыми на данный момент и будут такими, как минимум, до тех пор, пока машинный перевод остается несовершенным.

Список литературы

1. Лексические онтологии wordnet в технологиях semantic web [электронный ресурс] // Interface.ru – internet & software company: [сайт]. [2014]. URL: <http://www.interface.ru/home.asp?artId=36209>
2. Язык запросов SPARQL для RDF [электронный ресурс] // W3C: [сайт]. [2008]. URL: <https://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>
3. Семантическая паутина [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2016]. URL: https://ru.wikipedia.org/wiki/Семантическая_паутина
4. XML [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: <https://ru.wikipedia.org/wiki/XML>
5. RDF Это просто [электронный ресурс] // Хабрахабр: [сайт]. [2010]. URL: <https://habrahabr.ru/post/94707/>

ИСПОЛЬЗОВАНИЕ АДАПТИВНОГО ТЕСТИРОВАНИЯ ПРИ ДИСТАНЦИОННОМ ОБУЧЕНИИ

Иванов А.А.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
alexey.iva94@gmail.com

Аннотация

Иванов А.А. Использование адаптивного тестирования при дистанционном обучении. В статье приведено исследование использования адаптивного тестирования при дистанционном образовании. Приведена классификации стратегий и условия входа и выхода при адаптивном тестировании.

Ключевые слова: адаптивное тестирование, дистанционное обучение

Постановка проблемы. Исследование состояния контроля знаний студентов с применением тестовых измерителей выявило определенные проблемы при использовании тестов: недостаточное качество и валидность содержания тестовых заданий, ненадежность результатов тестирования, недостатки обработки результатов по классической теории тестов, отсутствие использования современной теории обработки тестовых материалов с применением вычислительной техники [1].

Высокая погрешность измерения тестовых результатов не позволяет говорить о высокой надежности результатов измерения. Применение методов искусственного интеллекта и инженерии знаний может превозмочь субъективность и прямолинейность процесса тестирования, поднимая уровень оценивания знаний машинными системами.

Цель статьи – провести исследование использования адаптивного тестирования в условиях дистанционного обучения, определить классификацию стратегий адаптивного тестирования, установить условия входа и выхода при адаптивном тестировании.

Использование тестирования при дистанционном обучении. Тест – это средство, которое позволяет выявить уровень и качество усвоения. Применение традиционных методов компьютерного обучения и контроля (прямое тестирование, бальная система, и т. д.) имеет существенный недостаток: процесс взаимодействия оценивающего и оцениваемого не поддается строгой формализации, поэтому основные алгоритмические

функции не смогут описать в полной мере данную предметную область [2]. Это значит, что осуществление автоматизированного контроля знаний, умений обучаемых, в первую очередь, включает решение проблемы определения совокупности требуемых качеств знаний, без которых критерии оценки знаний и способы определения уровня их усвоения, выявить нельзя.

Чаще всего при дистанционном обучении можно увидеть тесты, работающие в режиме реального времени, данная система представленная на рисунке 1. В этом случае тестируемый отвечает на вопросы теста в режиме прямого диалога с компьютерной программой удаленного сервера.

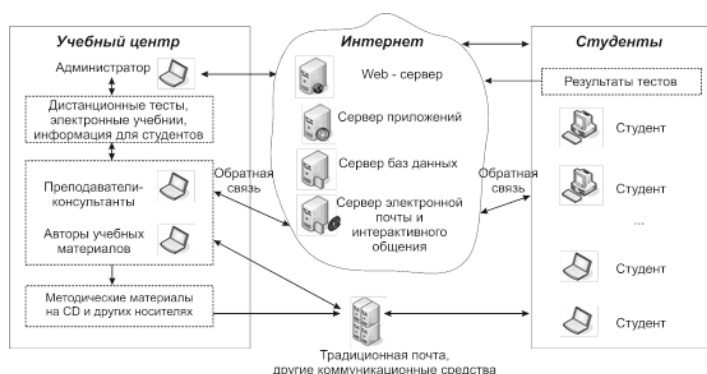


Рисунок 1. Общая схема обработки теста в режиме реального времени

Одно из важных требований при тестировании – наличие заранее разработанных правил выставления баллов. В общем случае применения тестов за правильный ответ в каждом задании дается один балл, за неправильный – ноль. Сумма всех баллов, полученных учащимся, дает число правильных ответов. Это число ассоциируется с уровнем его знаний и с понятием «тестовый балл испытуемого». Но существуют и другие, более сложные схемы оценивания, например рейтинговые. Дистанционное обучение предполагает разработку тестов главным образом в рамках критериально-ориентированного подхода, при котором задачей тестирования является сопоставление учебных достижений отдельного ученика с планируемым к усвоению объемом знаний, умений и навыков.

Понятие адаптивного тестирования. Под адаптивным тестовым контролем понимают компьютеризованную систему научно обоснованной проверки и оценки результатов обучения, обладающую высокой эффективностью за счет оптимизации процедур генерации, предъявления и оценки результатов выполнения адаптивных тестов [3].

Алгоритм отбора и предъявления заданий строится по принципу обратной связи, когда при правильном ответе испытуемого очередное задание выбирается более трудным, а неверный ответ влечет за собой предъявление последующего более легкого задания, чем то, на которое испытуемым был дан неверный ответ. Также есть возможность задания дополнительных вопросов по темам, которые обучаемый знает не очень хорошо для более тонкого выяснения уровня знаний в данных областях.

Таким образом, можно сказать, что адаптивная модель напоминает преподавателя на экзамене – если обучаемый отвечает на задаваемые вопросы уверенно и правильно, преподаватель достаточно быстро ставит ему положительную оценку. Если обучаемый начинает «плавать», то преподаватель задает ему дополнительные или наводящие вопросы того же уровня сложности или по той же теме. И, наконец, если обучаемый с самого начала отвечает плохо, оценку преподаватель тоже ставит достаточно быстро, но отрицательную.

Стратегии адаптивного тестирования. Стратегии предъявления тестовых заданий в адаптивном тестировании можно разделить на двухшаговые и многошаговые [4]. Двухшаговая стратегия предполагает наличие двух этапов. На первом этапе всем испытуемым выдается одинаковый входной тест, цель которого – осуществление предварительной дифференциации учащихся вдоль оси переменной измерения. По результатам дифференциации на втором этапе организуется адаптивный режим и строятся адаптивные тесты.

Многошаговые стратегии адаптивного тестирования подразделяются на фиксировано-ветвящиеся и варьирующе-ветвящиеся. Если один и тот же набор заданий с их фиксированным расположением на оси трудности используется для всех испытуемых, но каждый учащийся движется по набору заданий индивидуальным путем в зависимости от результатов выполнения очередного задания, то стратегия адаптивного тестирования является фиксировано-ветвящейся. Варьирующе-ветвящаяся стратегия адаптивного тестирования предполагает отбор заданий непосредственно из банка по определенным алгоритмам, которые прогнозируют оптимальную трудность последующего задания по результатам выполнения испытуемым предыдущего задания адаптивного теста. Таким образом, шаг за шагом из отдельных заданий получается адаптивный тест. В нем варьирует не только трудность, но и шаг, определяемый разностью трудностей двух соседних заданий адаптивного теста. Алгоритм, реализующий варьирующую стратегию адаптивного тестирования, носит циклический характер и представлен на рисунке 2.

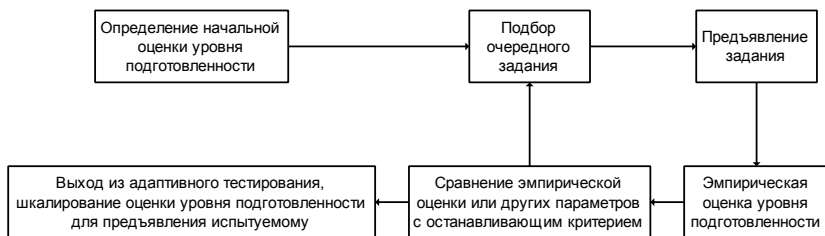


Рисунок 2. Алгоритм варьирующего многошагового тестирования

Вход и выход из адаптивного тестирования. Выбор начальных оценок для входа в адаптивное тестирование осуществляется в зависимости от вида стратегии и имеющихся технологических возможностей при генерации адаптивных тестов. Один из методов определения начальных оценок основан на выдаче испытуемым перед началом адаптивного тестирования входного предтеста. В предтест обычно включают 5–10 заданий из разлитых разделов содержания, охватывающих по трудности весь диапазон предполагаемого расположения тестируемой выборки учащихся на оси переменной измерения.

Для выхода из режима тестирования либо вводят ограничения по времени или по числу заданий, либо задаются планируемой точностью измерений. Ориентация на точность при организации адаптивных циклов порождает многообразие индивидуальных траекторий испытуемых, которые можно визуализировать в виде ломаных линий. Вершины ломаной линии соответствуют отдельным заданиям адаптивного теста, длина звена определяется варьирующим шагом, размер которого равен разности оценок параметра трудности двух смежных заданий адаптивного теста. Очевидно, что чем меньше длина ломаной, тем лучше структура знаний учащегося и эффективнее подобраны по трудности задания адаптивного теста.

На рисунке 3 изображены траектории адаптивного тестирования трех учащихся, начинавших свой вход в адаптивный режим по результатам выполнения предтеста.

Чем выше расположена вершина ломанной, тем труднее было первое задание адаптивного теста. На момент входа в предтесте самый высокий результат показал первый учащийся, поэтому он начинает адаптивное тестирование с более трудного задания.

Для удобства обсуждения результатов визуализации на рисунке приводятся непересекающиеся траектории. Над ломаными ставится «плюс» в тех случаях, когда испытуемый выполнил задание верно, или «минус», если испытуемый выполнил задание неверно. В качестве критерия окончания тестирования выбрано простое правило: тестирование прекращается, если учащимся подряд выполнены, верно, или неверно три задания адаптивного теста. Несмотря на высокий начальный результат, первый учащийся, по-видимому, обладает плохо структурированными знаниями, что следует из

чередования верных и неверных ответов. Тестирование первого ученика прекращается, если ему удастся справиться с идущими подряд тремя заданиями адаптивного теста. Траектория ответов второго учащегося намного короче благодаря хорошо структурированным знаниям. После неудачи при выполнении первого задания он все делает верно, и поэтому быстро заканчивает адаптивный тест. Третий учащийся самый слабый. Он начинает тестирование с наиболее легкого задания, с которым не справляется. Второе, более легкое задание он также выполняет неверно. Наконец, после трех следующих подряд неправильных ответов он выходит из адаптивного теста.

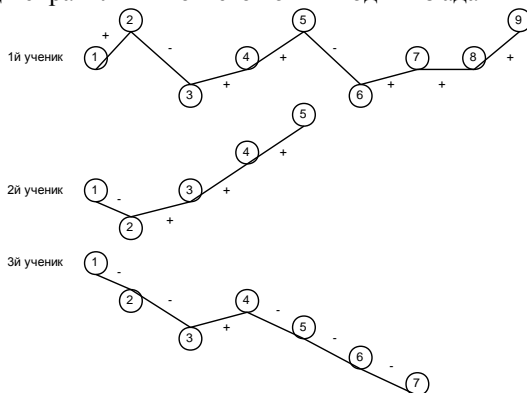


Рисунок 3. Визуализация индивидуальных траекторий испытуемых

Выводы. В статье было проведено исследование использования технологии адаптивного тестирования в дистанционном образовании. В результате была определена классификация стратегий адаптивного тестирования и критерии входа и выхода из него.

Список литературы

1. Новиков В.А., Кабанова Т.А., Разумовская Н.В. Тестирование как основной метод проверки знаний при дистанционном обучении./ Ж. Федерального центра тестирования, № 2 (14), Москва, 2005.
2. В.О. Любчак. Дистанционное обучение: опыт внедрения в украинском университете: монография / В.О. Любчак, О.В. Купенко, Т.В. Лаврик и др. – Сумы: СумГУ, 2009. – 160 с.
3. Зайцева, Л. В. Модели и методы адаптивного контроля знаний / Л. В. Зайцева, Н. О. Прокофьева. – Educational Technology & Society, 2004.
4. Чельшкова М.Б. Адаптивное тестирование в образовании (теория, методология, технология). – М.: ИЦПКПС, 2001.

УДК 519.71

**О ВЛИЯНИИ ПЕРЕБРОСКИ ДУГ НА ПОВЕДЕНИЕ
ОД-к-АВТОМАТА****А.А.Вергелес, О.М.Копытова**Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа

E-mail: Malinka0403@mail.ru

Аннотация

Вергелес А.А., Копытова О.М. О влиянии переброски дуг на поведение ОД-к-автомата. Рассматривается класс автоматов, полученных из ОД-к-эталона переброской одной или нескольких дуг при условии сохранения поведения. Под поведением понимается семейство множеств вход-выходных слов длины k , порождаемых всеми состояниями эталона. Найден автомат-характеризатор, по которому строятся все автоматы из указанного класса. Получены необходимые и достаточные условия принадлежности автомата рассматриваемому классу.

Ключевые слова: автомат, ОД-к автомат, поведение, переброска дуг.

Введение. В работе продолжается изучение автоматов, устойчивых по поведению к переброске дуг. Такое преобразование можно понимать как проявление "неисправности" функции переходов автомата при условии, что вход-выходные отметки дуг при этом остаются неизменными. Известно [1], что переброска в точности одной дуги в приведенном автомате всегда вызывает изменение его поведения – автомат становится не эквивалентным исходному автомату. Интерес представляет задача изучения условий устойчивости поведения приведенного автомата при перебросках более, чем одной дуги.

В [2] изучалось влияние переброски одной дуги на поведение инициального ОД-к-автомата с точки зрения оценки длины кратчайшего входного слова, различающего исходный и преобразованный автоматы.

В настоящей работе исследуется влияние переброски одной или нескольких дуг на поведение неинициального ОД-к-автомата. Целью работы является: 1) поиск в ОД-к-эталоне перебросок, в результате которых получается такой ОД-к-автомат, состояния которого порождают те же вход-выходные деревья высоты k , что и состояния эталона; 2) построение класса автоматов, соответствующих указанным переброскам и исследование их свойств.

Постановка задачи. Под автоматом будем понимать определенно-диагностируемый порядка k (ОД-к) конечный всюду определённый приведенный автомат Мили $A = (S, X, Y, \delta, \lambda)$, где S, X, Y – конечные

множества состояний, входных и выходных символов соответственно, δ , λ – функции переходов и выходов. Если $p=x_1x_2 \dots x_k$ – входное слово, то через $\delta(s,p)$ обозначается состояние, в которое автомат переходит из состояния s под действием p , а через $\lambda(s,p)$ – соответствующее выходное слово длины k . Слово $w=(p,q)=(x_1,y_1)(x_2,y_2) \dots (x_k,y_k)$, где $q= y_1,y_2 \dots y_k = \lambda(s,p)$, называется вход-выходным словом, которое порождается состоянием s и оканчивается в состоянии $\delta(s,p)$. Длину слова w обозначим $|w|$. Каждому состоянию s поставим в соответствие множество λ_s^k всех вход-выходных слов длины k , порождаемых этим состоянием. Это множество будем также называть деревом высоты k . Множество деревьев высоты k , порождаемых всеми состояниями автомата A , обозначим $\Lambda^k = \{\lambda_s^k | s \in S\}$ и назовём поведением автомата A . Там, где возникает неопределённость, k обозначениям δ , λ , λ_s^k , Λ^k будем добавлять нижний индекс A или значок штриха.

Как обычно, X^* обозначает множество всех входных слов конечной длины. Два состояния s и t одного и того же автомата A или двух разных автоматов A и B соответственно, называются эквивалентными, если для всякого входного слова $p \in X^*$ выполняется $\lambda(s,p)=\lambda'(t,p)$, где λ' – функция выходов автомата A или B . Автомат называется приведенным, если все его состояния попарно неэквивалентны. Напомним, что автомат A называется ОД- k -автоматом, если $k \geq 1$ – наименьшее целое такое, что для любого входного слова p длины k и любой пары различных состояний $s, t \in S$ выполняется неравенство $\lambda(s,p) \neq \lambda(t,p)$. Неопределяемые понятия теории автоматов можно найти в [3].

Автомат удобно задавать в виде графа переходов, вершины которого соответствуют состояниям из S , а дугами являются четверки (s,x,y,t) , где $t = \delta(s,x)$, $y = \lambda(s,x)$. Пара (x,y) называется отметкой дуги, s – её началом, а t – концом. Если $e=(s,x,y,t)$ – дуга в графе переходов автомата A , то переброской этой дуги из состояния t в состояние v , отличное от t , называем замену её дугой (s,x,y,v) .

Основные результаты. Пусть A – произвольный приведенный ОД- k -автомат (называемый эталоном), а автомат A' получен из A произвольной переброской в нем одной или нескольких дуг, т.е. автомат $A'=(S', X, Y, \delta', \lambda)$ отличается от эталона A функцией переходов. Состояния автомата A' будем пометать штрихом так, что переброшенная дуга в нём имеет вид $e' = (s', x, y, v')$. Нас будут интересовать такие переброски, при которых множество вход-выходных деревьев заданной высоты k остается неизменным. Будем говорить, что автомат A' сохраняет поведение на деревьях высоты k , если $\Lambda_A^k = \Lambda_{A'}^k$. Обозначим через $C_k(A)$ класс всех тех автоматов A' , полученных из A переброской одной или нескольких дуг, которые сохраняют поведение на

деревьях высоты k :

$$C_k(A) = \{ A' \mid A' \text{ получен перебросками дуг в } A \text{ и } \Lambda_{A'}^k = \Lambda_A^k \}.$$

Пусть w, w_1 – слова в некотором алфавите, а W, W_1 – некоторые множеств слов. Запись $w \leq w_1$ означает, что слово w является начальным отрезком слова w_1 , а запись $W \leq W_1$ – что каждое слово из первого множества является начальным отрезком некоторого слова из второго. Множество конечных отрезков тех слов из W , которые начинаются с w , обозначим $w \setminus W = \{ w_1 \mid ww_1 \in W \}$.

Построим по эталону A в общем случае недетерминированный автомат $H_k(A) = (S, X, Y, \Delta, \lambda)$, где $\Delta: S \times X \rightarrow 2^S$ определим следующим образом. Пусть в A есть дуга (s, x, y, t) , т.е. $\delta(s, x) = t$, $\lambda(s, x) = y$ и $(x, y) \setminus \lambda_s^k = \lambda_t^{k-1}$. Если некоторое состояние v порождает такое же дерево высоты $(k-1)$, как и состояние t , т.е. $\lambda_v^{k-1} = \lambda_t^{k-1}$, то к автомату A добавляем дугу (s, x, y, v) , полагая $t \in \Delta(s, x)$. Таким образом, в автомате $H_k(A)$ из состояния s исходят дуги с отметкой (x, y) во все те состояния, которые порождают дерево высоты $k-1$, совпадающее с λ_t^{k-1} .

Пусть A' получен в результате произвольной детерминизации автомата $H_k(A)$. Детерминизация состоит в том, для каждого состояния s и каждого входного символа x в автомате $H_k(A)$ удаляются все исходящие дуги с отметкой x , кроме одной. Заметим, что всякий такой A' есть результат некоторого множества перебросок дуг в эталоне. Назовём состояния s в автомате A и s' в автомате A' одноимёнными. Для одноименных состояний справедлива следующая

Теорема 1. Для любого A' , полученного детерминизацией $H_k(A)$, и каждого его состояния s' выполняется $\lambda_{s'}^k = \lambda_s^k$.

Доказательство проведём индукцией по высоте дерева. Так как отметки на дугах при переброске не изменяются, то $\lambda_{s'}^1 = \lambda_s^1$ для всех $s \in S$. Предположим, что $\lambda_{s'}^j = \lambda_s^j$ для всех $1 < j < k$. Покажем, что $\lambda_{s'}^k = \lambda_s^k$. Пусть для некоторого $x_k \in X$ выполняется $\delta(s, x_k) = t$, $\delta(s', x_k) = u'$, $\lambda(s, x_k) = \lambda(s', x_k) = y_k$. Из построения автомата A' следует, что в автомате A состояния t и u порождают одинаковые деревья высоты $k-1$. По индуктивному предположению, $\lambda_{u'}^{k-1} = \lambda_u^{k-1}$. Тогда $(x_k, y_k) \lambda_{u'}^{k-1} = (x_k, y_k) \lambda_u^{k-1}$. Объединение таких множеств по всем $x_k \in X$ приводит к одинаковым деревьям высоты k для одноименных состояний s и s' в автоматах A и A' соответственно, что и требовалось доказать.

Итак, с одной стороны, всякий A' , полученный детерминизацией $H_k(A)$, есть результат некоторых перебросок в эталоне, а с другой, он сохраняет

множество деревьев высоты k . Таким образом, из теоремы 1 вытекает

Следствие. Всякий A' , полученный детерминизацией автомата $H_k(A)$, приведен, является ОД- k -автоматом и принадлежит классу $C_k(A)$.

Рассмотрим свойства класса $C_k(A)$.

Теорема 2. Любые два автомата, полученные в результате различных детерминизаций $H_k(A)$, не изоморфны друг другу.

Доказательство. Пусть B и D получены указанным в условии способом. Покажем, что они не изоморфны. По теореме 1 они порождают то же множество попарно различных (в силу приведенности эталона) вход-выходных деревьев высоты k , что и сам эталон. Следовательно, всякое состояние s автомата B порождает дерево высоты k , отличное от всех деревьев, порождаемых состояниями автомата D , за исключением, возможно, одноименного состояния. Это значит, что оно отличимо от всех состояний автомата D , кроме, возможно, одноименного. Так как B и D получены различными детерминизациями, в них существует такая пара одноименных состояний s_B и s_D , которые по некоторому входному символу переходят в разноименные состояния, порождающие различные деревья высоты k (по теореме 1). Но тогда состояние s_B отличимо и от s_D , и значит, отличимо от всех состояний автомата D . Отсюда следует, что B и D не изоморфны.

Рассмотрим, в каком отношении между собой находятся класс $C_k(A)$ и класс автоматов, являющихся результатом детерминизации $H_k(A)$.

Напомним, что отображение множества состояний одного автомата во множество состояний другого автомата, сохраняющее функции переходов и выходов, называется гомоморфизмом. Формально, автомат $A=(S_A, X, Y, \delta_A, \lambda_A)$ гомоморфно вложим в автомат $B=(S_B, X, Y, \delta_B, \lambda_B)$, если существует такое отображение $\varphi: S_A \rightarrow S_B$, что для любых $s \in S_A$ и $x \in X$ выполняются равенства:

$$\varphi(\delta_A(s,x)) = \delta_B(\varphi(s),x);$$

$$\lambda_A(s,x) = \lambda_B(\varphi(s),x).$$

Теорема 3. Всякий автомат $A' \in C_k(A)$, отличный от эталона, является результатом некоторой детерминизации автомата $H_k(A)$, и наоборот, всякая детерминизация $H_k(A)$ приводит к автомату, отличному от A , из класса $C_k(A)$.

Доказательство. Пусть $A' \in C_k(A)$ и $A' \neq A$. Так как A' порождает то же множество деревьев высоты k , что и эталон, и все деревья различны, то легко видеть, что автомат A' гомоморфно вложим в характеристизатор $H_k(A)$ (состояние $s \in S_{A'}$, порождающее дерево λ_s^k , отображается на то состояние автомата $H_k(A)$, которое порождает такое же дерево). Поскольку $A' \neq A$, то в автомате $H_k(A)$ будут присутствовать дуги, которые имеются в A' и которых нет в эталоне, т.е. переброшенные дуги. Выбирая в процессе детерминизации именно эти дуги для тех состояний, из которых они переброшены, мы получим

автомат A' . Обратное утверждение сформулировано в следствии из теоремы 1. Теорема доказана.

Из теоремы 3 следует, что класс $C_k(A)$ в отличие от класса автоматов, получающихся в результате детерминизации $H_k(A)$, может содержать автоматы, изоморфные эталону. Другими словами, автомат $H_k(A)$ является характеристиком класса тех автоматов, которые получаются перебросками дуг в эталоне, сохраняют при этом поведение на деревьях высоты k и отличны от эталона.

На следующем рисунке показан эталон, являющийся ОД-2-автоматом, в котором переброшены 2 дуги из состояний 1 и 2 по входному символу 0. При этом новый автомат A' оказывается изоморфным исходному, а характеристизатор $H_k(A)$ совпадает с эталоном.

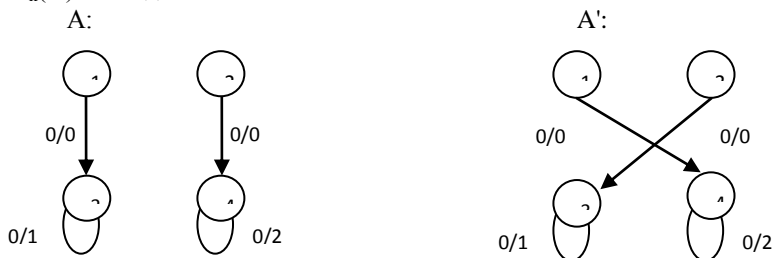


Рис. 1 – Пример переброски, которая приводит к изоморфному автомату

Выводы. В работе получены следующие результаты:

- 1) построен автомат-характеризатор, позволяющий получить все автоматы, являющиеся результатом переброски некоторых дуг в ОД- k -эталоне и сохраняющие деревья высоты k ;
- 2) показано, что результатом переброски может оказаться автомат, изоморфный исходному, который не обнаружим с помощью характеристизатора.

Список литературы

- 1 Грунский И.С., Копытова О.М. О структуре контрольного эксперимента для определенно-диагностируемого автомата // Теория управляющих систем.- К.: Наук.думка, 1987.- С.40-54.
- 2 Швец О.С., Копытова. О.М. О сравнении поведения ОД k -эталона и автоматов, порождаемых его локальными преобразованиями // Материалы V всеукраинской научно-технической конференции студентов, аспирантов и молодых ученых «Информационные управляющие системы и компьютерный мониторинг (ИУС КМ 2014)» (22.04 – 23.04.2014). Донецк: ДонНТУ, 2014. – С. 365 – 369.
- 3 Капитонова Ю.В., Кривий С.Л., Летичевский А.А., Луцкий Г.М., Печурин Н.К. Основы дискретной математики. – К.: Наукова думка, 2002. – 568 с.

Секция № 7
Медиаиндустрия и
дизайн



УДК 004.92

ПРОБЛЕМА АДАПТАЦИИ САЙТА ДЛЯ СЛАБОВИДЯЩИХ ЛЮДЕЙ

Сидорчук В.И., Харитонова В.В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
E-mail: vladimir.sidorchuk.96@mail.ru

Аннотация

Сидорчук В.И., Харитонова В.В. Проблема адаптации сайта для слабовидящих людей. Рассмотрена проблема юзабилити сайта человеком с плохим зрением, предложены решения проблемы, сделаны выводы.

Ключевые слова: сайт, юзабилити, медиапространство, веб-ресурс, адаптация, сайт.

Постановка проблемы. Каждый третий человек из общества имеет доступ в интернет, а каждый пятый среди них является слабовидящим. Поэтому, актуальной проблемой является адаптация сайтов для слабовидящих людей.

Основная часть. Большинство людей имеют плохое зрение, а у многих в результате использования очков быстро устают глаза, из-за чего просмотр веб-страниц может быть затруднительным.

Решением данной проблемы – добавление специального режима для слабовидящих, благодаря которому, люди смогут настроить сайт в соответствии с потребностями, при этом, не теряя информацию.

На примере сайта факультета, будет рассмотрен режим «Для слабовидящих» и все его настройки.

Для включения режима «Для слабовидящих» на сайте должна присутствовать специальная кнопка как на Рисунке 1. Значок должен быть расположен на видном месте, чтобы можно было бы с легкостью его найти. Также, значок должен быть достаточно крупным и легко заметен, чтобы человек со слабым зрением сразу его увидел.

Печатный текст – простой и универсальный способ получения информации. Отличительной чертой текстового формата в том, что его воспринимаемость можно улучшить: слабовидящему благодаря увеличению шрифта текста, слепому при помощи шрифта Брайля.

Режим «Для слабовидящих» содержит следующие настройки: размер шрифта, керлинг, интервал, цветовая схема и шрифт. Внешний вид режима можно рассмотреть на Рисунке 2.

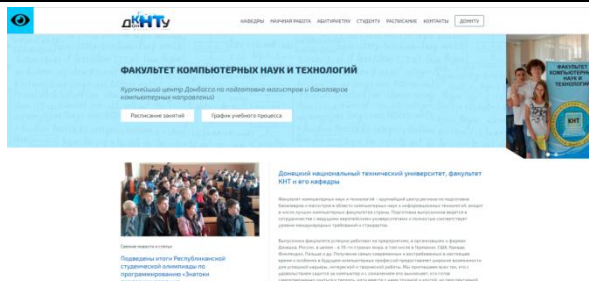


Рисунок 1 – пример сайта со значком режима «Для слабовидящих»

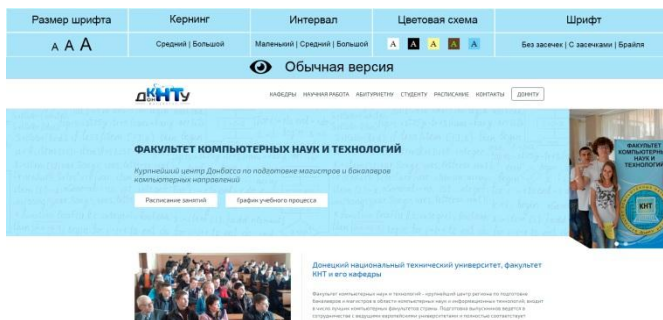


Рисунок 2 – внешний вид режима «Для слабовидящих»

В разделе «Шрифт» пользователь сможет увеличить размер шрифта. Шрифт становится крупнее, из-за чего слабовидящий пользователь сможет без проблем читать текст. Примеры можно увидеть на Рисунке 3.

В большинстве случаев, из-за статичного интервала между буквами, пользователям с плохим зрением бывает затруднительно читать, слова воспринимаются «склеенными». Поэтому, в режиме «Для слабовидящих» должен присутствовать раздел «Кернинг», в котором пользователь сможет изменить расстояние между буквами, для своего удобства. Примеры данного раздела на Рисунке 4.



Свежие новости и статьи

Донецкий национальный технический университет, факультет КИТ и его кафедры

Факультет компьютерных наук и технологий - крупнейший центр региона по подготовке бакалавров и магистров в области компьютерных наук и информационных технологий, входит в число лучших компьютерных факультетов страны. Подготовка выпускников ведется в сотрудничестве с ведущими европейскими университетами и полностью соответствует уровню международных требований и стандартов.

Выпускники факультета успешно работают на предприятиях, в организациях и фирмах Донецка, России, в целом - в 70-ти странах мира, в том числе в Германии, США, Канаде.



Свежие новости и статьи

Донецкий национальный технический университет, факультет КИТ и его кафедры

Факультет компьютерных наук и технологий - крупнейший центр региона по подготовке бакалавров и магистров в области компьютерных наук и информационных технологий, входит в число лучших компьютерных факультетов страны. Подготовка выпускников ведется в сотрудничестве с ведущими европейскими университетами и полностью соответствует уровню международных требований и стандартов.



Свежие новости и статьи

Донецкий национальный технический университет, факультет КИТ и его кафедры

Факультет компьютерных наук и технологий - крупнейший центр региона по подготовке бакалавров и магистров в области компьютерных наук и информационных технологий, входит в число лучших компьютерных факультетов страны.

Рисунок 3 – примеры размерности шрифтов

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ
 Крупнейший центр Донбасса по подготовке магистров и бакалавров компьютерных направлений
 Расписание занятий График учебного процесса



Свежие новости и статьи

Донецкий национальный технический университет, факультет КИТ и его кафедры

Факультет компьютерных наук и технологий - крупнейший центр региона по подготовке бакалавров и магистров в области компьютерных наук и информационных технологий, входит в число лучших компьютерных факультетов страны. Подготовка выпускников ведется в сотрудничестве с ведущими европейскими университетами и полностью соответствует уровню международных требований и стандартов.

Выпускники факультета успешно работают на предприятиях, в организациях и фирмах Донецка, России, в целом - в 70-ти странах мира, в том числе в

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ
 Крупнейший центр Донбасса по подготовке магистров и бакалавров компьютерных направлений
 Расписание занятий График учебного процесса



Свежие новости и статьи

Донецкий национальный технический университет, факультет КИТ и его кафедры

Факультет компьютерных наук и технологий - крупнейший центр региона по подготовке бакалавров и магистров в области компьютерных наук и информационных технологий, входит в число лучших компьютерных факультетов страны. Подготовка выпускников ведется в сотрудничестве с ведущими европейскими университетами и полностью соответствует уровню международных требований и стандартов.

Рисунок 4 – примеры разных кернингов

Аналогична ситуация с расстоянием между строками. Зачастую оно статично и человек испытывает проблемы с чтением текста из-за недостаточной величины интерлиньяжа. С помощью раздела «Интервал» человек сможет увеличить межстрочное расстояние, и тем самым улучшить восприятие текста. Примеры можно увидеть на Рисунке 5.

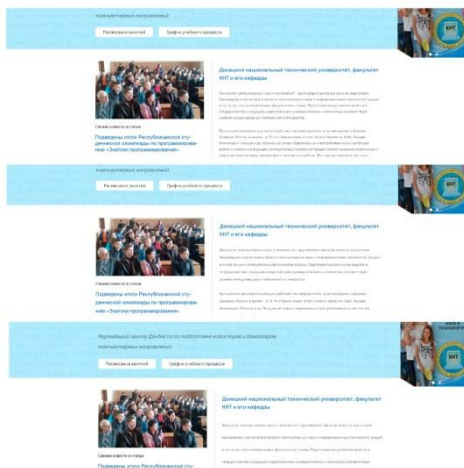


Рисунок 5 – примеры разных интервалов

В разделе «Цветовая схема» пользователь сможет изменить цветовую схему сайта на определенные цвета. Данный раздел особенно подходит для людей с отклонениями при восприятии цвета, например, дальтонизмом или пользователями, которые испытывают затруднения во время чтения в темноте. Примеры применения различных цветовых схем на Рисунке 6.



Рисунок 6 – примеры разных цветовых схем

Особое внимание стоит уделить разделу «Шрифт», в котором пользователь сможет включить шрифт с засечками или без, а также включить режим «шрифт Брайля».

Для передачи цифрового текста слепым используются специальные терминалы, работающие по схожему с тактильным шрифтом принципу. Для воспроизведения шрифта Брайля на бумаге используются специализированные принтеры или печатают на обыкновенной бумаге и с помощью грифеля выдавливают текст.



Рисунок 7 – примеры разных шрифтов

Выводы. С каждым днем количество пользователей в интернете увеличивается, а вместе с ними и количество слабовидящих людей. Для обеспечения удобства таким людям, был предложен режим «Для слабовидящих», в котором смогут настроить сайт в соответствии с потребностями, при этом, не теряя информацию. Были затронуты проблемы кернинга и междустрочного интервала на веб-ресурсах. Была рассмотрена проблема адаптации сайта для пользователей, у которых отсутствует зрение.

Список литературы

1. Рельефно-точечный шрифт Брайля [электронный ресурс] // В помощь родителям для незрячего ребенка [сайт]. [2017]. URL: <http://specposobie.narod.ru/index/0-27>
2. Адаптация сайта [электронный ресурс] // Хабрахабр [сайт]. [1994-2013]. URL: <https://habrahabr.ru/company/netcat/blog/165697/>

УДК 004.51

К ВОПРОСУ О ПОНЯТИЯХ UX UI ДИЗАЙНА

Дворянин Д.М., Харитонова В.В.

Донецкий национальный технический университет
кафедра компьютерного моделирования и дизайна
E-mail: dvoruanin@gmail.com

Аннотация

Дворянин Д.М., Харитонова В.В. К вопросу о понятиях UX/UI дизайна. Рассмотрены основные средства взаимодействия пользователя с функциональной и эстетической частью дизайна интерфейса предполагаемого программного продукта. Определены основные вопросы, решаемые UX дизайном и пользовательским интерфейсом (UI).

Актуальность темы

Разработка интерфейсов для различных приложений в последние годы вышла на лидирующее место среди IT профессий. Многие люди используют понятия UX и UI дизайн, не до конца понимая разницу между ними.

Перед разработчиками веб-интерфейсов ставится задача создания комфортного, дружелюбного по отношению к пользователю интерфейса. Главные требования здесь – удобство, практичность и интуитивная понятность. Именно в этот момент вступают в силу такие понятия как UX и UI дизайн, которые зачастую путаются.

Основная часть

User Experience Design (UX) в переводе означает «опыт взаимодействия» и включает в себя различные UX-компоненты: информационную архитектуру, проектирование взаимодействия, графический дизайн и контент. В целом, UX дизайн подразумевает комплексный подход к взаимодействию пользователя с интерфейсом, будь то веб-сайт, мобильное приложение или любая другая программа. Человек, который занимается этой работой – UX-дизайнер – при разработке интерфейса должен по возможности максимально учесть все мелочи, начиная от среды пользователя и типа электронного устройства и заканчивая способами ввода и отображения информации.

Основные вопросы, решаемые UX дизайном:

1. постановка целей и задач – чего в итоге необходимо достичь;
2. подбор подходящих UX инструментов для реализации целей;

3. разработка продукта, максимально удобного и легкого в восприятии целевой аудиторией;
4. анализ конечного результата – соответствует ли продукт ожиданиям заказчика и насколько высок уровень удовлетворенности пользователей.

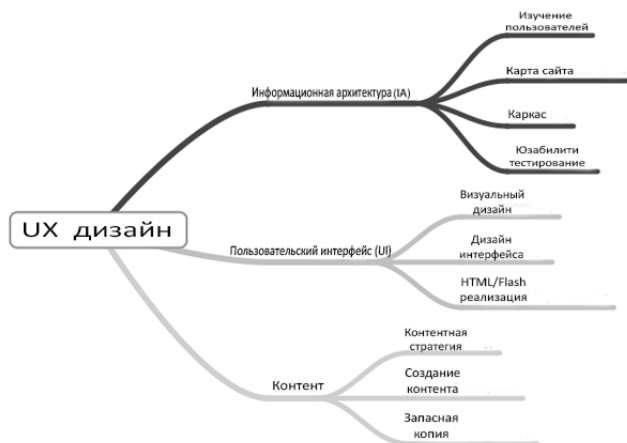


Рисунок 1 – Структура UX дизайна

User Interface Design (UI) или пользовательский интерфейс – это более узкое понятие, включающее в себя определенный набор графически оформленных технических элементов (кнопки, чекбоксы, селекторы и другие поля). Его задача – помочь пользователю организовать взаимодействие с программой/сайтом. На сегодняшний момент существуют некоторые правила UI дизайна:

1. Организованность элементов интерфейса. Это означает, что они должны быть логически структурированы и взаимосвязаны.
2. Группировка элементов интерфейса. Подразумевает объединение в группы логически связанных элементов (меню, формы).
3. Выравнивание элементов интерфейса. Сложно представить, что плохо выровненный интерфейс может быть для кого-то удобным.
4. Единый стиль элементов интерфейса. Стилевое оформление играет не последнюю роль, ведь именно оно сохраняется в памяти пользователя.
5. Наличие свободного пространства. Это позволяет разграничивать информационные блоки, сосредотачивая внимание на чем-то одном.

Полезный продукт удовлетворяет потребность, с которой еще не сталкивался рынок. Процесс исследования UX включает в себя конкурентный анализ, разработку персон, а затем создание минимально жизнеспособного

продукта; продукта, который будет ценным для определенной целевой аудитории. Это подтверждается путем тестирования на протяжении всего жизненного цикла продукта.

После того, как потоки пользователей и ваерфрэймы запрототипированы и протестированы, в процесс внедряется UI – его задачей будет заставить всё это красиво выглядеть. Это включает в себя выбор цветовой схемы и оформления, которое будет и красивым, и простым в использовании. Тем не менее, выбор цвета, оформление и взаимодействия основываются не на личных предпочтениях дизайнера, а на четко сформулированных причинах, специфических для персон, разработанных UX дизайнерами. С их помощью UI дизайнеры реализуют визуальную иерархию, которая послужит для пользователей проводником, объясняющими им что и когда делать, чтобы достичь своей цели.

Хорошо продуманная иерархия будет выделять одну главную цель на странице давая пользователям понять, где на сайте они находятся, и что они могут сделать в любой данный момент времени. Иерархия справится с этим при помощи конвенций и шаблонов, уже знакомых пользователям. Эти шаблоны будут подсказывать пользователям направление.

Дизайн пользовательских интерфейсов направлен на проектирование графических интерфейсов для техники различного назначения. Это не означает, что он ограничен графическим пользовательским интерфейсом компьютеров, планшетов, и мобильных устройств. В наши дни интерфейсы можно встретить во многих других продуктах, таких, как часы, стиральные машины, приборные панели в автомобилях, торговых автоматах, и многом другом.

В сети во множестве разбросаны примеры цифровых веб-продуктов с хорошим UI и плохим UX. Наиболее распространенные из них — длинные лид-формы с множеством полей для заполнения (Рисунок 2).

CONTACT US
 If you would like to get ahold of us, please fill in the form below...

Salutation (optional)

First and Last Name

Company or Organization (optional)

Email Address

Phone Number (optional)
 1 () - -

Fax Number (optional)
 1 () - -

Subject or Topic

Comments or Questions

Newsletter (optional)
 If Yes, I would like to receive a monthly newsletter about deals and offers!

CONTACT US
 If you would like to get ahold of us, please fill in the form below...

Name

Email Address

Comments or Questions

Рисунок 2 – пример хорошего и плохого UX/UI

Существуют веб-ресурсы, дающие посетителям некоторое представление о размещенном на них цифровом контенте с последующей просьбой о регистрации для просмотра всего содержимого электронной книги, журнала, статьи.

Некоторые ресурсы, предоставляющие информацию о проектировании интерфейсов UX/UI, обладают не лучшими интерфейсами, что когда-либо приходилось видеть. Эти сайты не следуют актуальным трендам в разработке и не стремятся к новейшим тенденциям в сфере веб-дизайна. Однако не смотря на эти недостатки, такие ресурсы предоставляют пользователю всю необходимую информацию, имеют понятную навигацию, являются источниками качественного контента. Примером служит UX Matters (Рисунок – 3).



Рисунок 3 – плохой UI хороший UX

На некоторых таких целевых страницах действительно наличествует приятный визуальный дизайн, следующий новейшим тенденциям разработки UI. Но пользовательский опыт, который они предоставляют, иначе как негативный, отталкивающий.

Необходимость регистрации для чтения даже бесплатных электронных книг/статей выглядит как бесцеремонное принуждение. Подобный подход, игнорирует свободную волю посетителя, в виду этого, на некоторых интернет ресурсах вы никогда не узнаете, что находится последующих страницах, не заведя аккаунт.

Подходящим примером хорошего UI и отвратительного UX: является интернет-магазин с дружественным и приятным интерфейсом, однако посетителю придется нажимать визуальный блок определенных товаров, чтобы перейти на другую страницу и только там узнать цену заинтересовавшего продукта. Так же неудобство доставляет факт того, что переход на карточку товара может осуществляться под средством кнопки «Купить сейчас», что вынуждает пользователя выполнять дополнительные не нужные действия при работе с ресурсом.(Рисунок – 4).

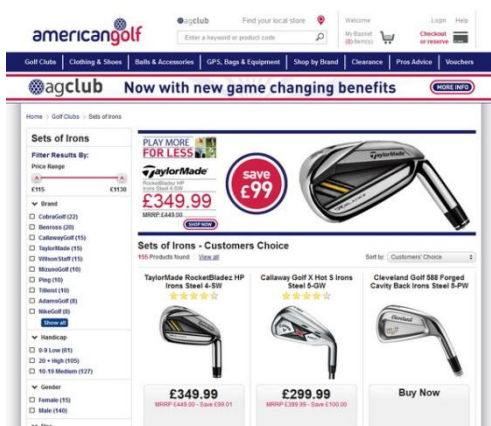


Рисунок 4 – хороший UI и плохой UX

Выводы

В ходе исследования было определено, что UI и UX — разные понятия. Однако, в процессе разработки они используются в паре, для достижения общей цели: заинтересовать пользователей в готовом продукте и/или передать им определенную информацию. Было выявлено, что UX может существовать и работать очень эффективно с довольно простым пользовательским интерфейсом. Также встречаются приложения, которые выглядят невзрачно, но оказываются очень удобными в использовании (плохой UI, хороший UX). Было выявлено, что UI отвечает за визуальную составляющую предоставления информации пользователю, а UX в свою очередь отвечает за функциональное восприятие и взаимодействие пользователя с ресурсом.

Тем не менее, современные тренды, тенденции и технологии в области дизайна UI направлены на одну цель: сделать онлайн UX лучше, проще и понятнее.

Список литературы

1. Аарон Уолтер. Эмоциональный веб-дизайн. 2012.
2. Гарретт Д. Веб-дизайн: книга Джесса Гарретта. Элементы опыта взаимодействия = The Elements of User Experience: User-Centered Design for the Web. — Символ-Плюс, 2008. — С. 192

УДК 004.51

АДАПТАЦИЯ ДИЗАЙНА ДЛЯ ЛЮДЕЙ С НАРУШЕННЫМ ЦВЕТОВОСПРИЯТИЕМ

Тилинина Н. Ю.

Донецкий национальный технический университет
кафедра компьютерного моделирования и дизайна
E-mail: natasha_tilinina@mail.ru

Аннотация

Тилинина Н.Ю. Адаптация дизайна для людей с нарушенным цветовосприятием. Рассмотрены основные приемы для адаптации и улучшения восприятия визуальной составляющей ресурсов, направленные на людей с различными типами цветовой слепоты. Представлены сайты и приложения с модификациями, упрощающими их использование для определенной группы пользователей.

Актуальность темы

Разработка пользовательского интерфейса и дизайна любого сайта или приложения основана, в первую очередь, на привлечение посетителей и впоследствии комфортное взаимодействие пользователя с ресурсом.

На данный момент, разной степенью дальтонизма страдают 7% мужчин и 0,4% женщин. Это говорит о том, что на 100 пользователей приложения или сайта, приходится около 7 человек, которые будут видеть контент не так, как изначально было запланировано. А неглижировать даже небольшой группой пользователей, означает такое же уменьшение эффективности ресурса.

Основная часть

Часть людей страдающих дальтонизмом способны различать вещи так же четко, как основная часть населения. Их проблема заключается только в неспособности различать зеленый, синий или красный цвет. Чаще всего встречается слепота в области красного цвета, когда люди будут путать синий и фиолетовый за неимением возможности определить красный элемент в фиолетовом цвете. Более редко встречается дейтеранопия, при которой человек не видит зеленый цвет, и тританопия – вид заболевания, который относится к тем, кто не может отличить синий или желтый цвет. На данном рисунке представлено как люди из каждого типа цветовой слепоты могут видеть радуугу.

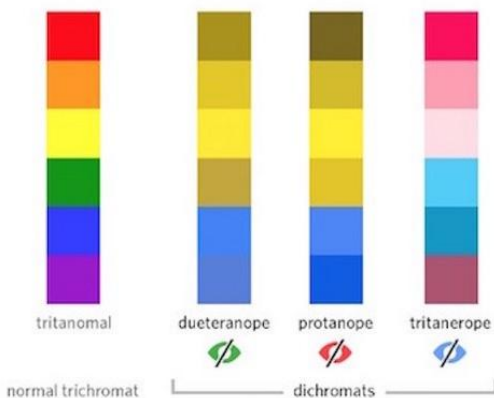


Рисунок 5 – Восприятие цветов людьми с разными типами дальтонизма

Дальтонизм индивидуален в каждом конкретном случае, потому может выражаться не только в искаженном восприятии некоторых цветов, но и неправильном восприятии геометрических форм.

Невозможно полностью адаптировать пользовательский интерфейс и дизайн для всех возможных проявлений дальтонизма, но элементы интерфейса удобные для людей с этим заболеванием, может стать хорошим улучшением дизайна и для восприятия остальной части пользователей.

Снизить искажение восприятия дизайна ресурса людьми с различными типами цветовой слепоты можно за счет:

1. применения белого фона и черного текста на нем;
2. максимального упрощения геометрических форм;
3. использования паттернов и текстуры;
4. отказа от авангардного дизайна;
5. использования фотографий и картинок с резкими контурами;
6. применения безопасных цветов и т.д.

Рассмотрим некоторые пункты на примере уже готовых ресурсов.

Одним из возможных решений проектирования доступного дизайна для дальтоников является соблюдение минимализма. Достаточно ограничить цветовую палитру, при создании ресурса, и это уже приведет к снижению вероятности создания путаницы. К тому же, чем проще геометрические формы, используемые в дизайне, тем меньше отличий в восприятии человека с нормальным зрением и человека с данным заболеванием. На рисунке изображен сайт с минималистическим дизайном, к тому же он предназначен

для того, чтобы конвертировать исходную цветовую схему сайта в цветовую схему для определенного типа дальтонизма.

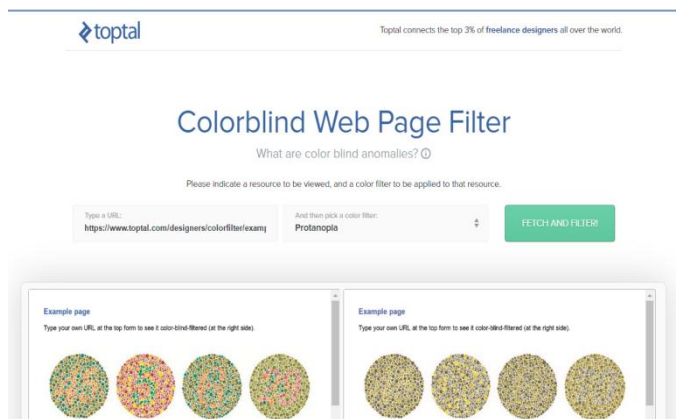


Рисунок 6 – Сайт для конвертации цветовой схемы

Еще один элемент, на который стоит обратить внимание при создании дизайна, доступного для людей с искаженным цветовосприятием, – это использование не только цвета, но и символов. К примеру, привычное для нас красное сообщение об ошибке может быть затруднительным или даже невозможно для прочтения людям с некоторыми разновидностями цветовой слепоты. В данном случае не стоит полагаться только на цвет, чтобы привлечь на себя внимание пользователей. Создатель социальной сети Facebook Марк Цукерберг является протанопом, то есть не различает красные и зеленые цвета, поэтому сайт создали именно в синих цветах, а также здесь очень хорошо разработаны поля формы и сообщения об ошибках.

Существует популярное приложение Two dots суть которого заключается в соединении точек одинакового цвета. Разработчики также позаботились о людях с цветовой слепотой и создали режим «color blind mode» при котором на точках используются определенные символы или рисунки для разных цветов.

Похожим образом создаются кнопки. Для людей с некорректным цветовосприятием может оказаться проблематичным распознавание активной кнопки сайта, если для её выделения использован только цвет. В таком случае разработчик может добавить к активному разделу подчеркивающую линию или другой графический элемент. Аналогичным образом можно изменять навигационное меню и переключатели.

Регистрация

Это бесплатно и всегда будет бесплатно.

Как вас зовут?

Дата рождения

Зачем указывать дату рождения?

Женцина
 Мужчина

Рисунок 7 – Форма регистрации Facebook

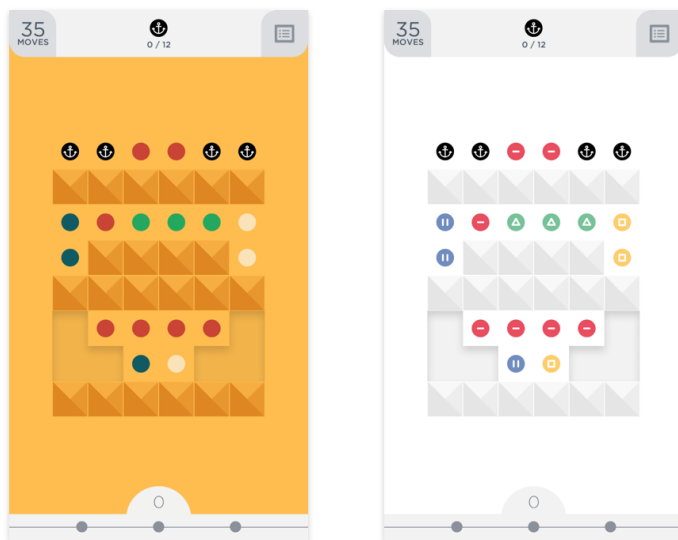


Рисунок 8 – Скриншоты игры Two dots

Необязательно использовать только черный и белый, как единственные безопасные контрастные цвета, существует ряд других цветов и оттенков которые можно с легкостью отличить независимо от наличия или отсутствия цветовой слепоты у пользователя. Примером применения такого решения является дизайн приложения Wordfeud. Все четыре цвета, используемые в

интерфейсе игры, входят в таблицу безопасных цветов, рекомендуемых для экранного дизайна. То есть все цвета будут переданы правильным образом, вне зависимости от монитора, не смешиваясь между собой. Таким образом, также предотвращается путаница цветов у дальтоников.

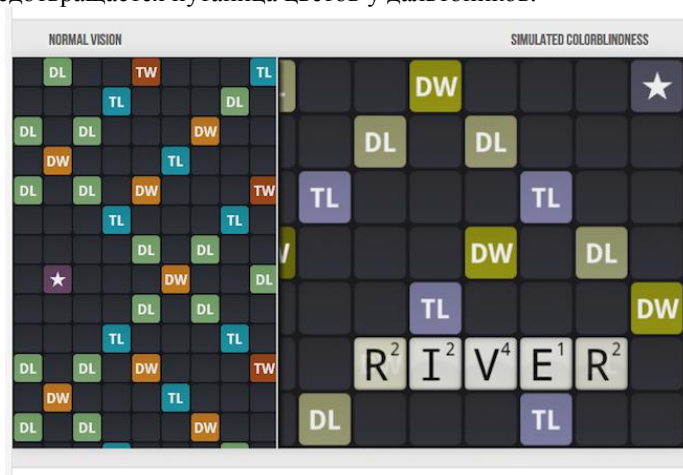


Рисунок 9 – Скриншоты игры Wordfeud

Выводы

В ходе исследования было определено, что невозможно полностью адаптировать ресурс под все типы заболевания без ущерба эстетической целостности дизайна. Однако, существуют определенные приемы в разработке программного интерфейса и дизайна для снижения разницы восприятия между людьми с цветовой слепотой и людьми со стандартным зрением. Были предоставлены сайты и приложения, которые помогают людям с некорректным цветовосприятием комфортно взаимодействовать с различными ресурсами.

Все это настроено на то, чтобы современные сайты и приложения были не только эффективными и коммерчески успешными, но и дружелюбными для всех пользователей.

Список литературы

1. Дежурка - Дизайн журнал №1/ Интернет-ресурс. - Режим доступа : [www/ URL: http://www.dejurka.ru/](http://www.dejurka.ru/) - Загл. с экрана.
2. Хабрахабр/ Интернет-ресурс. - Режим доступа : [www/ URL: https://habrahabr.ru/](https://habrahabr.ru/) - Загл. с экрана.

УДК 004.946

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОЧКОВ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Ю.О.Тахтаров, В.В.Харитонова

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
yura.tahtargov@mail.ru

Аннотация

Тахтаров Ю.О., Харитонова В.В. Сравнительный анализ очков виртуальной реальности. Выполнено сравнение различных гаджетов VR по ряду параметров, их актуальности на данный момент.

Ключевые слова: очки виртуальной реальности, Oculus Rift, компьютерные технологии, IT-индустрия.

Актуальность: В связи с быстрым развитием IT-индустрии, многие не успевают отслеживать новые тенденции, новые технологии. Таким образом, возникает необходимость оповещения о нововведениях которые возникают на рынке IT-индустрии. Одним из таких нововведений есть развитие и популяризация очков виртуальной реальности, их влияние на игровую индустрию и другие сферы жизни человека.

Цель статьи: рассмотреть устройства погружения в виртуальную реальности, их актуальность на данный момент, провести сравнительный анализ различных гаджетов VR по ряду параметров.

Очки виртуальной реальности. Развитие IT-индустрии сопровождается появлением на рынке самых необычных решений. Подобные устройства представлены в продаже в самом широком спектре разновидностей. Девайсы, о которых идет речь, предназначены для визуализации мультимедийного потока, обычно в формате 3D, с целью формирования у пользователя ощущения присутствия в игре или ином графическом окружении. Очки виртуальной реальности для ПК или мобильного устройства могут быть представлены в различных технологических модификациях. В ряде случаев их называют шлемами, если предполагается их исполнение в виде девайса, который нужно надевать подобно соответствующему приспособлению. Во многих контекстах термины «очки» и «шлем» не будут ошибкой, их можно и вовсе считать синонимами. Это новый уровень погружения. Даже если у вас есть очень большой дисплей, игра все равно заканчивается за его пределами. В очках виртуальный мир существует вокруг вас, куда бы вы ни посмотрели. Из-за искривляющих линз и отслеживающих датчиков устройства мозг воспринимает картинку не как изображение, а как окружение.

Исторические моменты. Известно, что гаджеты-VR появились до компьютерных технологий. Виртуальная реальность началась с идеи соединения восприятия действительности, звука и ощущения. Первый прототип очков виртуальной реальности разработал в 1968 году американский ученый Айвен Сазерленд. Устройство отображало примитивную графику, а его конструкция оказалась настолько тяжелой, что ее пришлось крепить к потолку. После этого многие компании, а также НАСА взяли на вооружение только что открытые технологии, что дало многомиллионное спонсирование. В начале 1990-х, когда виртуальная реальность считалась перспективной технологией, вышли очки для Sega Mega Drive и прототип Atari Jaguar VR. Но к концу десятилетия интерес к ним угас, а в 2000-х о виртуальной реальности забыли. Снова о теме заговорили только в 2012 году, когда 19-летний инженер Палмер Лаки и Джон Кармак, 42-летний разработчик игр Doom, Wolfenstein 3D и Quake вышли на Kickstarter с очками Oculus Rift. Фанат виртуальной реальности Лаки собрал первый прототип всего за год до этого — он хотел получить картинку лучше, чем у своих прошлых очков. Oculus Rift собрали на краудфандинговой платформе почти 2,5 миллиона долларов, что в десять раз превысило минимально необходимую сумму, а через два года Facebook неожиданно купил компанию Oculus VR за два миллиарда долларов. Тогда о новой виртуальной реальности начали говорить все.

Очки виртуальной реальности для ПК имеют как свои положительные характеристики, так и отрицательные.

Рассмотрим все по порядку. Плюсы:

- Возможность играть в игры, которые не предназначены для таких очков, но при этом передача 3D изображения будет не полной;
- За небольшой период времени уже выпущено несколько игр для 3D гейминга на ПК, которые стоят того, чтобы на них обратили внимание пользователи;
- Качество изображения, которое передают очки, будет очень хорошим, при условии, если ваш ПК оснащен мощным железом.

На этом плюсы заканчиваются, поговорим о минусах таких устройств:

- Не получится получить полноценную волю передвижения в таких очках, так как ваши движения ограничены длиной кабелей подключения;
- Для запуска игр и фильмов в таких очках, потребуется современный компьютер с мощными комплектующими;
- Такие очки предназначены только для работы с ПК.

Так что, если вы обладатель нового, современного мощного ПК, то стоит задуматься о покупке таких очков.

Анализ различные очки и шлемов VR. На данный момент существуют 7 различных гаджетов VR: Google Cardboard 2.0, Samsung Gear VR, Homido VR, Fibrum Pro, Oculus Rift, HTC Vive, Sony Playstation VR. В связи с таким

различием данные гаджеты можно сравнить гаджеты любой ценовой категории и составить их рейтинг на основе отзывов пользователей и специалистов. Сравнение представлено в табл. 1.

Таблица 1 – цена и рейтинг различных VR-гаджетов

Категория	Место	Наименование	Рейтинг	Цена
Лучшие недорогие очки и шлемы виртуальной реальности	1	Google Cardboard 2.0	9.0/10	500 P
Лучшие очки и шлемы виртуальной реальности средней ценовой категории	1	Samsung Gear VR	9.9/10	5 900 P
	2	Homido VR	9.7/10	3 900 P
	3	Fibrum Pro	9.5/10	6 000 P
Лучшие очки и шлемы виртуальной реальности премиум-класса	1	Oculus Rift	10/10	27 000 P
	2	HTC Vive	9.9/10	68 000 P
	3	Sony Playstation VR	9.8/10	30 000 P

Выводы. Произведен анализ VR гаджетов по цене и составлен их рейтинг на основе отзывов пользователей и специалистов. Результаты показали, что все гаджеты хороши и имеют средне высокий рейтинг от пользователей и специалистов. Согласно прогнозам, уже в ближайшем будущем объем рынка виртуальной реальности может составить более 15 миллиардов долларов в год. Эксперты убеждены, что быстрое развитие виртуальной реальности буквально перевернет всю игровую индустрию, и станет новой эрой в сфере развлечений. Кроме этого, виртуальная реальность может внести множество новых возможностей в ряд других отраслей деятельности человечества, где она может стать незаменимым инструментом в различных сферах. В результате, уже совсем скоро VR может стать неотъемлемой частью жизни для большинства людей на планете. В наше время люди все чаще стремятся перенестись из реального мира в мир реальности, подчиненной сознанию пользователя.

Список литературы

1. Виртуальная реальность [электронный ресурс] // Википедия – виртуальная реальность: [сайт]. URL: https://ru.wikipedia.org/wiki/Виртуальная_реальность
2. Как выбрать очки виртуальной реальности [электронный ресурс] // Лайфхакер: [сайт]. URL: <https://liferhacker.ru/2016/05/01/vr-headsets/>

УДК 004.921

МЕТОДИКА СОЗДАНИЯ 3D-ГРАФИКИ В GAME-ИНДУСТРИИ

Писарев Д.В., Харитонова В.В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна.
E-mail: daniil.pisarev.97@mail.ru

Аннотация

Писарев Д.В., Харитонова В.В. Методика создания 3D-графики в game-индустрии. Рассмотрено определение термина «игра», процессы создания компьютерных игр; основные этапы создания и визуализации 3D моделей в game-индустрии.

Ключевые слова: моделирование, 3D-графика, анимация, визуализация, риггинг.

Актуальность. На сегодняшний день ни один современный фильм и компьютерная игра не обходятся без трехмерной графики. 3D визуализация позволяет нам увидеть предметы, которых на данный момент не существует, либо нет возможности увидеть их в реальности. Когда объекты создает профессионал, достигается эффект присутствия и реальности. Разработчик, который создает трехмерный объект или сцену, должен обладать соответствующими навыками моделирования и придерживаться определенной последовательности в процессе.

Постановка задачи исследования. Определить последовательность этапов моделирования объектов и сцен в процессе создания компьютерных игр. Рассмотреть содержание каждого этапа.

Основная часть.

Игра – это интерактивное взаимодействие человека и виртуального мира. Поэтому можно выделить главные факторы которые учитываются при создании игры:

- интерактивность;
- бесперебойное функционирование;
- и только затем визуальный аспект.

Основные этапы создания и визуализации 3D моделей в game-индустриях

1. Моделирование – создание трехмерных объектов.
2. Текстурирование – наложение текстур и материалов на 3D-модели.

3. Риггинг – создание виртуального «скелета», набора «костей», «суставов» для последующей анимации персонажа.
4. Анимация - «оживление», анимирование трехмерного персонажа.
5. Рендеринг (3D визуализация) – визуализация созданной графики и запись.
6. Композитинг – объединение отдельных элементов в финальную сцену.

Моделирование.

Существует множество способов моделирования, рассказать обо всех в одной статье невозможно, поэтому мы затронем лишь самые популярные методы моделирования.

1. Способ моделирования.

В моделях для игр обычно используют полигоны (полигональное моделирование), потому что их проще всего визуализировать.

2. Количество полигонов у модели.

Чем больше полигонов у объекта, тем выше детализация и качество. В связи с этим выделяют высокополигональные (high poly) и низкополигональные (low poly) модели.

В играх используются низкополигональные модели, то есть это модели которые состоят из минимального числа полигонов. При этом их достаточное количество для визуального восприятия получаемого объекта. Часто в компьютерных играх встречается LOD – технология (Level of Detail – «уровень детализации»). Она состоит в упрощении 3D – моделей при помощи подмены их на более примитивные, когда игрок удаляется от них. Это разгружает систему, и она обрабатывает только те объекты, которые находятся в пределах видимости.

Для реализации этапов разработки модели существует большое количество разных программ для 3D моделирования. Такие как Autodesk Maya, Autodesk 3Ds Max и Cinema 4D. Также можно выделить Modo и Blender.



Рисунок 1. – пример низкополигональной (low poly) модели

Текстурирование.

Текстурирование – не просто подбор цвета и материалов для модели, это целое искусство, которым занимается отдельный специалист - художник по текстурам. Перед его работой моделер создает текстурную развертку (UV-развертка) – двумерное изображение, содержащее поверхность модели. Далее рисуются текстуры и привязываются к модели. Создается целый набор текстур: цвет, карта неровностей, карта нормалей, карта рельефа, карта бликов, карта прозрачности и многие другие. Таким образом, создается готовый визуальный образ модели или персонажа.

Создавать текстурные развертки и текстуры можно в тех же программах, что и модели. Но часто удобнее делать это в UVLayout.



Рисунок 2 – Пример наложение текстуры на 3D – модель

Риггинг.

Риггинг – создание «скелета», костей или суставов модели. Занимаются этим в game – индустрии художники по «оснастке» модели, «сетаперы». Сетаперы создают скелет внутри 3D модели и средства (контроллеры) для взаимодействия с этими костями, с помощью которых аниматоры могут «оживить» модель.

В играх обычно создается множество сложных контроллеров для аниматоров. Например, для лицевой анимации и мимики модели.

Для риггинга модели подойдут те же 3D-редакторы, о которых сказано выше.



Рисунок 3 – Пример скелета и манипуляторов 3D-модели

Анимация.

Чтобы «оживить» 3D модель за работу берутся аниматоры. Главная задача аниматора заключается в том, что бы сделать движение модели

максимально реалистичными и плавными. Простейшим методом анимации персонажей является анимация по ключевым кадрам (keyframes). Аниматор указывает положение персонажа в начальном и конечном кадрах, а промежуточные кадры вычисляются программой. Это с одной стороны простой способ, но достаточно трудоемкий, потому что для создания сложных движений требуется много времени, чтобы придать реалистичности персонажу. Также существует технология Motion Capture (система захвата движения). Она подразумевает под собой наложение движения реальных актеров на 3D персонажей. Эта технология во много раз упрощает анимацию.

Отличными инструментами для анимации являются 3Ds Max и Cinema 4D.

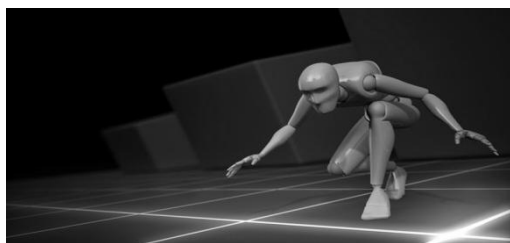


Рисунок 4 – Пример анимации 3D-модели

Рендеринг.

Существует два вида рендеринга – рендеринг в реальном времени и рендеринг не в реальном времени «пре-рендеринг». В компьютерных играх используется рендеринг в реальном времени. Реакции на действия игрока происходят моментально. Свет, цвет и тени формируются с помощью ранее просчитанных карт и текстур, а объекты перспективно проецируются на экран. Чтобы качество графики при этом не пострадало, в играх часто используются 3D ускорители. Главный критерий в игре – скорость выполнения подсчета.



Рисунок 5 – Пример итоговой визуализации (rendering) полученной сцены

Среди методов рендеринга можно выделить:

- растеризацию с методом сканирования строк (scanline, rasterization);
- трассировку лучей (raytracing);

- метод излучательности (radiosity).

Очень часто методы raytracing и radiosity комбинируются для достижения впечатляющих фотореалистичных результатов.

Можно выделить такие рендер-движки как: Mental Ray, V-Ray, Renderman.

Композитинг.

Композитинг является важным завершающим этапом постпродакшена. Это не просто работа над цветом и слоями: композитер должен собрать все части в единое целое, интегрирует съемочный материал трехмерных персонажей и другие 3D объекты, устраняет недочеты и убирает все лишнее, работает над эффектами. Одним словом это человек, который создает одну реалистичную сцену. Профессиональные программы для композитинга – Nuke, Adobe After Effects, Eyeon Fusion.



Рисунок 6 – Пример готовой сцены

Выводы. В результате исследования выявлено что, создание игры это продолжительный и трудоёмкий процесс, состоящий из самых разнообразных этапов таких как: моделирование, текстурирование, риггинг, анимация, рендеринг и композитинг, которые включают в себя как технические, так и творческие моменты. Этапы разработки, выделенные в процессе исследования, должны проходиться в указанной последовательности с использованием соответствующих профессиональных программ.

Список литературы

1. Компьютерные игры как искусство [электронный ресурс] // Этапы создания компьютерной игры: [сайт]. [2009-2017]. URL: http://gamesisart.ru/game_dev_create.html

2. Все о визуальных эффектах и моушн-дизайне [электронный ресурс] // Процесс создания 3D-графики: [сайт]. [2008-2017]. URL: <https://videosmile.ru/lessons/read/protsess-sozdaniya-3d-grafiki-v-filmah-i-igrakh.html>

3. Трехмерная графика [электронный ресурс]// Википедия – свободная энциклопедия: [сайт]. [1997-2017]. URL: https://ru.wikipedia.org/wiki/Трехмерная_графика

УДК 004.92

ОСОБЕННОСТИ ПСИХО-ФИЗИОЛОГИЧЕСКОГО ВОСПРИЯТИЯ ЦВЕТА ДЕТЬМИ ДОШКОЛЬНОГО И МЛАДШЕГО ШКОЛЬНОГО ВОЗРАСТА НА ПРИМЕРЕ ИНТЕРФЕЙСА ОБУЧАЮЩЕЙ СИСТЕМЫ "КОСМО"

Д.С. Кладько, В.В. Харитонова

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
this_is_not_email@mail.ru

Аннотация

Кладько Д.С., Харитонова В.В. Особенности психо-физиологического восприятия цвета детьми дошкольного и младшего школьного возраста на примере интерфейса обучающей системы "Космо". Рассмотрены основные особенности цветовосприятия детей дошкольного и младшего школьного возраста. Выполнен анализ программ аналогичной направленности. На основе анализа полученной информации найдено психологическое обоснование цвето-графического решения интерфейса обучающей системы «Космо».

Ключевые слова: цвет, интерфейс, ассоциация, обучающая система, навигация, программа, восприятие, воздействие.

Актуальность: В данный момент в сети Интернет существует большое число обучающих программ на космическую тематику, однако, далеко не каждая из них является понятной и полезной для детской аудитории младшего школьного возраста. Создание обучающей системы, интерфейс которой будет обусловлен психологическими особенностями восприятия целевой аудиторией, является приоритетной и актуальной задачей в современном сетевом пространстве.

Задачи исследования: Рассмотреть особенности психо-физиологии восприятия цвета детьми дошкольного и младшего школьного возраста. В соответствии с результатами научных исследований, подобрать и обосновать цвето-графическое решение интерфейса обучающей системы «Космо» для детей дошкольного и младшего школьного возраста. Провести предварительный анализ цветов с позиции их влияния на психику указанной целевой аудитории, вызываемые ими эмоциональные и поведенческие реакции.

Особенности цветовосприятия детей младшего школьного возраста:

Цвета, воспринимаемые человеком, позволяют познавать мир, общаться с другими людьми, эмоционально развиваться. Цвета обращаются к человеческим чувствам, а не к логике, а именно: они вызывают произвольную психологическую реакцию. Между цветовым решением интерфейса программ и естественным восприятием человека существует определенная зависимость. Достоверно установлено, что каждый цвет вызывает подсознательные ассоциации. Теплые цвета (красный, оранжевый, желтый) – активные и стимулирующие, бодрят, призывают к немедленным действиям. В то время как холодные цвета (синий, зеленый, белый) – успокаивают, оказывают тормозящее воздействие на нервную систему, создают приятную ровную атмосферу.

Восприятие цвета так же зависит и от формы, в которую заключен тот или иной цвет, рядом стоящие цвета, их пропорции. Для целенаправленного использования цвета необходимо знать устойчивые связи между цветом и психологической реакцией человека. Предпочтительное отношение к определенным цветам заметно проявляется в разных возрастных категориях. При разработке интерфейса для детей, как правило, важно в первую очередь учитывать тот факт, что интуитивное восприятие ребенка реагирует раньше, чем начинаются процессы осознания и мышления. Из этого следует, что цвет в данном случае – наиболее важный элемент дизайна обучающей системы, поскольку именно он первым передает пользователю информацию об объекте и создает первое впечатление. Дети младшего школьного возраста предпочитают яркие, теплые, насыщенные цвета, заключенные в простые формы – квадрат, круг. Яркие контрастные цвета вызывают интерес у ребенка, побуждают к исследованию и способствуют вовлечению в обучающий процесс. Именно на этом основании все детали обучающих программ для указанной возрастной категории детей младшего школьного возраста должны быть яркими, умеренно контрастными, желательно – с теплым подтоном. Все элементы интерфейса и навигации в обучающей программе создаются с учетом особенностей восприятия детей и должны отвечать возложенным на них задачам. Иконки и пиктограммы программы создаются максимально понятными для ребенка, с использованием чистых цветов, простых форм, а так же с дополнительно всплывающими подсказками при наведении на элемент, чтобы избежать негативного восприятия программы и повысить эффективность пользовательской работы с ней.

Рассмотрим аналоги обучающих систем, посвященных изучению космоса:

1. Celestia

Свободный симулятор космоса, позволяющий исследовать Вселенную в трех измерениях. Приложение дает возможность изучать Солнечную Систему

и более 100000 звезд за ее пределами, по сути, является каталогом галактик, планет, лун, астероидов, комет и космических аппаратов. Программа ориентирована на взрослую аудиторию, содержит большое количество сложной информации, достаточно однообразна по подаче информации. Интерфейс не предназначен для использования программы детьми, цветовые решения не продуманы в соответствии со спецификой их восприятия – темные цвета, много мелких деталей (см. рис.1). Программа не содержит в себе тестов для закрепления информации и проверки усвоенных знаний, что делает ее окончательно не рекомендуемой для детской аудитории.



Рисунок 1 – Интерфейс приложения Celestia.

2. Explorium: Space for Kids

Яркие оттенки, простой интерфейс и красота оформления делают эту программу привлекательной для детей (см.рис.2). Однако, данное приложение ориентировано на развлекательный аспект в большей степени, нежели на обучающий. Программа содержит в себе много мини-игр, которые дают мало информации и нацелены на веселое проведение досуга с незначительными элементами обучения. Кроме того, невозможно проверить качество и объём знаний, полученных в процессе взаимодействия с программой, так как тестов и проверочных вопросов нет.



Рисунок 2 – Интерфейс приложения Explorium: Space for Kids.

Поиск решения: При создании интерфейса обучающей системы «Космо» были учтены вышеперечисленные особенности восприятия детей младшего школьного возраста, в связи с чем было создано несколько поисковых эскизов, эволюцию которых мы увидим ниже.

В первоначальном эскизе присутствовала высокая контрастность фона, большое количество элементов, окрашенных в яркие цвета, много текстовой информации (см.рис.3).



Рисунок 3 – Первоначальный вариант интерфейса обучающей системы «Космо».

Во втором варианте эскиза черный (тревожный и мрачный для восприятия ребенка) фон заменен на более спокойный синий, способствующий расслаблению нервной системы. На новом фоне все цвета элементов стали выглядеть светлее, восприниматься менее агрессивными (см.рис.4).



Рисунок 4 – Второй вариант интерфейса обучающей системы «Космо».

В последнем варианте эскиза обучающей программы «Космо» было решено изменить расположение основных элементов, их пропорции, а так же полностью убрать текст, чтобы облегчить восприятие картинки. Фон был разбавлен более светлыми точками-звездами, что дополнительно добавило эскизу легкости и образности (см.рис.5).

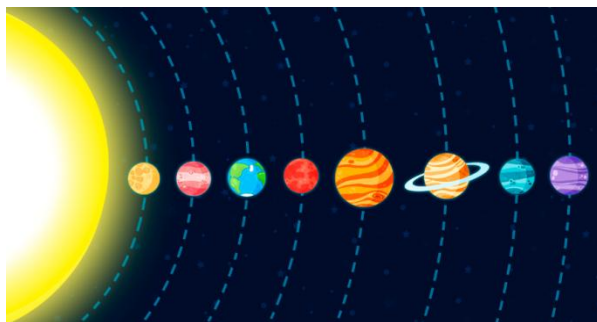


Рисунок 5 – Финальный вариант интерфейса обучающей системы «Космо».

Решение проблемы: На основе проведенных исследований было решено предпринять следующие шаги по оформлению обучающей системы «Космо»:

1. Использовать в качестве фона темно-синий оттенок, способствующий расслаблению нервной системы и улучшению восприятия информации у детей.
2. Использовать планеты в качестве основных интерактивных элементов обучающей системы: простая форма круга облегчает восприятие

объекта детьми младшего школьного возраста, и позволяет им выстраивать позитивные ассоциативные ряды.

3. Цветовое решение системы выполнено с использованием ярких, теплых, насыщенных оттенков, которые вызывают интерес у ребенка, побуждают к исследованию и способствуют вовлечению в обучающий процесс.
4. Использовать как можно меньше текстовых элементов для облегчения процесса восприятия информации

Выводы

В ходе проведения исследования было рассмотрено и подробно изучено несколько обучающих систем для детей на космическую тематику, а именно Celestia и Explorium: Space for Kids. В соответствии с результатами данных научных исследований, было найдено психологическое обоснование цветографического решения интерфейса обучающей системы «Космо» для детей дошкольного и младшего школьного возраста. Выявлено, что указанная аудитория предпочитает яркие, теплые, насыщенные цвета, заключенные в простые формы – квадрат, круг. Так же в ходе исследования было установлено, что яркие контрастные оттенки вызывают наибольший интерес у ребенка, побуждают его к изучению материала и способствуют вовлечению в обучающий процесс. По итогу исследований для оформления обучающей программы «Космо» выбраны простые формы интерактивных элементов (такие как круг) и яркие, чистые, теплые, контрастные цвета. (красный, желтый, оранжевый, розовый и т.д.). Таким образом, по результатам данной работы найдено фактическое подтверждение наиболее эффективных цветовых решений при создании интерфейса обучающих систем для детей.

Список литературы

1. Особенности цветовосприятия и цветоощущения. Физиологическое, оптическое и эмоциональное воздействие цвета [Electronic resource] / Интернет-ресурс. – Режим доступа : <http://www.dailyeducator.ru/moseks-380-1.html>
2. Особенности цветовосприятия детей дошкольного возраста [Electronic resource] / Интернет-ресурс. – Режим доступа : <http://allbest.ru/o-3c0b65635a2ac68b5c53a88521206d37-5.html>
3. Explorium – Космос для детей [Electronic resource] / Интернет-ресурс. - Режим доступа : <https://4pda.ru/forum/index.php?showtopic=622178>
4. Психология восприятия цвета детей [Electronic resource] / Интернет-ресурс. – Режим доступа : <http://www.proego.ru/articles/20>

УДК 004.5

РАЗРАБОТКА СТРУКТУРЫ ВЕБ-ОРИЕНТИРОВАННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ ДЛЯ СРЕДЫ LABVIEW

А.А. Штанько, А. Ю. Харитонов

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
shtanko.andrei@mail.ru

Аннотация

Штанько А.А., Харитонов А. Ю. Разработка структуры веб-ориентированной обучающей системы для среды LabVIEW. Проведен анализ сайтов-аналогов. Разработана внешняя и внутренняя структура веб-ориентированной обучающей системы для среды LabVIEW.

Ключевые слова: LabVIEW, веб-обучение, портал, внешняя структура сайта, внутренняя структура сайта.

Постановка проблемы: Среда программирования LabVIEW имеет большую популярность, используется в системах сбора и обработки данных, а также для управления техническими объектами и технологическими процессами, помогает справляться с множеством задач максимально быстро и эффективно. Чтобы обеспечить удобное обучение в этой среде программирования, необходима система, в которой можно найти все материалы, а так же возможность проверить свои знания в виде тестов. Лучшим решением этой задачи является сайт. Так как он будет ссылаться на множество интернет источников (видео уроки, статьи, примеры и т.д.).

Чтобы приступить к реализации необходимо выполнить следующее:

- проанализировать аналогичные веб ориентированные обучающие порталы и на основе полученных результатов разработать структуру сайта;
- спроектировать внутреннюю структуру сайта;
- спроектировать внешнюю структуру сайта.

Цель статьи: Провести анализ сайтов-аналогов и разработать внешнюю и внутреннюю структуру веб-ориентированной обучающей системы для среды LabVIEW.

Анализ веб-порталов Веб-портал — сайт в компьютерной сети, который предоставляет пользователю различные интерактивные интернет-сервисы, которые работают в рамках этого сайта. Веб-портал может состоять из

нескольких сайтов [1].

Среди русскоязычных порталов обучения LabVIEW были выделены следующие:

labviewportal.ru;
picad.com.ua.

В первом случае это сайт с множеством ссылок (уроки, загрузки, чат, галерея, ссылки, обратная связь) но из всего перечня функций работает только форум. Что не позволяет назвать этот сайт полноценным обучающим порталом.

Второй пример – это сайт-журнал, в котором собрано множество уроков и статей по изучению LabVIEW. Внешние ссылки и тестирование отсутствуют. Из положительных сторон этого сайта можно отметить значительное количество статей, касающихся всех аспектов работы в LabVIEW.

Все иностранные источники ссылаются на официальный сайт National Instruments, в котором есть множество уроков с самыми разными фильтрами (регион, цена, длительность курса). И в этот самый курс уже входят все необходимые материалы лекций, видеоуроков и т.д.

Анализ внутренней структуры сайтов: Структура сайта — основа для выстраивания последовательности и формы отображения имеющихся данных на сайте [2].

Линейная структура сайта – структура, в которой веб-страницы идут друг за другом, и пользователь просматривает их последовательно. В линейной структуре контент не разделяется на уровни. Сама по себе линейная структура является простой в реализации, но, не смотря на это, минусов в ней больше, чем плюсов и благодаря этому область применения тут четко ограничена. К примеру, такая структура может использоваться в имиджевых сайтах.

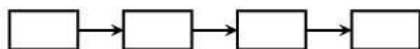


Рисунок 1 Линейная структура сайта.

Линейная структура сайта с альтернативами и вариантами – это структура, для которой основой является линейное размещение веб-страниц, но сам пользователь может использовать тот поиск информации, который ему удобен и идти по более понятному пути. В основном такая структура используется во время регистрации или покупки чего-либо.

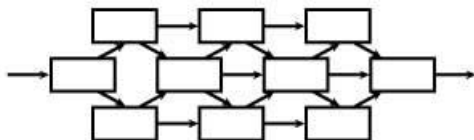


Рисунок 2 Линейная структура сайта с альтернативами и вариантами.

Линейная структура сайта с ответвлениями – это структура, в которой пользователь последовательно переходит с одной страницы на другую, но при этом возможны ответвления от основного линейного пути. Если часть информации показалась интересной, то чтобы узнать подробности гость может перейти, к следующему уровню, а потом вернуться обратно, к основной ветке.

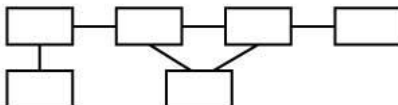


Рисунок 3 Линейная структура сайта с ответвлениями

Древовидная структура – это самая универсальная структура, которая подходит для сайтов любого типа. Принцип этой структуры заключается в том, что при попадании на главную страницу гость может выбирать, куда идти дальше. И выбрав нужный раздел, при необходимости уже выбирать подраздел.

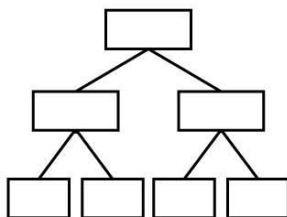


Рисунок 4 Древовидная структура сайта

Решетчатая структура – самая сложная из представленных структур. Страницы размещаются в разных ветках, но при этом пользователь имеет возможность перемещаться между ветками, которые находятся не только на одном уровне, но и на разных. Благодаря такому специфичному виду связи структуру эту используют крайне редко.

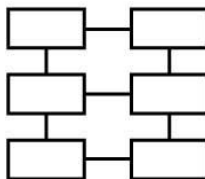


Рисунок 5 Решетчатая структура сайта

Анализ внешней структуры сайтов: Внешняя структура сайта – это некая блок-схема, которая определяет порядок размещения отдельных

элементов проекта на его страницах [4].

При разработке внешней структуры сайта важно представлять месторасположение всех информационных блоков. Разрабатывается внешняя структура на этапе создания макета сайта. Часто сайты создаются, используя самые смелые и нестандартные решения, но есть некие общепринятые рекомендации по поводу составления внешней структуры. К этим рекомендациям относятся: шапка, навигационное меню, основная часть и подвал.

Первое что видит любой пользователь, заходя на сайт – это шапка сайта. И первое впечатление о сайте создает именно она. Поэтому нужно разобрать этот аспект более подробно.

«Шапка – это блок в верхней части страницы сайта, в котором, как правило, размещается логотип и слоган сайта, краткая контактная информация, основное горизонтальное меню и другие элементы, которые считаются наиболее важными в зависимости от специфики ресурса» [5].

Чтоб создать правильную шапку сайта нужно быть осторожным с количеством элементов. Лучше, если в шапке вынесены только главные элементы, иначе пользователю будет сложно отыскать в этом изобилии данных, то, что ему нужно.

Важно так же, написать название сайта или спектра услуг, который предоставляет портал. Для этого может еще подойти логотип. Обычно его размещают в левом верхнем углу, так как люди привыкли читать слева на право, сверху вниз.

Создание внутренней структуры сайта: Исходя из рассмотренных аналогичных сайтов, и внутренней структуры сайтов, было принято решение создать древовидную структуру. Благодаря своей универсальности структура хорошо впишется в сайт

Таким образом, пользователь сможет находить любую информацию в три клика. Что является неким правилом для веб сайтов [6]. Понятная система уровней и удобная навигация поможет пользователю найти интересующую его информацию за максимально короткий срок.

Создание внутренней структуры сайта: При создании внешней структуры сайта решено было не отходить от общепринятой концепции и оставить привычные блоки: шапка, навигационное меню, основная часть и подвал. Обычный пользователь, который впервые попал на этот портал, легко сможет сориентироваться и найти необходимую информацию. Так же в шапке будет расположено название сайта: «Информационный портал по LabVIEW», что будет служить еще и ссылкой на главную страницу, а так же поможет пользователю точно определить назначение сайта.

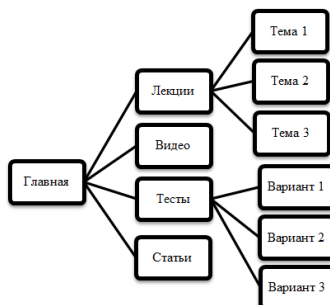


Рисунок 6 – Структура веб-ориентированной обучающей системы для среды LabVIEW



Рисунок 7 – Макет структуры «веб-ориентированной обучающей системы для среды LabVIEW»

Выводы: Рассмотрены аналогичные веб-порталы по LabVIEW, определены их положительные и отрицательные стороны. Рассмотрены возможные виды внутренних структур сайта. Разработана внешняя и внутренняя структура сайта, соответствующая основным правилам разработки структуры портала.

Список литературы

1. <https://ru.wikipedia.org/wiki/Веб-портал>
2. <http://sait-sozdat.ru/etap-sozdanie-sait/proekt-sait/sait-struktura-osnovnye-vidy.php>
3. Александров Е. Л. Интернет - легко и просто! Популярный самоучитель. - СПб.: Питер, 2005 - 208 с
4. http://www.aklychin.ru/view_post.php?id=59
5. <https://seopult.ru/library/Header>
6. https://ru.wikipedia.org/wiki/Правило_трёх_кликов

УДК 004.051

ИССЛЕДОВАНИЕ ОСНОВНЫХ ЭТАПОВ ВНУТРЕННЕЙ ОПТИМИЗАЦИИ САЙТА

А.Ю. Фролов, Грунский И.С.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
sovereignty@mail.ru

Аннотация

Фролов А.Ю., Грунский И.С. Исследование основных этапов внутренней оптимизации сайта. Выполнен анализ этапов внутренней оптимизации сайта. Определены основные шаги SEO-оптимизации чтобы сайт был максимально релевантен продвигаемым запросам.

Ключевые слова: внутренняя оптимизация, On-Page оптимизация, карта сайта, аффилиаты, дубликаты.

Постановка проблемы. В процессе разработки сайта рано или поздно наступает этап, когда стоит подумать о том как вывести его в топ в поисковых системах. Можно конечно попытаться продвигать сайт ссылками и статьями тратя на это большие деньги, однако это не очень эффективно. Гораздо правильней проводить внутреннюю оптимизацию сайта. Дело в том, что продвигать сайт в поисковых системах без внутренней seo-оптимизации сайта не имеет смысла, т.к. поисковые системы хорошо научились определять сайты для людей и сайты сделанные только для заработка. При плохой и не продуманной структуре сайта выйти в топ невозможно. Правда внутренняя оптимизация это процесс трудоемкий и содержащий множество этапов. Понять какие из них важны больше, а какие меньше не так просто, особенно новичкам в области web-разработки.

Цель статьи – провести анализ проблем продвижения сайта среди поисковых систем и выведения его в топ, и определение основных этапов внутренней оптимизации сайта.

Семантика и структура. Моментом №1 в SEO анализе и оптимизации является то, насколько структура сайта позволяет его продвигать. Довольно часто возникает такая ситуация, что клиент хочет или требует оптимизацию сайта под семантику, к которой сайт совершенно не готов структурно. На сайте может не быть нужной вложенности, категорий и подкатегорий, страниц

отзывов и прайсов, без которых эффективной оптимизации под интересную для клиента семантику не получится.

Поэтому, крайне важно параллельно выбирать и анализировать семантику, а также давать рекомендации и правки по структуре сайта.

Семантика + структура - очень серьезная связка. Лучше согласовать и внедрить все моменты сразу уже при начале разработки, чем когда уже выйдет готовый продукт и придется его переделывать. Но также важно помнить, что не любая семантика, которую предлагает или на которой настаивает клиент является правильной. В силу незнания он может быть убежден в одном, но в реальности дела могут обстоять иначе. Нивелируйте это своим погружением в бизнес и опытом.

Настройка карты сайта, ЧПУ и хлебных крошек. Правильная навигация важна для поисковых роботов. Он не будет тратить время на то, чтобы сканировать страницы с непонятным назначением, уровнем вложенности и неизвестной глубины. Ему необходимо видеть эту структуру и навигацию. Поэтому на сайте обязательно должны быть настроены человекопонятные url-адреса (ЧПУ).

Все url-адреса на сайте должны быть приведены к ЧПУ-виду и иметь вид `http://url/`. Кириллица должна трансформироваться в латиницу. В качестве разделителя слов в ЧПУ следует использовать дефис "-". Также будет неоспоримым плюсом, если в url-адресах будет соблюдаться полная вложенность, т.е. `http://url/>/`.

Пример не ЧПУ адреса может выглядеть так: `site.domain/___index.php?id=20`. Примером ЧПУ адреса для этой же страницы может быть `site.domain/name_page`.

После настройки ЧПУ следует также сделать 301 редирект с не-ЧПУ урлов на ЧПУ. В нашем случае это 301 редирект с `site.domain/index.php?id=20` на `site.domain/name_page`.

Настройка ЧПУшек и соответствующих редиректов решается подключением стандартных плагинов. Если речь идет о самописной системе, скорее всего нужно будет «допиливать» соответствующий функционал.

Под «хлебными крошками» (Breadcrumbs) подразумевается навигационная цепочка, которая отражает вложенность структуры и чаще всего располагается над или под основным текстовым заголовком страницы. К примеру : Главная / Каталог / Катриджы / Canon / Canon NP-7161.

Реализация Breadcrumbs полезна не только для того, чтобы показать поисковику глубину структуры сайта, но и с точки зрения юзабилити. Посетителю будет всегда удобно выйти на уровень или два уровня выше, допустим в каталог, если со страницей товара он полностью ознакомился, но хочет посмотреть что-то еще.

Карта сайта (sitemap) — документ, как правило, в формате xml, который нужен, чтобы улучшить индексацию страниц вашего сайта. Как правило, данные карты генерируются стандартными плагинами в зависимости от того, на какой CMS сделан сайт. Либо есть универсальные решения, например, специальная CMS для генерации карт, которая интегрируется в корень сайта и парсит всю его структуру, в результате чего и генерируется карта сайта. Адрес, по которому должна быть доступна карта, желательно делать `site.domain/sitemap.xml`.

Но и это еще не все. Необходимо после генерации карты отдать ее поисковому роботу в Google Webmasters Tools (GWT) в разделе «Файлы Sitemap» (см. рис. 1).

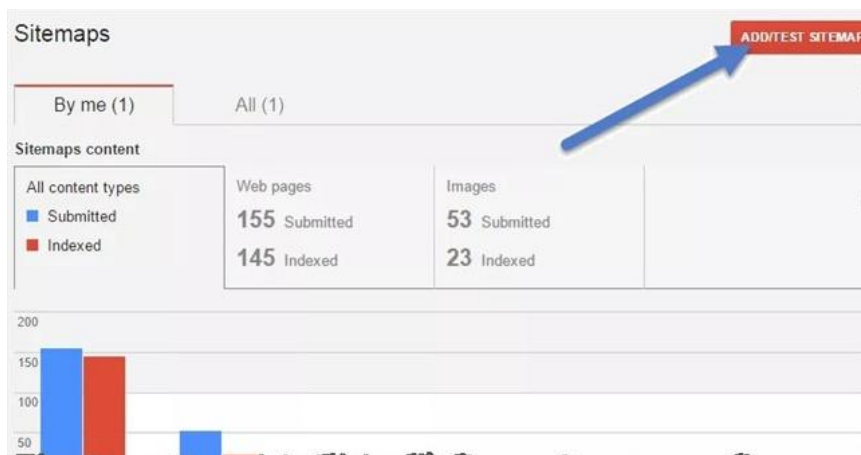


Рисунок 1 – Загрузка карты сайта поисковому роботу в GWT

Если с содержанием карты будет что-то не в порядке, Вы увидите соответствующие предупреждения об этом, которые потребуются исправить. Генерация карты сайта происходит в самом конце внедрения ТЗ оптимизации, когда устранены дубли, настроены правильные ЧПУ и т.д.

Диагностика и устранение дублей. От дублей нужно избавляться. И поисковые роботы в этом молодцы. Они такое не терпят, занижают рейтинг целевых страниц и сайта в целом, если в его содержимом содержатся дубли. Каким образом их диагностировать? Для этого есть несколько методов. Самый простой - это воспользоваться данными по повторяющимся метатегам в разделе «Оптимизация HTML» сервиса GWT (см. рис. 2).

Диагностировать дубли можно и с помощью программы NetpeakSpider. Такой вариант даже лучше в том плане, что он сканирует актуальную

структуру сайта, а страницы в GWT сканируются периодически и информация может быть не первой свежести на момент, когда Вы ищите дубли.

Панель инструментов

Сообщения (4)

Вид в поиске 1

- Структурированные данные
- Маркер
- Оптимизация HTML**
- Ссылки сайта
- Быстро загружающиеся страницы для мобильных устройств
- Поисковый трафик

Оптимизация HTML

Устранив следующие проблемы, вы сможете повысить качество и эффективность сайта.

Метаописание	Страницы
Повторяющееся метаописание	31
Длинное метаописание	0
Короткие метаописания	2

Тег заголовка	Страницы
Отсутствующие заголовки (теги title)	0
Повторяющиеся заголовки (теги title)	31

Рисунок 2 – устранение проблем дублей с помощью сервиса GWT

Дальше сам собой напрашивается вопрос о том, как нам избавиться от них. Если дубль полный, т.е является 100% клоном страницы, которую дублирует, стоит проставить 301 редирект с дубля на основную страницу и отправить данный дубль на принудительную переиндексацию в разделе «Просмотреть как Googlebot» (см. рис. 3).

Просмотреть как Googlebot

Посмотрите, как Googlebot видит вашу страницу. Подробнее...

Оставьте поле URL пустым, чтобы увидеть содержание главной страницы. Обработка запросов может занимать несколько минут.

ПК

СКАНИРОВАТЬ

ПОЛУЧИТЬ И ОТОБРАЗИТЬ

тут будет URL вашего сайта

просто жмем "Сканировать"

Рисунок 3 – отправка страницы на принудительную переиндексацию

Борьба с аффилиатами. Под аффилиатами подразумевается другой сайт клиента с такими же контактными данными, названием компании, структурой и тематикой. Поисковые системы “за естественную конкуренцию” и ведут борьбу с монополизацией рынка. Поэтому, склеивают аффилиаты, накладывая фильтры и занижая рейтинг сразу всех сайтов. Что в таком случае требуется делать? Алгоритм простой:

- аргументируем клиенту всю ситуацию, запрашиваем доступы к аффилиатам, проверяем их;
- добавляем каждый аффилиат в Google Webmaster Tools;
- настраиваем 301 редиректы со всех страниц аффилиатов на главную страницу основного сайта;

– отправляем на принудительную переиндексацию аффилиаты.

Если нет возможности проставить 301 редирект, нужно воспользоваться опцией в GWT в разделе «Удалить URL-адреса» (см. рис. 4).

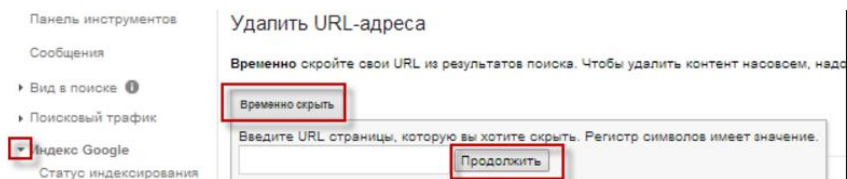


Рисунок 4 – удаление страницы аффилиата с помощью GWT

On-Page оптимизация. Подразумевает заточку и оптимизацию конкретных целевых страниц под семантическое ядро. Здесь важны следующие моменты:

- написание и размещение оптимизированных метатегов;
- подготовка и размещение полезного структурированного качественного текстового контента, при этом оптимизированного под нужные поисковые запросы;
- помимо контента текстового следует использовать качественные (желательно “не тяжелые” по объему памяти) изображения на всю ширину страницы. Не помешают также тематические видеоролики, которые точно повлияют на поведенческие факторы ПФ (снижат показатели отказов и увеличат время нахождения пользователя на странице).

Выводы. Выполнен анализ этапов внутренней оптимизации сайта. Определены основные шаги SEO-оптимизации. Проведен анализ важности сопоставления семантики со структурой сайта. Продемонстрирован способ настройки карты сайта, написания url-адресов и навигационной цепочки. Рассмотрено как с помощью Google Webmasters Tools бороться с дублями страниц и аффилиатами в целом.

Список литературы

1. Поисковая оптимизация [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: http://ru.wikipedia.org/wiki/Поисковая_оптимизация
2. SEO-продвижение [электронный ресурс] // студия разработки Abweb: [сайт]. [2017]. URL: <http://seo.abweb.com.ua/>
3. Frolov A. Y, Grunsky I. S. Research of the main stages of internal optimization of the website. The analysis of stages of internal optimization of the website is made. The main steps of SEO optimization are defined that the website was most relevant to the advanced inquiries.

УДК 004.9

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПЕРСОНАЛЬНЫМИ САЙТАМИ СТУДЕНТОВ С ВОЗМОЖНОСТЬЮ ИХ ВЕРИФИКАЦИИ

Павлий В.А., Воронков В.А.

Донецкий национальный технический университет
кафедра компьютерного моделирования и дизайна

E-mail: miletone@yandex.ru

Аннотация

Павлий В.А., Воронков В.А. Разработка информационной системы управления персональными сайтами студентов с возможностью их верификации. В статье представлена разработка системы управления персональными сайтами студентов с возможностью их верификации. Рассмотрены аналоги. Описаны поставленные задачи и пути их решения. Сделаны соответствующие выводы.

Ключевые слова: валидатор формата, веб-сайт, PHP, парсинг.

Постановка задачи. Сегодня разработка персонального сайта является одним из основных видов итогового контроля при изучении курса, посвященного web-технологиям. Однако, большое количество ошибок, возникающих при разработке сайта студентом, усложняет объективный контроль преподавателя. Вместе с тем, большая часть ошибок является в некотором смысле типовыми - они встречаются достаточно часто в большинстве студенческих работ.

Исходя из этого, необходима разработка системы, которая, с одной стороны, позволит опубликовать студенческий персональный сайт в сети Интернет, а с другой - позволит осуществлять контроль ошибок на этапе его создания и публикации.

Аналогом представленной системы служит веб-сайт «Портал магистров ДонНТУ» [1]. Но он не анализирует сайты на наличие ошибок кода и не предоставляет никакой информации об этих ошибках.

Описание системы. На этапе планирования разработки данной системы было решено реализовать её в формате веб-сайта с целью обеспечения простоты доступа к предоставляемым ею данным.

Веб-сайт – это совокупность логически связанных между собой веб-страниц; также место расположения контента сервера. Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователем как единое целое [2].

Для разработки используется язык программирования PHP, т.к. в открытом доступе отсутствуют плагины, реализующие необходимый системе функционал.

Используя FTP-клиент, студенты загружают свои персональные сайты на сервер в папку с открытым доступом, которая является временным хранилищем.

В системе реализован интерфейс администратора, который позволяет преподавателю перенести студенческий сайт на постоянное хранение, если в нём отсутствуют ошибки.

Для интеграции в систему студенческий персональный сайт в корневой директории должен содержать файл «meta.txt» заранее определенной структуры, содержащей основную информацию о студенте:

- ФИО;
- дата рождения;
- путь расположения фотографии;
- адрес электронной почты.

Для экономии места на сервере, система должна следить, чтобы общий объем персонального сайта студента не превышал некоторый допустимый предел (в среднем допускаются значения от 1 до 3 Мб). Если объем сайта превышает указанное значение, система должна выводить соответствующее сообщение об ошибке.

Как следствие из этого, на студенческом сайте не должны публиковаться музыкальные и видео файлы.

Ширина всех страниц персональных сайтов студентов не должна превышать 1000 пикселей. Это требование введено с целью обеспечения правильного отображения контента сайтов на не широкоформатных мониторах и мониторах с низким разрешением.

Студенческий персональный сайт в своей директории не должен содержать специальный файл «thumbs.db», который используется операционными системами компании Microsoft для хранения эскизов изображений для режима «Эскизы страниц». Он создается в каждой папке, которая содержит изображения, при первом отображении её содержимого.

В директории персональных сайтов также должен отсутствовать файл «.htaccess». Он является файлом-конфигуратором серверов Apache, который дает возможность конфигурировать работу сервера в отдельных директориях, например, устанавливать права доступа к файлам в директории.

Также одними из основных требований к разработке персонального сайта (которое студенты часто игнорируют) является наличие, как минимум, 10 мета-тегов на каждой html-странице и отсутствие ошибок в структуре html-документа.

Таким образом, полный перечень ошибок, которые должна контролировать система, следующий:

- отсутствие ФИО в файле «meta.txt»;
- отсутствие даты рождения в файле «meta.txt»;
- отсутствие пути расположения фотографии в файле «meta.txt»;
- отсутствие адреса электронной почты в файле «meta.txt»;
- отсутствие файла «meta.txt»;
- присутствуют файлы недопустимого формата;
- присутствует файл «thumbs.db»;
- присутствует файл «.htaccess»;
- отсутствие или недостаточное количество мета-тегов;
- присутствуют ошибки в структуре html-документа;
- ширина страниц сайта превышает 1000 пикселей;
- размер сайта превышает 3мб.

Разработка системы. Дизайн и пользовательский интерфейс системы выполнены с помощью стандартных средств HTML и CSS.

Для реализации задачи анализа html-файлов персональных сайтов студентов на наличие ошибок программного кода используем валидатор формата [3]. Он находится в открытом доступе и сертифицирован консорциумом Всемирной паутины. Консорциум Всемирной паутины - это организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины [4]. Валидатору отправляется запрос на анализ html-файла, в ответ приходит страница с результатами анализа. Результаты следует отделить от остального контента, размещённого на странице, путём парсинга её содержимого. Парсинг используется и для определения количества метатегов в html-коде страниц.

Правильность содержания файла «meta.txt» проверяется с помощью регулярных выражений.

Проверка объёма памяти, занимаемого сайтом на сервере, определяется с помощью функции filesize(), которая является стандартной в языке PHP.

После копирования сайта по FTP система его автоматически подхватывает и формирует ссылку на него на соответствующей странице. Вся необходимая информация для этого берётся из файла «meta.txt».

Если система обнаружила какие-либо ошибки, то рядом с ссылкой на соответствующий сайт отобразится значок восклицательного знака. При нажатии на этот значок выведется список ошибок. Все ссылки группируются по студенческим группам (таким образом, структурно система должна представлять собой нечто вроде студенческого портала).

Общий алгоритм реализации функционала системы показан на рисунке 1.

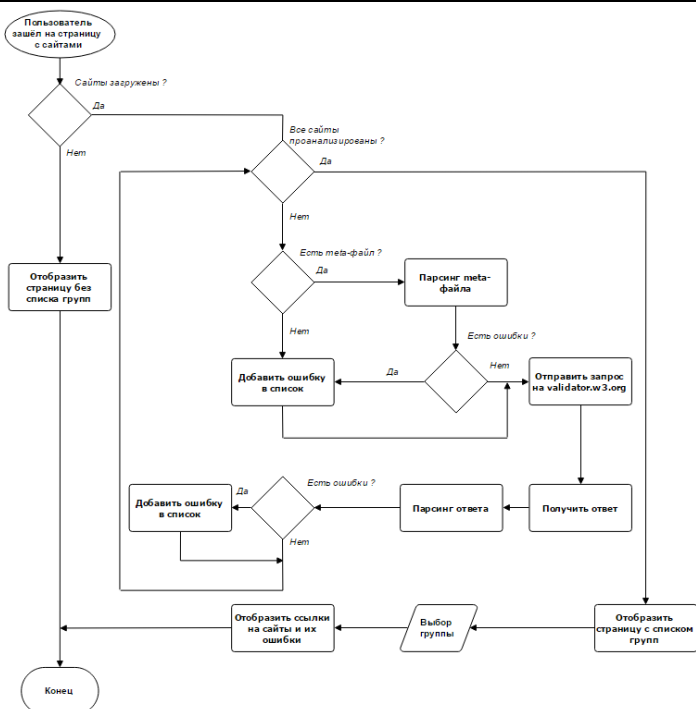


Рисунок 1 – Алгоритм реализации функционала системы

Выводы. В заключение отметим, что сейчас автоматизация является неотъемлемой частью нашей жизни. Этот процесс прошёл путь от преимущества до необходимого стандарта и приветствуется в различных аспектах человеческой деятельности. Поэтому планируется организация работы представленной в данной статье системы на общедоступном домене и её дальнейшее развитие.

Список литературы

1. Портал магистров ДонНТУ [электронный ресурс] – URL: <http://masters.donntu.org>
2. Wikipedia - Сайт [электронный ресурс] – URL: <https://ru.wikipedia.org/wiki/Сайт/>
3. The W3C Markup Validation Service [электронный ресурс] – URL: <https://validator.w3.org>
4. World Wide Web Consortium (W3C) [электронный ресурс] – URL: <https://www.w3.org>

УДК 004.9

РАЗРАБОТКА СТРУКТУРЫ ДЛЯ WEB-РЕСУРСА «ИНТЕРНЕТ-МАГАЗИН АКСЕССУАРОВ В ДНР»

Е.В. Лях

Донецкий национальный технический университет, г. Донецк
 кафедре компьютерного моделирования и дизайна
 liakheveniy@yandex.ua

Аннотация

Лях Е.В. Разработка структуры для web-ресурса «Интернет-магазин аксессуаров в ДНР». Выполнен анализ и сравнение существующих интернет-магазинов в данном направлении. Определены популярные разделы сайтов, разработана оптимальная структура для создания ресурса.

Ключевые слова: интернет, магазин, структура web-сайта, аксессуар.

Постановка проблемы. С развитием сферы информационных технологий все большее количество пользователей активно использует всемирную сеть для приобретения товаров абсолютно разных категории непосредственно в интернете.

Современная мода и активная пропаганда на стильный внешний вид постепенно наращивает численность населения, интересующегося различным аксессуарами (сумки, очки и т.д.). В связи с этим возникает смысл создания web-ресурса, на котором пользователь сможет посмотреть и, конечно же, приобрести аксессуары совершенно разных категорий.

Для реализации этого необходимо выполнить следующее:

- проанализировать распространенные интернет-магазины, выделить их достоинства и недостатки;
- определить популярные категории товаров, изучить отзывы посетителей;
- представить созданную с учетом полученных сведений структуру портала.

Цель статьи – проектирование интернет-магазина аксессуаров в Донецкой Народной Республике для удобства приобретения товаров данной категории.

Анализ интернет-магазинов. Интернет-магазин – сайт, торгующий товарами посредством сети Интернет.

Аксессуар – необязательный предмет, сопутствующий чему-либо; принадлежность чего-либо.

Существующие в мире и соседних странах интернет-магазины аксессуаров пользуются популярностью, а владельцы таких ресурсов зарабатывают на этом. По данным сайта pexht-invest.ru заработок таких магазинов может быть совершенно разным, но в среднем магазины, которые работают на протяжении нескольких лет зарабатывают около 150 000 тыс. рублей в месяц.

Название	Трафик	Доход (в мес.)
Интернет-магазин одежды «Васька»	197	8 000 руб.
Сервис приема платежей и обменный пункт «WMer»	200	52 000 руб.
Интернет-магазин детских товаров (более 9 лет)	904	550 000 руб.
Интернет-магазин ортопедических товаров (медицина)	28	105 000 руб.
Smoky-Tale. Интернет-магазин кальянной продукции и табака для кальяна	8	2 000 руб.

Рисунок 1 – пример интернет-магазинов, посещаемости (трафика) и дохода.

Из данной статистики вытекает актуальность разработки интернет-магазина в Донецкой Народной Республике.

Определение популярных разделов. Анализ пяти среднестатистических интернет-магазинов аксессуаров позволяет извлечь информацию о самых популярных категориях. С помощью этих данных мы поймём какие категории товаров нужно внедрить в наш собственный интернет-магазин. Сравнительная характеристика приведена в рисунке 2.

Исходя из анализа существующих интернет-магазинов аксессуаров, а также учитывая предложения пользователей, была создана структура «Интернет-магазин аксессуаров». Целесообразным является внедрение всех пунктов таблицы, кроме категории «бутылочки», а также дополнительная категория «опросник», в которой пользователю будет предложено ответить на несколько вопросов, чтобы улучшить работу сайта и подобрать ассортимент, под своих покупателей. Структура интернет-магазина аксессуаров проиллюстрирована на рисунке 3.

Разделы	Название ресурса				
	a-shop	garms	acces sorize	lady- new	issasho p
Очки	+	-	+	-	+
Чокеры	+	-	+	-	+
Сумки, рюкзаки	+	+	+	-	+
Бутылочки	+	-	-	-	-
Кошельки, клатчи, зажимы	-	+	+	+	+
Часы	-	-	-	+	+
Ремни	+	-	-	-	+
Контакты	+	+	+	+	+

Рисунок 2 – Сравнительная характеристика популярных интернет-магазинов аксессуаров.



Рисунок 3 – Структура web-ресурса «Интернет-магазин аксессуаров в ДНР»

Максимальной популярностью пользуются разделы: Сумки, рюкзаки и кошельки, клатчи, зажимы, а также контакты. Большинство сайтов не имеют взаимодействия с пользователем в интерактивном виде, внедрение которых и станет особенностью интернет-магазина. Посетителям ресурса будет предложено поучаствовать в опросах касательно ассортимента интернет-магазина.

Разделы интернет-магазина находятся в меню и попасть в них можно из любой страницы:

1. Главная;
2. Очки;
3. Чокеры;
4. Сумки, рюкзаки;
5. Кошельки, клатчи, зажимы;
6. Часы;
7. Ремни;
8. Контакты.

Данная структура является удобной и интуитивно понятной как для опытных, так и для начинающих пользователей. Она основана на опыте существующих сайтов и порталов, отзывах посетителей, а также психологии проектирования пользовательских и человеко-машинных интерфейсов.

Выводы. Развитие моды в Донецкой Народной Республике порождает необходимость облегчения приобретения аксессуаров, создания интернет-магазина в данной тематике. Разработка web-ресурса «интернет-магазин аксессуаров» позволит пользователю посмотреть каталог и приобрести аксессуары совершенно разных категорий в сети Интернет.

Список литературы

1. Веб-портал [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: www.ru.wikipedia.org/wiki/Веб-портал
2. Аксессуар [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: www.ru.wikipedia.org/wiki/Аксессуар
3. Next-invest.ru. Сколько зарабатывают интернет-магазины [электронный ресурс] // Next-invest.ru: [сайт]. [2017]. URL: www.next-invest.ru/skolko-zarabatyivayut-internet-magazinyi.html
4. Ян В. Уайт. //Редактируем дизайном – СПб: Университетская книга, 2009
5. А. Уолтер. //Эмоциональный веб-дизайн – М.: Манн, Иванов и Фербер, 2012 г.

УДК 766

ИСПОЛЬЗОВАНИЕ ЦВЕТА В РАЗРАБОТКЕ АЙДЕНТИКИ БРЕНДА

Кандаурова А.О., Харитонова В.В.

Донецкий национальный технический университет
кафедра компьютерного моделирования и дизайна

E-mail: rafalelka97@gmail.com

Аннотация

Кандаурова А.О., Харитонова В.В. Использование цвета в разработке айдентики бренда. В статье рассмотрено общее понятие айдентики бренда, ее главные задачи и составляющие. Приведены основные рекомендации для подбора цвета при разработке айдентики. Также рассмотрен пример создания айдентики производственного предприятия.

Ключевые слова: айдентика, бренд, цвет, цветовосприятие.

Актуальность исследования. Цвет на подсознательном уровне вызывает у зрителя определенные впечатления и эмоции, которые влияют на оценку и формирование образа того или иного бренда, поэтому его правильный выбор с точки зрения восприятия целевой аудиторией является одним из ключевых вопросов разработки айдентики бренда.

Задачи исследования. Рассмотреть значение понятия «айдентика» и его составляющих. Выделить особенности подбора цветового решения в процессе формирования имиджа бренда. Обосновать выбор цветовой гаммы айдентики производственного предприятия на примере «Донпромкабель».

Понятие айдентики. Айдентика является одной из составляющих частей брендинга. Под ней понимают набор всех элементов, отвечающих за визуализацию и эмоционально-смысловое восприятие конкретного бренда. Исходя из понятия айдентики, определяются ее основные составляющие:

- логотип – основной элемент построения айдентики, поскольку является главным графическим символом бренда;
- фирменный стиль – единый визуальный образ бренда, создаваемый вокруг логотипа, и впоследствии проявляющийся в каждом носителе айдентики;
- брендбук – специальное руководство, которое содержит полное описание бренда, в частности, подробные инструкции к

использованию каждого элемента фирменного стиля на различных носителях, как рекламных, так и корпоративных;

- дополнительные средства связи и действия.

Айдентика бренда ставит перед собой несколько основных задач:

- создание и закрепление узнаваемого образа бренда у потребителей, который будет понятен, и легко идентифицируем;
- выделение бренда среди конкурентов соответствующего сегмента рынка, привлечение к нему внимания.

Использование цвета в айдентике. Цветовосприятие играет важную роль при создании айдентики бренда. Правильный выбор цветового решения для бренда положительно влияет на формирование позитивного имиджа и возникающих эмоций. Грамотно подобранный цвет может решить следующие основные задачи:

- обеспечение заметности бренда среди конкурентов, для достижения которой важно проанализировать используемые цвета ключевых конкурентов и лидеров рынка;
- выделение нового товара бренда, которое заключается в использовании для него новых цветов, с целью привлечения внимания целевого потребителя;
- усиление невербальной связи потребителя с образом бренда;
- формирование положительного восприятия и доверия у новых потребителей.

Цвет, используемый в айдентике, определяет, как именно потребитель воспринимает бренд продукта, то есть отражает ли он индивидуальность бренда среди конкурентов. Поэтому при определении цветовой гаммы разрабатываемой айдентики бренда следует учесть некоторые рекомендации:

- следует учитывать территориальные культурные особенности, поскольку в разных культурах цвет может иметь кардинально разные значения;
- цвет – часть создаваемого визуального образа бренда, которая при недостаточном внимании к разработке других составляющих не сможет привести бренд к успеху;
- использование определенного цвета для всех носителей айдентики способствует однозначному и правильному пониманию бренда потребителем;
- поскольку цвет является необходимым элементом для создания индивидуальности бренда, то рекомендуется использовать цветовую гамму, которая будет выделять бренд среди конкурентов;
- следует уделить внимание уместности сочетания бренда с выбранным цветом;

В разработке айдентики бренда, как правило, используется ограниченное количество цветов, посредством которых передается определенное сообщение целевой аудитории, вызываются нужные эмоции. Для этого важно учитывать символическое значение выбираемых цветов, а также уметь их гармонично сочетать.

В качестве примера создания айдентики бренда был выбран Завод-изготовитель кабельно-проводниковой продукции «Донпромкабель».

Вначале, было изучено поле деятельности предприятия – производство и продажа кабельно-проводниковой продукции общепромышленной и шахтной групп. Далее следовал поиск идей, образов и ассоциаций для создания айдентики предприятия. Таким символом стало стилизованное изображение бухты кабеля.

Следующим этапом стал подбор корпоративных цветов. В процессе анализа всего спектра деятельности завода-изготовителя были выбраны два основных цвета – черный и красный. Выбор каждого цвета получил свое логическое обоснование с точки зрения цветового восприятия: черный цвет был выбран как олицетворение мощности и уверенности предприятия, а красный символизирует энергию, а также высокий темп, с которым ток проходит через кабель.

Найденный образ в сочетании с корпоративными цветами нашли свое отображение в логотипе предприятия.

После создания логотипа был разработан фирменный стиль, который впоследствии был отражен на каждом носителе айдентики бренда, а именно официальных бланках, личных и корпоративных визитка, конвертах, кабельных бирках, сувенирной продукции и т.д.



Рисунок 1 – Носители адентики бренда

Выводы. В данном исследовании было рассмотрено общее понятие айдентики бренда, ее основные составляющие и задачи. Основное внимание было уделено подбору цвета при разработке айдентики, так как цвет несет важную эмоциональную и смысловую нагрузку в процессе продвижения бренда на рынке. Также был предствлен пример создания айдентики производственного предприятия «Донпромкабель». Обоснован выбор цветовой гаммы логотипа.

Список литературы

5. Википедия свободная энциклопедия / Интернет-ресурс. – Режим доступа: [www/ URL: https://ru.wikipedia.org](http://www.wikipedia.org).
6. Психология цветовосприятия в брендинге и маркетинге / Интернет-ресурс. – Режим доступа: <http://5sfer.com/9405-psixologiya-cvetovospriatiya-v-brendinge-i-marketinge.html>
7. Что такое айдентика? / Интернет-ресурс. – Режим доступа: <http://say-hi.me/design/chto-takoe-ajdentika-10-osnovnyx-pravil-po-sozdaniyu-firmennogo-stilya.html>

УДК 7.013

ИСПОЛЬЗОВАНИЕ ТРЕНДОВ WEB-ДИЗАЙНА 2017 ГОДА ПРИ РАЗРАБОТКЕ ПЕРСОНАЛЬНОГО САЙТА

Е. Ю. Жиленкова, В.В. Харитонова

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна

zhilenkova18@gmail.com

Аннотация

Жиленкова Е.Ю., Харитонова В.В. Использование трендов web-дизайна 2017 года при разработке персонального сайта. Выполнен анализ и сравнение существующих трендов веб-дизайна. Определены оптимальные приемы для создания персонального сайта.

Ключевые слова: тренд, нестандартная сетка, шрифт.

Актуальность исследования. Информации в сети становится всё больше, в ряд задач и требований для проектирования сайтов добавляются новые пункты. Инструменты веб-дизайна обновляются с каждым годом. Чтобы удержать пользователя, разработчикам и дизайнерам необходимо следовать современным трендам и тестировать разные способы пользовательского взаимодействия.

Цель. Проанализировать применение лучших трендов веб-дизайна, в персональных сайтах.

Основная часть.

Для создания качественно дизайна, который будет хорошо взаимодействовать с пользователем, нужно не только учитывать насколько легко пользователю достичь цели, но и внести в проект самые последние тренды **веб-дизайна**. Рассмотрим два приема, которые активно применяются при создании персональных сайтов.

Нестандартная сетка.

Стандартизированная сетка утомила пользователей и перестала притягивать их внимание. В 2016 году появилось много разнообразных способов расположения элементов на странице сайта. Классическая сетка

используется реже. Дизайнеры продолжают искать способы представления контента пользователям. Один из самых таких способов – отказаться от привычного расположения элементов в виде сетки, продиктованное правилами адаптивного дизайна.

Теперь у страницы нет деления на привычную сетку: фотографии, текст и другие элементы могут располагаться хаотично, перекрывать друг друга. Такой прием позволяет составить необычные композиции и привлечь интерес пользователя. В сумме с качественной анимацией при прокрутке – этот тренд отлично подходит не только для применения при разработке персонального сайта, а и для дизайна интернет-магазина, портала, корпоративного сайта, сайта-визитки.

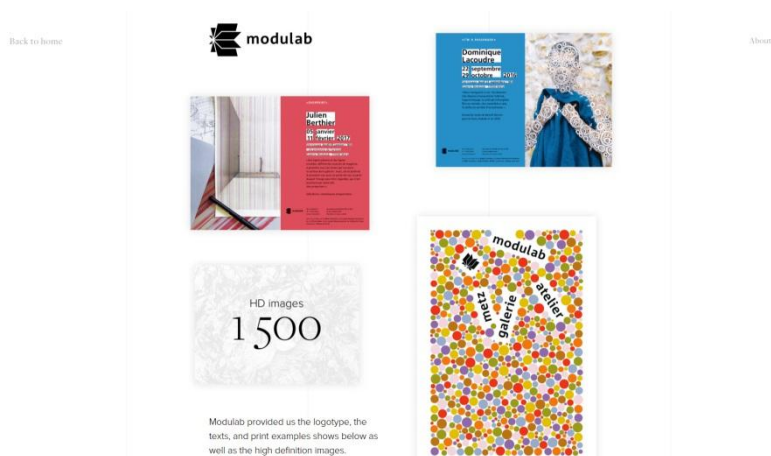


Рисунок 1 – Пример нестандартной сетки

При разработке макетов для персонального сайта был выбран прием нестандартной сетки, т. к. стояла задача разместить большое количество элементов в вольном стиле. Страницы отличаются не только хаотичным расположением фотографий, но и их размером, контрастностью.

Шрифт.

Изменение тенденций можно также проследить и в используемых шрифтах. До недавнего времени использовались простые и качественные гротескные шрифты. Немного более «декоративный» гротеск чаще используется в заголовках, а его более простая версия выбирается для остального текста. Помещать на страницу разные виды шрифтов было

дурным тоном. Теперь в дизайне принято ломать рамки и смело объединять противоположные элементы в одну композицию. В современных разработках можно увидеть использование различных видов шрифтов, сочетание шрифтов с засечками и без них. Тексты анимируются, с помощью них можно выделить главное, разместив в буквах изображения и видео, подчеркнуть шрифтом стилистику информации. Более смелые решения влияют на рост разнообразия веб-типографики.

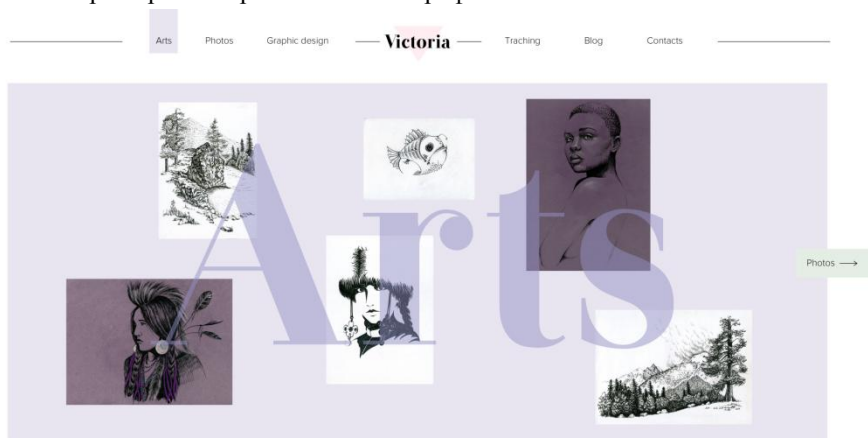


Рисунок 2 – Применение нестандартной сетки на персональном сайте

На руку этому тренду играет и мода на минимализм. Надписи становятся самостоятельными элементами в чистом виде. Стоит также заметить, что шрифт сам по себе – очень сильный графический элемент.

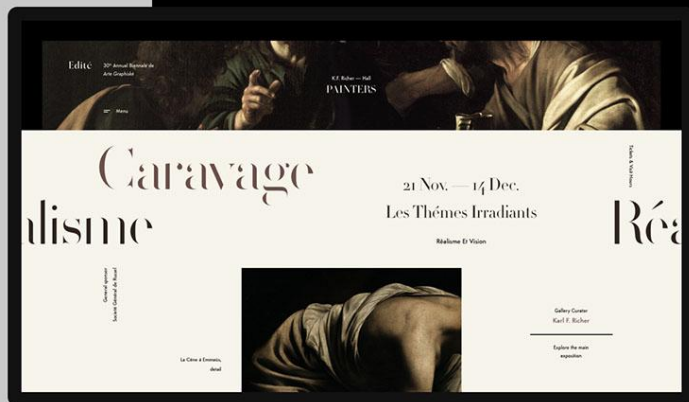


Рисунок 3 – Использование контрастных шрифтов

Типография стала занимать главную часть страницы, вытесняя собой другие элементы. Шрифт, как основной элемент графического дизайна, хорошо смотрится не только на больших экранах, но и на дисплеях смартфонов и планшетов. На данный момент, на большинстве современных сайтах присутствуют большие кричащие заголовки. Главная задача этого тренда – обратить внимание на услугу или продукт. Подобный заголовок должен быть краток и понятен для любого пользователя. При разработке стояла задача – выделить шесть основных пунктов меню. Был выбран прием использование декоративного шрифта для акцентирования на содержимом страницы, а также комбинирование таких шрифтов как: геометрического гротеска Proxima Nova и антиквы нового стиля JWH.

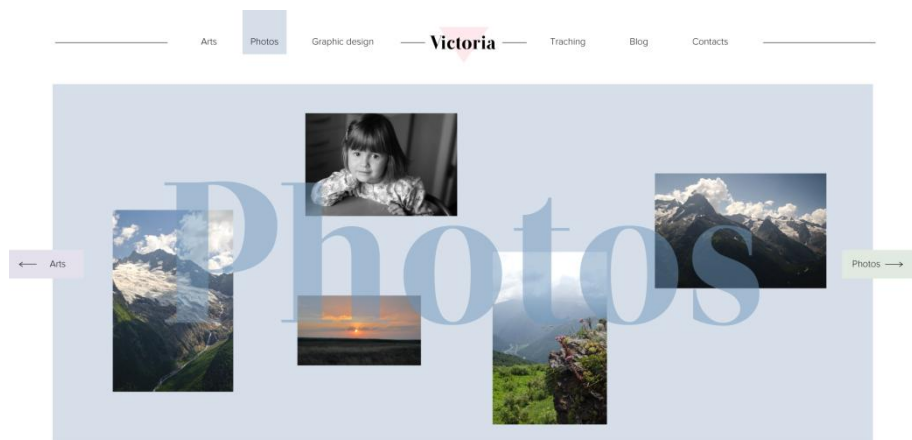


Рисунок 4 – Использование разных шрифтов в оформлении сайта

Выводы. В ходе исследования были проанализированы основные тренды web-дизайна, такие как: нестандартная сетка и шрифт. Применение новых трендов в веб-дизайне удерживает внимание пользователя и качественно доносит информацию.

Список литературы

6. Теплица социальных технологий [электронный ресурс] URL: <https://te-st.ru/>
7. Awwwards [электронный ресурс] URL: <https://www.awwwards.com/>
8. Spark [электронный ресурс] URL: <https://spark.ru/>

УДК 378.14

СОСТАВЛЯЮЩИЕ ДИЗАЙНЕРСКОЙ КОМПЕТЕНТНОСТИ БУДУЩИХ РАЗРАБОТЧИКОВ МЕДИАСИСТЕМ

В.В. Харитонова

Старший преподаватель кафедры компьютерного моделирования и дизайна,
ДонНТУ

victoria.kharytonova@mail.ru

Аннотация

Харитонова В.В. Составляющие дизайнерской компетентности будущих разработчиков медиасистем. В статье рассмотрены составляющие дизайнерской компетентности. Выделены виды деятельности, освоение которых является необходимым условием формирования профессиональной компетентности будущих разработчиков медиасистем.

Ключевые слова: *дизайн, компетентность, медиасистема, медиаиндустрия.*

Актуальность темы.

На данном этапе развития ИТ трудно переоценить то значение, которое приобретает профессиональная деятельность в сфере компьютерного дизайна в современном мультимедийном обществе. Особенности разработчика медиасистем таковы, что его личностные характеристики, наряду с профессиональными и творческими качествами, являются факторами, определяющими удобство взаимодействия продуктов разработки с пользователями, а также привлекательность визуальной оболочки приложения. Таким образом, актуализируется значимость формирования дизайнерской компетентности при подготовке разработчиков медиасистем в техническом университете.

Цель исследования.

Выявить основные составляющие дизайнерской компетентности. Рассмотреть аспекты формирования дизайнерской компетентности а процессе профессиональной подготовки будущих разработчиков медиасистем.

Основная часть.

Традиционно программистов готовят в технических вузах, а дизайнеров в художественных. Анализ требований работодателей, предъявляемых к разработчику медиасистем, показывает спрос на разработчиков с навыками дизайн-проектирования.

Рассмотрим основные составляющие дизайнерской компетентности.

Прежде всего, это: художественно-творческая деятельность, проектная, информационно-коммуникативная, производственно-технологическая. Эти виды деятельности включают в себя профессионально-функциональные знания и умения, которые рассматриваются как владение методами творческого процесса в профессиональной деятельности [1].

Художественно-творческая деятельность. Будущий разработчик медиасистем должен владеть методами творческого процесса проектирования. Творческие способности – способности обобщать опыт творческой деятельности других; способность генерировать идеи в привычной профессиональной деятельности; способность построения ассоциативных связей, аналогий; интуиция, способность прогнозирования предстоящего развития модных тенденций. Необходимыми личностными качествами при этом являются воображение и наглядно-образное мышление. Невозможно развивать и демонстрировать названные способности, не обладая навыками изобразительного искусства и не владея основами классического дизайна.

Проектная деятельность предполагает работу над проектом от идеи до полного воплощения на практике: разработка и выполнение эскизов, дизайн-макетов, объектов дизайна на основе методов дизайн-проектирования, эргономики и антропометрии, основ изобразительного искусства, проектной графики, компьютерных технологий в медиаиндустрии.

Информационно-коммуникативная деятельность. В современных условиях развития профессиональной среды от специалиста требуется не только высокий уровень навыков разработки медиасистем, но и способность ориентироваться в информационном пространстве и средствах коммуникации.

Производственно-технологическая деятельность включает в себя навыки работы в графических редакторах при разработке визуальной оболочки медиасистем. Профессионально востребованным становится тот дизайнер, который действует креативно в ситуации невозможности выбора, т.е. создает оригинальный дизайн-продукт с высокими эстетическими и эргономическими качествами на имеющемся оборудовании, с применением имеющихся материалов и средств [2].

Кафедра компьютерного моделирования и дизайна проводит набор и обучение студентов по профилю подготовки «Информационные технологии в медиаиндустрии и дизайне», где наряду с базовыми учебными дисциплинами по направлению «Компьютерные науки» студенты приобретают знания и умения в области дизайна.

Именно в процессе систематизированной подготовки студента происходит формирование дизайнерских компетенций:

- способность анализировать поставленную задачу и находить пути ее решения;
- владение художественными навыками: рисования, композиционного построения, составления гармоничных цветовых и шрифтовых решений;
- проявление оригинальности мышления, умение работать в различных стилях;
- владение программными продуктами, необходимыми для реализации проектов.

Для формирования дизайнерской компетентности у будущих разработчиков медиасистем необходима соответствующая организация учебного процесса, включающего в себя такие аспекты как:

- получение студентами фундаментальных научно-теоретических знаний в области теории дизайна;
- овладение навыками работы на всех стадиях проектирования от эскиза до готового продукта;
- формирование стремления к творческому самовыражению [3].

Учебно-исследовательская работа студентов основана на следующих принципах:

- учета индивидуального направления творческого и профессионального развития личности студента, использование различных видов творческой деятельности;

- сотрудничества студентов и руководителей научно-исследовательской работы; дизайнерской направленности профессионально-ориентированной учебно-исследовательской работы.

За время подготовки студенты кафедры компьютерного моделирования и дизайна изучают дисциплины, направленные на формирование дизайнерской компетентности: «Теоретические основы дизайна», «Основы композиции в дизайне», «Проектирование в графическом дизайне», «Полиграфические компьютерные системы», «Коммуникативный дизайн».

Выводы. В современных социально-экономических условиях развития общества потребность в высокопрофессиональных ИТ специалистах затрагивает систему их подготовки. Качество профессиональной деятельности специалиста определяется его компетентностью. Подготовка разработчиков медиасистем имеет свою специфику. С одной стороны, это образование направленно на формирование проектно-технических компетенций. А с другой стороны обязательным требованием работодателей является дизайнерская компетентность разработчиков медиасистем. Таким образом, формирование выделенных автором составляющих дизайнерской компетентности в процессе подготовки разработчиков медиасистем является залогом высокого профессионализма будущих специалистов.

Список литературы

1. Тарасова О.П., Яньшина М.М. формирование профессиональных компетенций будущего дизайнера в учебно-профессиональной деятельности // Вестник ОГУ. – 2014. - №5(166).

2. URL: <http://www.science-education.ru/ru/article/view?id=15136> (дата обращения: 02.05.2017).

3. Королева Л.Ю. Современные требования к профессиональной подготовке будущих дизайнеров // Современные проблемы науки и образования. – 2014. – № 5..

Секция № 8
Компьютерная инженерия



УДК 004.42

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ «НАГРУЗКА ПРЕПОДАВАТЕЛЯ». ОЦЕНКА ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ ORM ENTITY FRAMEWORK

Прищепа В.В., Краснокутский В.А., Меренкова Л.Л.

Донецкий национальный технический университет, г. Донецк

Кафедра компьютерной инженерии

e-mail: prishepa_vladimir_94@mail.ru

Аннотация

Прищепа В.В., Краснокутский В.А., Меренкова Л.Л. Разработка программного модуля «Нагрузка преподавателя». Оценка эффективности использования ORM Entity Framework. Рассмотрены проблемы автоматизации планирования учебного процесса в ДонНТУ, в частности формирования нагрузки преподавателя. Разработана и описана модель хранения данных нагрузки преподавателя. Проведена разработка модуля для АСУ с использованием Entity Framework, приведены преимущества и недостатки его использования.

***Ключевые слова:** учебный процесс, учебная нагрузка, схема данных, нагрузка преподавателя, автоматизация, планирование учебного процесса, быстроедействие, данные.*

Постановка проблемы. Планирование, управление и контроль за учебным процессом являются одними из основных функций любого учебного заведения. Выполнение этих функций подразумевает решение широкого спектра задач в жестко ограниченные сроки с требованием высокой достоверности результатов. Распределение учебной нагрузки среди профессорско-преподавательского состава (ППС) является одной из фундаментальных функций в планировании учебного процесса. Издержки же, вследствие нарушения сроков и неточностей результата планирования, представляют собой привлечение дополнительных человеческих ресурсов для решения возникшей проблемы, что влечет за собой ухудшение качества знаний.

Планирование и формирование учебного процесса в ДонНТУ (см. рис. 1) происходит с некоторой долей автоматизации. В данный момент для создания учебных планов используется программный модуль собственной разработки «Учебные планы», он предоставляет пользователю широкий набор инструментов для создания и редактирования планов, их проверки на корректность с формированием отчета об ошибках. Формирование нагрузки кафедр и преподавателей происходит с использованием табличного процессора

Microsoft Excel, большое количество данных вводится вручную, это влечет за собой появление трудно отыскиваемых ошибок. Расписание занятий составляется диспетчерской службой полностью вручную без каких-либо средств автоматизации.

В целях минимизации ошибок при планировании учебного процесса и уменьшения сроков процесса планирования была поставлена задача автоматизировать формирование нагрузки кафедры и преподавателей, составление расписания.

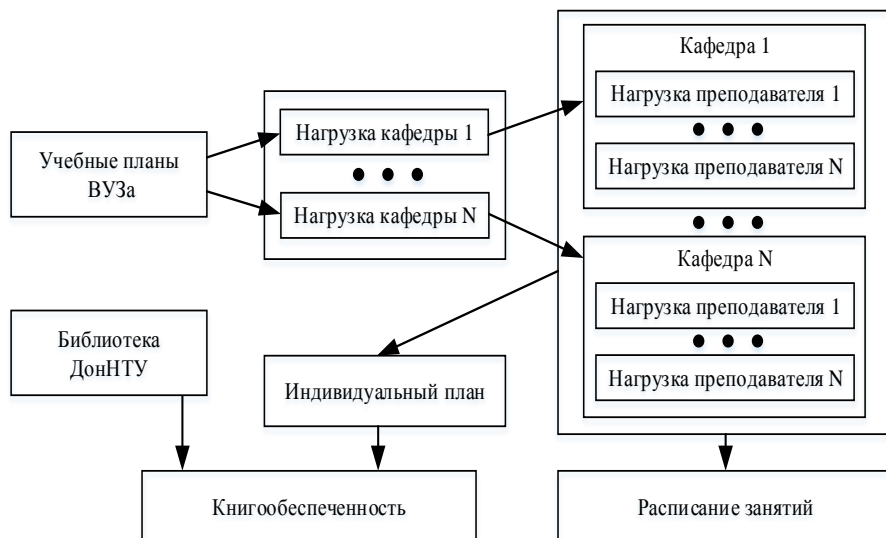


Рисунок 1 – Схема формирования и планирования учебного процесса в ДонНТУ

Пути решения. Описанную проблему можно решить двумя путями:

- 1) приобретение готового программного продукта для решения данных задач;
- 2) разработка собственного программного обеспечения силами отдела разработки ПО.

Наилучшим готовым решением, представленным на рынке программного обеспечения, по мнению авторов, является комплекс от разработчика «Лаборатория ММИС». Его качество подтверждается тем, что он широко используется в учебных заведениях России. Этот программный продукт имеет модульную архитектуру, что позволяет каждому учебному заведению приобрести индивидуальный набор модулей для решения актуальных задач [3].
Плюсы приобретения готового решения:

- получение готового протестированного продукта в кратчайшие сроки.

Минусы приобретения готового решения:

- высокая цена в условиях недостаточного финансирования вуза;
- без разработки специального конвертора данных невозможно интегрировать модуль в действующую АСУ университета, что нивелирует единственную выгоду покупки готового решения.

Учитывая все вышесказанное, было принято решение о разработке собственного программного инструмента для планирования учебного процесса.

Разработка схемы данных нагрузки преподавателя. Проектирование базы данных является фундаментальным этапом в разработке программного обеспечения. От решений, принятых на данном этапе, зависит дальнейшее развитие и успех проекта. В этот момент закладывается потенциал для дальнейшей модернизации, закладываются особенности, которые повлияют на скорость и удобство разработки, сбор статистики.

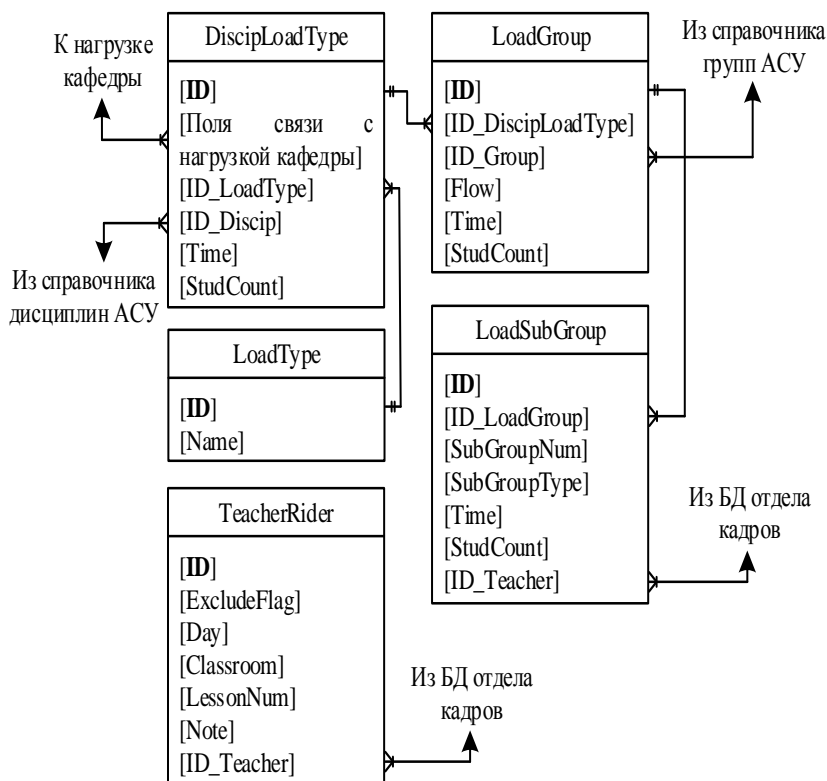


Рисунок 2 – Схема данных нагрузки преподавателя

На рисунке 2 представлена разработанная схема для модуля нагрузки преподавателя. Схема состоит из следующих таблиц:

- DiscipLoadType – главная таблица в части нагрузки преподавателя, содержит в себе все виды учебной нагрузки для каждой дисциплины, которая читается на кафедре, часы выделенные на каждый вид нагрузки и количество студентов;
- LoadType – справочник видов нагрузки (лекции, лабораторные работы, НИРС и т.д.);
- LoadGrup – с помощью связи с ACU извлекается вся информация о группе. В таблице содержатся номер потока, время, выделенное на группу, количество студентов в группе;
- LoadSubGrup – содержит информацию о делении группы на подгруппы, отображается номер подгруппы, количество студентов, время на подгруппу, идентификационный код преподавателя за которым закреплена подгруппа;
- TeacherRider – содержит пожелания преподавателя, данная информация представляет собой исходные данные для начала формирования расписания. Дается возможность исключить какой-либо день недели (поле «Day»), пару (поле «LessonNum») путем указания их порядкового номера, с помощью комбинаций этих полей возможно указание конкретных пар в конкретные дни. Так же возможно указать желаемые аудитории (поле «Classroom») для проведения занятий.

Как известно, очень часто группы объединяются в лекционные потоки, так же не исключено существование лабораторных и практических потоков. Данная сущность специально не была добавлена в схему для уменьшения количества связей в целях уменьшения длины цепочки связанных таблиц при выполнении запросов во время работы модуля формирования расписания. Принадлежность группы к потоку определяется с помощью поля «Flow» в таблице «LoadGroup», тип потока определяется полем «ID_LoadType» в таблице «DiscipLoadType».

Сравнение быстродействия выполнения запросов и скорости разработки с использованием Entity Framework и без него. При разработке с использованием платформы .NET программисту доступны две технологии доступа к данным: ADO.NET и ORM Entity Framework (EF).

ADO.NET (ActiveX Data Object for .NET) – технология доступа к реляционным данным, которая является неотъемлемой частью платформы .NET. Она основана на работе с объектами, называемыми наборами данных (data set), данные в датасет попадают после выполнения SQL-запроса, который хранится в

программе в объекте типа string. SQL-запросы строятся с помощью функционала класса QueryBuilder [1].

ORM EF (Object-relational mapping system – система объектно-реляционного отображения) – предоставляет объектно-ориентированный доступ к реляционным данным средствами языка LINQ (Language Integrated Query – технология от Microsoft, предоставляющая язык запросов, встроенный в язык программирования, с синтаксисом напоминающим SQL). Основная задача фреймворка – преобразовать данные из реляционного вида в объектный для разработки с использованием технологий ООП [2].

Модуль формирования нагрузки преподавателей использует результаты предыдущего шага – формирования нагрузки кафедры. Для исследования был выбран запрос на выборку нагрузки кафедры для одного семестра. Запрос выполнялся с помощью обоих инструментов несколько раз, фиксировалось время выполнения, после чего было вычислено среднее арифметическое время выполнения для каждой технологии и занесено в таблицу 1.

Таблица 1 – Среднее время выполнения запроса

Технология	ADO.NET	ORM EF
Среднее время выполнения, (мс)	13	22

Увеличение времени выполнения запроса при использовании EF вызвано тем, что до выполнения запроса система анализирует данные какого типа были запрошены, формирует SQL-запрос и отправляет его на выполнение. В данном примере БД вернула 130 строк, замедление выполнения при использовании EF составило ~70%. Если же использовать запрос, который будет возвращать большее количество строк, то процент замедления будет уменьшаться, потому что основную часть времени выполнения будет занимать считывание данных с носителя.

При разработке приложения использовался подход Database First, когда БД уже существует. В этом случае EF избавляет программиста от рутинной работы по описанию классов, повторяющих структуру таблиц уже имеющейся базы данных. С помощью специального инструмента построения программной модели данных создаются классы, отображающие структуру таблиц. После выполнения запроса EF возвращает объект типа List<T>, после чего программист сразу может обрабатывать данные в программе, как объекты класса T.

При использовании технологии ADO.NET, программисту необходимо описать соответствующие классы. После выполнения запроса возвращается объект типа DataSet, данные из которого необходимо преобразовать в объекты ранее описанного класса, вся эта рутинная работа требует написания дополнительного кода и затрат времени.

Количество кода, написанного программистом для получения данных из БД в виде объекта класса List<T> при проведении эксперимента, занесено в таблицу 2, подсчитано время необходимое на написание данного кода с учетом средней скорости печати автора ~170 символов в минуту.

Таблица 2 – Количество кода для получения данных из БД

Технология	ADO.NET	ORM EF
Количество кода, (символов)	4688	151
Время написания, (мин.)	27,5	0,9

Указанное количество времени будет затрачено, если писать весь код вручную, но на практике, при использовании возможностей современных сред разработки, оно уменьшается в два-три раза. При использовании EF дополнительно единожды затрачивается не более двух минут на создание программной модели, отражающей структуру базы данных, при использовании специального инструмента «Entity Data Model Wizard».

Выводы

В работе рассмотрены текущие проблемы автоматизации формирования и планирования учебного процесса в ДонНТУ.

Предложены возможные решения проблемы, проведено их сравнение.

Из-за технических сложностей интеграции в действующую АСУ вуза, применить готовые разработки, выполняющие необходимые функции, представляется невозможным.

Кратко описана разработанная реляционная схема хранения данных. Проведено сравнение технологий доступа к данным ADO.NET и ORM Entity Framework с точки зрения оценки быстродействия, скорости и удобства разработки на примере операции извлечения данных из БД.

Учитывая полученные результаты исследования, целесообразно использовать EF при разработке программного комплекса, несмотря на некоторое замедление работы готового продукта, потому что в современной индустрии разработки ПО одной из главных целей является уменьшение длительности разработки.

Список литературы

1. Microsoft Developers Network. ADO.NET // MSDN. [Электронный ресурс] – Режим доступа: [https://msdn.microsoft.com/ru-ru/library/e80y5yhx\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/e80y5yhx(v=vs.110).aspx)

2. Microsoft Developers Network. Introduction to Entity Framework // MSDN. [Электронный ресурс] – Режим доступа: [https://msdn.microsoft.com/en-us/library/aa937723\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx)

3. Лаборатория ММИС. Автоматизированная система «Нагрузка вуза» // MMIS LAB [Электронный ресурс] – Режим доступа: <http://www.mmis.ru/programs/nagruzka>

УДК 004.3

СИСТЕМА КОНТРОЛЯ И УПРАВЛЕНИЯ ДОСТУПОМ

Юрьев Н.Н., Бельков С.Д., Суббота Н.С., Васяева Т.А.

Донецкий национальный технический университет

кафедра автоматизированных систем управления

E-mail: kolyayurev@mail.ru, stepok1995@gmail.com, subbota.nikita@mail.ru,

vasyaeva@gmail.com

Аннотация

Юрьев Н.Н., Бельков С.Д., Суббота Н.С., Васяева Т.А. Система контроля и управления доступом. Рассмотрена классификация компонентов систем контроля и управления доступом. Определены свойства и структура разрабатываемой системы.

Ключевые слова: Системой контроля и управления доступом, контроллеры.

Анализ систем КУД. Системой контроля и управления доступом (СКУД) называется совокупность программно-технических средств и организационно-методических мероприятий, с помощью которых решается задача контроля и управления посещением отдельных помещений, а также оперативный контроль перемещения персонала и времени его нахождения на территории объекта. Система в общем случае состоит из следующих элементов, которые можно разделить на категории:

Технические средства:

- стационарное оборудование, такое как сервер, станция оператора и т.д.;
- управляющее устройство - контроллер. Пожалуй, один из самых главных устройств системы. Непосредственно именно это устройство принимает решение о пропуске или запрете доступа на охраняемый объект, путем непосредственного управления запирающими устройствами. В энергонезависимой памяти хранит базу идентификаторов доступа, права доступа, события предоставления или отказа в доступе;
- запирающие устройства. К ним относятся турникеты, дверные замки, защелки, шлюзовые кабины, шлагбаумы, сплошные двери;
- считывающее устройство, предназначенное для считывания данных с идентификаторов доступа и отправляющие их на контроллер;
- идентификаторы доступа, это магнитные, штрих-кодовые карты, радиобрелки, биометрические данные человека (рисунок сетчатка глаза, геометрия ладони, отпечаток пальца), проксимити карты и т.д.;
- блоки питания, обеспечивают питание элементам системы;
- устройства извещающие о тревожных ситуациях, таких как взлом двери, несанкционированный проход и т.д.;

Программное обеспечение:

- администрирование системы, возможность удаленного управления доступом;
- мониторинг в реальном времени;
- учет рабочего времени;

По степени сложности построения системы, их можно разделить на следующие виды:

1. Автономная СКУД. Предназначены для управления одним или несколькими преграждающими устройствами без передачи информации на центральный пульт и без контроля со стороны оператора[1]. Обычно это простейшие СКУД состоящие из автономного контроллера, считывателя и запирающего устройства, для открытия последнего изнутри помещения, обычно используется либо кнопка, либо датчик-коммутатор движения.

Если система должна управлять доступом не в одно-два помещения, а в несколько десятков, то она строится по сетевому принципу.

2. Сетевая СКУД. Сетевые системы предназначены для обеспечения контроля и управления доступом на крупных объектах (банки, учреждения, предприятия и т.п.) для управления несколькими пунктами прохода (проходные, офисные помещения). Взаимодействует с пропускными конструкциями, осуществляет обмен информацией с центральным пультом. Оператор может оперативно управлять системными устройствами – дистанционно заблокировать замки или открыть их (например, в случае пожара).

Сетевые системы контроля и управления доступом в отличие от автономных, могут обеспечивать ряд дополнительных возможностей:

- сбор и обработку информации о перемещении лиц по объекту;
- организацию и учет рабочего времени;
- управление режимами работы и автоматикой автостоянок.

Сетевая СКУД представляет собой совокупность объединённых автономных систем. В ней все контроллеры управления имеют специальные контакты для подключения внешнего интерфейса и объединены в одну локальную сеть.

3. Интегрированная СКУД, может быть частью общей системы охранной и пожарной безопасности, которая объединяет в себе комплекс технических и программных средств по обеспечению безопасности предприятия и его жизнедеятельности.

Реализация СКУД на малом предприятии. Разобравшись в видах и составляющих систем контроля и управления доступом, приступим к построению собственной сетевой СКУД на малом предприятии. Вначале необходимо определить задачи и функции будущей системы. Задачами, стоящими перед системой, являются обеспечение санкционированного доступа

на территорию предприятия и контроль перемещения сотрудников в помещения имеющие высокий приоритет доступа.

Функции которые обязана выполнять данная система:

- идентификация сотрудника;
- управление доступом;
- регистрация новых пользователей;
- формирование отчетов;

Для осуществления данного проекта, определим структурные элементы, используемые при построении системы:

- сетевые контроллеры «Z-R5 net 8000 IronLogic»;
- считыватели бесконтактных проксимити карт «Matrix II (EM-Marin 125KHz)»;
- турникет «FORMA Бизант 5.3»;
- дверные электромагнитные замки «УМ-180»;
- кнопки открытия двери;
- датчики открытия двери «Страж П-408», доводчики дверей;
- блоки питания «ИБПС-12-1»;
- конвертер интерфейсов USB/RS-485, «Z-397 Guard IronLogic»;
- компьютер с программным обеспечением и базой данных;
- бесконтактные карты «EM-Marin ISO 125 kHz»;

Утрированная структурная схема системы собранная из перечисленных выше компонентов представлена на рисунке 1.

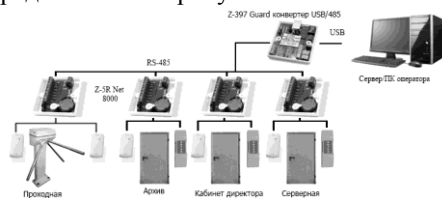


Рис. 1. Структурная схема

Контроллеры объединяются в сеть с использованием интерфейса RS-485 и работают под управлением компьютера. Специализированное ПО, в частности таких компаний как Guard Ligh и СтражЪ, позволяет программировать контроллеры, управлять их работой, скачивать события, регистрировать пользователей в системе, производить удалённое управление открытием дверей и турникета. Каждый контроллер работает автономно, принимая решения о предоставлении доступа по карточке/ключу независимо от того, подключен он к ПК или нет. Контроллеры соединятся между собой последовательно друг за другом. Номер контроллера в сети выставляется специализированным ПО. Изначально все контроллеры имеют сетевой адрес "1". При установке необходимо назначить новые сетевые адреса контроллерам. Максимум в сети может быть 255 контроллеров такого типа.

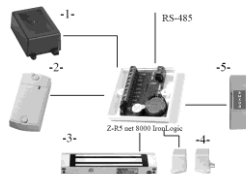


Рис.2. Схема подключения элементов к контроллеру Z-5R net 8000

Рассмотрим подробно схему подключения оборудования к контроллеру, изображенную на рисунке 2. Под номером 1 указан типовой источник питания «ИБПС-12-1», со входом 220В и выходом 12В, в герметичном исполнении. Под номером 2 изображен считыватель «Matrix II».

Под номером 3 указан электромагнитный замок «УМ-180». Подключение контроллера «Z-5R net» возможно также и к электромеханическим замкам. Под номером 4 показан датчик открытия двери, контроллер не включит замок, пока доводчик закрывания двери не прижмет ее к косяку. Номером 5 обозначена выходная кнопка, устанавливаемая внутри помещения. Также к контроллеру могут дополнительно быть подключены светодиод и устройство звуковой сигнализации, которые работают на время открытия замка, но в данной реализации системы они не были использованы. Схема реализации проходной имеет схожую структуру, включая в свой состав сам контроллер, турникет, 2 считывателя на вход/выход и блок питания.

Принцип действия системы. При поднесении прокарты к считывающему устройству, тот считывает код, который в дальнейшем поступает на контроллер. Он в свою очередь принимает решение о предоставлении доступа на основании базы кодов ключей хранящейся в его энергонезависимой памяти. В случае разрешения прохода, контроллер подает сигналы на открытие замка двери (турникета), включение светодиода и зуммера если таковые имеются, которые будут работать пока дверь будет открыта. Контроллер выполняет протоколирование событий, максимальное количество запоминаемых событий 8192. События скачивают из памяти контроллера при помощи специализированного ПО для создания отчетности. Как было указано выше, контроллер не включит замок до тех пор, пока не поступит сигнал о закрытии двери. Открытие двери из помещения осуществляется по нажатию выходной кнопки установленной внутри помещения.

Выводы. Очевидными плюсами такой системы является её дешевизна входящих в неё компонентов, простотой их конфигурации и совместимостью между собой. Для задач и масштабов спроектированной системы, выбранных компонентов будет вполне достаточно.

Список литературы

1. Ворона, В. А. Система контроля и управления доступов / В. А. Ворона, В. А. Тихонов. – М.: Горячая линия - Телеком, 2010. – 272с.

УДК 531.8

УСТРОЙСТВО ИЗМЕРЕНИЯ СКОРОСТИ ВРАЩЕНИЯ ДИСКА ИСПЫТАТЕЛЬНОГО СТЕНДА

В.А. Краснокутский, Ю.Е. Серпуховитин

ГОУ ВПО «Донецкий национальный технический университет»

кафедра компьютерной инженерии

E-mail: progressiff@gmail.ru

Аннотация

Краснокутский В.А., Серпуховитин Ю.Е. Устройство измерения скорости вращения диска испытательного стенда. Рассмотрены вопросы разработки устройства измерения скорости вращения диска испытательного стенда. Испытательный стенд предназначен для отладки системы измерения скорости вращения летательного аппарата.

Ключевые слова: устройство измерения, скорость вращения, микроконтроллер, испытательный стенд.

Постановка проблемы. Для построения модели и системы управления летательного аппарата, построенного на принципе планирования крылатки, необходимо измерять скорость вращения и азимутальный угол в процессе вращения. В работе [1] рассматриваются вопросы построения устройства для измерения скорости вращения летательного аппарата в режиме авторотации. Определение скорости вращения летательного аппарата основано на измерении положения летательного аппарата относительно магнитного поля земли. Для этого используется магнитометр НМС5883L. С целью сокращения времени на разработку системы измерения и материальных средств отладку и предварительные испытания системы измерения скорости вращения летательного аппарата целесообразно проводить на наземном испытательном стенде. В работе рассматриваются вопросы разработки испытательного стенда для измерения скорости вращения летательного аппарата.

Цель статьи – провести анализ вопросов построения испытательного стенда для измерения скорости вращения летательного аппарата.

Решение поставленной задачи.

Испытательный стенд представляет собой диск с изменяемой скоростью вращения. Скорость вращения диска может быть 33, 45 и 78 об/мин. При отладке системы измерения скорости вращения летательного аппарата необходимо сравнивать измеренную скорость вращения с эталонной скоростью вращения

диска испытательного стенда. Устройство измерения скорости вращения диска испытательного стенда разработано на основе учебно-отладочного стенда EV 8031/AVR. Учебно-отладочный стенда EV 8031/AVR построен на базе микроконтроллера I8051. Структурная схема устройства измерения скорости вращения диска испытательного стенда показана на рисунке 1.

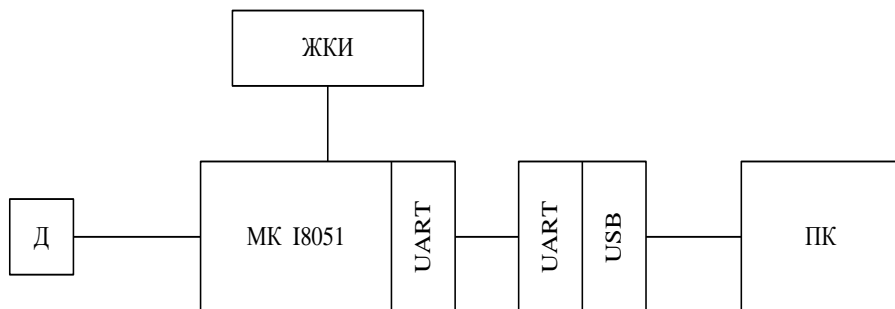


Рисунок 1. Структурная схема устройства измерения скорости вращения диска

Сигналы от датчика Д поступают в микроконтроллер МК I8051. Микроконтроллер измеряет время между двумя срабатываниями датчика и вычисляет текущую скорость вращения диска. Измеренная величина выдается на жидко кристаллический дисплей ЖКИ и передается через преобразователь интерфейсов последовательного порта UART микроконтроллера в USB в персональный компьютер для последующей обработки.

Датчик представляет собой герконовые контакты, которые замыкаются при прохождении рядом магнита, закрепленного на диске испытательного стенда. Датчик подключается к выводу P1.6 микроконтроллера через резистор 1 кОм с помощью нижнего контакта разъема J8. Схема подключения датчика показана на рис. 2 и 3. Схема на рис. 2 простая и не требует дополнительных элементов. Однако необходимо предусмотреть устранение дребезга контактов программным путем. В схеме на рис. 3 дребезг контактов устраняется аппаратно с помощью конденсатора величиной 0,1 мкФ и инвертора на транзисторе. Дребезг контактов можно устранить с помощью RS – триггера с инверсным управлением.

Контакт датчика подключается к порту микроконтроллера через резистор величиной в 1 кОм. При замыкании контакта на землю выход порта ведет себя как источник тока величиной 50 мкА. Этот ток протекает через резистор и создает на нем падение напряжения

$$U = I_{\text{IL}} \cdot R = 50 \text{ мкА} \cdot 1 \text{ кОм} = 50 \text{ мВ}.$$

Допустимое максимальное напряжение логического нуля на входе микроконтроллера равно 0,9 В. Таким образом, падение напряжения на входном резисторе мало и можно пренебречь при реализации схем на рис. 2 и 3.

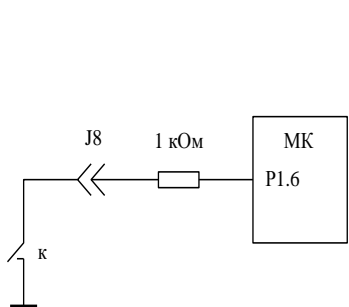


Рисунок 2. Схема подключения датчика

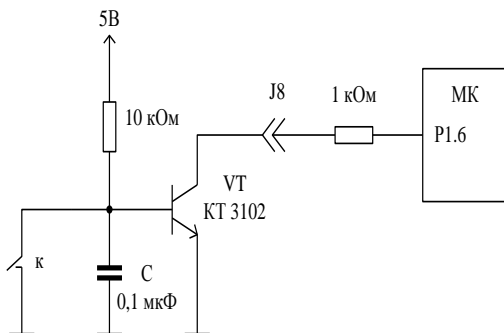


Рисунок 3. Схема подключения датчика с устранением дребезга контактов

Измерение скорости вращения диска основано на измерении времени одного оборота. Время одного оборота – это время между двумя маркерами, которые формирует датчик. Время одного оборота измеряется с помощью таймера 0 микроконтроллера. При появлении маркера (начало оборота) запускается таймер, который каждую миллисекунду инкрементирует переменную *time*. После появления очередного маркера переменная *time* содержит время одного оборота в миллисекундах. Сигнал маркера поступает на вход P1.6 через J8 (нижний контакт). Активный уровень маркера 0 В. Таймер работает по прерываниям в режиме 1. Сигнал маркера сканируется программно путем опроса порта P1.6. При поступлении маркера считывается переменная *time*, рассчитывается скорость вращения диска и выводится на ЖКИ значение скорости в об/мин. В программе предусмотрена возможность передачи скорости вращения диска в ПК.

Определение константы пересчета таймера

Частота внешнего кварцевого резонатора микроконтроллера учебно – отладочного стенда EV8031/AVR равна $f_{\text{osc}} = 7,3728 \text{ МГц}$. Таймер инкрементируется в каждом машинном цикле, т. е. через каждые 12 периодов резонатора. Следовательно, частота работы таймера равна $f_{\text{тайм}} = f_{\text{osc}} / 12$, а период равен $T_{\text{тайм}} = 1 / f_{\text{тайм}} = 12 / f_{\text{osc}} = 1,6276 \text{ мкс}$.

Максимальный интервал времени в 16 – разрядном режиме будет равен

$$T_{16 \max} = 65535 \cdot T_{\text{тайм}} = 65535 \cdot 1,6276 \text{ мкс} = 106,66476 \text{ мкс.}$$

Этого времени недостаточно для измерения скорости вращения диска испытательного стенда. Скорость вращения диска в оборотах в минуту определяется по формуле

$$n_{\text{об/мин}} = \frac{60000}{\text{time}}$$

где time время одного оборота в миллисекундах. Соответственно время одного оборота в миллисекундах составит

$$\text{time} = \frac{60000}{n_{\text{об/мин}}}.$$

Так время одного оборота диска при скорости 100 оборотов в минуту будет равно

$$\text{time}_{100} = \frac{60000}{100} = 600 \text{ мкс},$$

а при скорости 33 оборота в минуту 1816 мкс.

Таким образом, таймер микроконтроллера не позволяет организовать измерение времени одного оборота диска испытательного стенда прямым способом для заданных скоростей. Для измерения времени одного оборота диска необходимо организовать счетчик времени программным путем. Текущее время формируется в переменной time путем инкрементирования ее содержимого каждую миллисекунду. Интервал времени в 1 мкс формируется с помощью таймера 0, работающего в режиме 1. Таймер работает по прерыванию, которое наступает по его переполнению. Определим код пересчета 16 – разрядного вычитающего таймера. Для получения переполнения таймера каждые N мкс код пересчета равен

$$T_{16 N \text{ мкс}} = \frac{65535 \cdot N \text{ мкс}}{106,66476 \text{ мкс}}.$$

Код пересчета для интервала 1 мкс равен

$$T_{16 1 \text{ мкс}} = \frac{65535 \cdot 1 \text{ мкс}}{106,66476 \text{ мкс}} = 614,402 \approx 614.$$

Так как счетчик микроконтроллера суммирующий, то код пересчета будет равен

$$T_{1 \text{ мкс}} = 65535_{10} - 614_{10} = 64921_{10} = \text{FD99}.$$

Программное обеспечение устройства.

Программа микроконтроллера написана на языке С и обеспечивает измерение скорости вращения диска испытательного стенда с выводом информации на ЖКИ – дисплей, а также при необходимости передачи скорости вращения по последовательному интерфейсу. Фрагмент программы, выполняющей измерение скорости и вывода на ЖКИ – дисплей приведен на рис. 4.

```
M1:    if (P1_6 == 1) goto M1;
        TR0 = 1;    // запуск таймера 0
        time1 = time;
        time = 0;
        speed = 60000/time1;
        sprintf (buf, "%7.2f", speed);
        string_to_hd(1, 1, buf);
        goto M1;
```

Рисунок 4. Фрагмент программы микроконтроллера

Программа опрашивает вход P1.6, на который поступает сигнал от датчика. Когда состояние входа станет равным 0 (приход маркера), запускается счетчик для формирования времени в переменной time. Затем считывается время в буферную переменную time1. Переменная time обнуляется для подготовки формирования нового интервала времени между двумя маркерами, поступающими от датчика. Рассчитывается скорость одного оборота диска, которая преобразуется в формат для вывода на ЖКИ и передается на дисплей. Далее управление передается на начало цикла, где проверяется состояние входа P1.6 и процесс повторяется. Следует отметить, что при первом измерении выдается не правильный результат. Синхронизация работы наступает после первого измерения.

Выводы

В результате работы были рассмотрены вопросы измерения скорости вращения диска испытательного стенда. Представлен способ решения поставленной задачи. Проведенные эксперименты показали работоспособность разработанного устройства.

Список литературы

1. Елагин И.А., Краснокутский В.А. Исследование магнитометра НМС5883L в системе измерения скорости вращения летательного объекта/ Программная инженерия: методы и технологии разработки информационно - вычислительных систем (ПИИВС-2016): сборник научных трудов I научно-практической конференции (студенческая секция). 16-17 ноября 2016 г. – Донецк, ГОУ ВПО «Донецкий национальный технический университет», 2016. – 325 с.

УДК 004.3

ВОПРОСЫ ОРГАНИЗАЦИИ АППАРАТНОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СТРУЕВЫХ МЕТОДОВ ИССЛЕДОВАНИЯ БЫСТРЫХ ХИМИЧЕСКИХ РЕАКЦИЙ

Плахин С.В., Краснокутский В.А.

Донецкий национальный технический университет
кафедра компьютерной инженерии
e-mail: maroder666@bk.ru

Аннотация

Плахин С.В., Краснокутский В.А. Вопросы организации аппаратного и программного обеспечения струевых методов исследования быстрых химических реакций. Рассмотрены особенности аналого-цифрового преобразования на базе Arduino Nano и возможность реализации обработки полезного сигнала, полученного струевыми методами исследования быстрых реакций. Приведен вариант программы аналого-цифрового преобразования сигнала. Предложена структура устройства, которое целесообразно реализовать на базе Arduino Nano.

***Ключевые слова:** микроконтроллер, метод остановленной струи, оцифровка сигнала, аналого-цифровой преобразователь.*

Общая постановка проблемы

Обычные методы исследования стационарной кинетики не позволяют наблюдать за самыми быстрыми стадиями ферментативного процесса. Их применение обоснованно только в том случае, если время полупревращения для реакции превышает 10 с. Для измерения скоростей реакций со временами полупревращения до 1 с были разработаны несколько методов, в число которых и входят струевые методы исследования [1].

Широкое применение нашел метод остановленной струи, который основан на быстром смешивании двух растворов за интервал времени, составляющий всего 1-2 мс. Струю создают и останавливают посредством плунжера, который приводит в движение два медицинских шприца. Оба раствора поступают в смеситель и далее к кювету, где после остановки струи измеряется поглощение света [1]. Метод остановленной струи позволяет изучать быстрые жидкофазные реакции с временами превращения до 10^{-3} сек, например, по изменению оптической плотности в ультрафиолетовой или в видимой области спектра. Главной особенностью является то, что метод требует быстрой регистрации. Для измерения поглощения света используют специальный фотоумножитель, подсоединенный к осциллографу. Изменения поглощения, происходящие за доли секунды, регистрируют на экране осциллографа и полученный «след» кривой фотографируют.

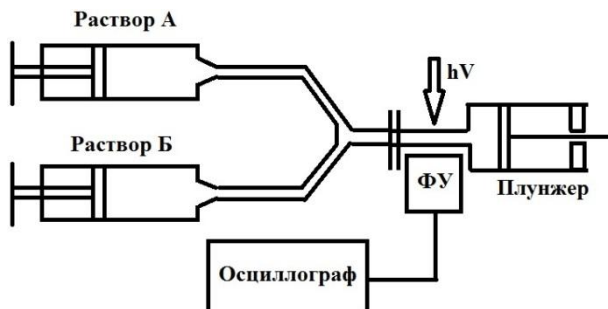


Рисунок 1 – Схема установки, использующей метод остановленной струи.

К сожалению, дальнейшая обработка полученного результата затруднена, так как система не содержит компонента, отвечающего за оцифровку получаемых данных оптической плотности вещества. Использование принтера для графической визуализации малоэффективно, так как появляется необходимость выполнять расчеты вручную, которые отбирают достаточно много времени. Современный рынок предлагает установки, которые позволяют выполнять оцифровку данных с возможностью обработки, однако их стоимость составляет десятки тысяч долларов.

Появляется необходимость в разработке недорогого и эффективного устройства, обладающего достаточным быстродействием и возможностью синхронизации с персональным компьютером для дальнейшей визуализации и обработки данных. Устройство должно выполнять оцифровку сигнала амплитудой до 1 В и иметь достаточную точность и быстродействие, позволяющее регистрировать аperiодичное изменение сигнала длительностью от 10 мс до 30 мс. Для этой цели целесообразно использовать платы линейки Arduino, в частности модель Arduino Nano.

Пути решения проблемы

Для выполнения аналого-цифрового преобразования можно использовать АЦП на базе микроконтроллеров Arduino. В линейке представлены несколько моделей, которые удовлетворяют требованиям поставленной задачи. Одним из них является Arduino Nano на микроконтроллере Atmega328.

Плата отличается компактными размерами, все контакты выведены на две линейки с шагом 2,54 мм, расстояние между линейками 15 мм. Встроенный bootloader и преобразователь USB ↔ COM на базе микросхемы CH340 позволяет обновлять прошивку без использования программатора единственным нажатием кнопки на компьютере. Преимуществом является создание виртуального COM порта на компьютере при подсоединении через USB, что исключает использование стороннего преобразователя.

Микроконтроллер Atmega328 имеет в своем распоряжении 32 КБайт Flash памяти программ с возможностью самопрограммирования, а также 2 КБайта ОЗУ и 1 Кбайт постоянной памяти EEPROM. Максимальная частота ядра доходит до 20 МГц. В состав микроконтроллера входят два 8-битных и один 16-битный таймер [2].

Аналого-цифровой преобразователь Atmega328

Для решения поставленной задачи в микроконтроллере используется 8-канальный 10-разрядный АЦП. Плата Arduino включает в себя 6 аналоговых входов. Напряжение, поданное на аналоговый вход, лежит в пределах от 0 до 5 вольт и будет преобразовано в значение от 0 до 1023, это 1024 шага с разрешением 0.0049 вольт [3]. Дополнительным преимуществом является установка опорного напряжения 1,1 вольта, что позволяет повысить точность аналого-цифрового преобразователя. Также можно использовать внешний ИОН. Недостатком можно назвать достаточно долгую работу стандартной функции библиотеки Arduino. Время преобразования в среднем занимает до 100 мкс.

Пример программы аналого-цифрового преобразования

Среда разработки Arduino IDE состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста (консоли), панели инструментов с кнопками часто используемых команд и нескольких меню. Для загрузки программ и связи среда разработки подключается к аппаратной части Arduino. Помимо стандартных библиотек пользователь имеет возможность подключать сторонние и создавать свои собственные. Дополнительным преимуществом является наличие встроенной программы мониторинга последовательной шины (Serial Monitor). Она отображает данные, посылаемые через последовательный порт.

Оболочка Arduino IDE позволяет выполнять программирование АЦП как через регистры и команды ассемблера, так и при помощи функций стандартных библиотек. Загрузка программы в плату Arduino осуществляется при помощи специального загрузчика - небольшой программы, прошитой в памяти микроконтроллера. Она позволяет загружать в него код без внешних аппаратных средств. Загрузчик активизируется на несколько секунд после сброса устройства, после чего он запускает на выполнение последнюю загруженную в контроллер программу. При запуске загрузчика будет мигать встроенный светодиод, подключенный к 13 ножке контроллера.

Микроконтроллер оснащён тремя таймерами-счётчиками с номерами 0, 1 и 2. Таймеры 0 и 2 имеют разрешение в 8 бит, то есть их счётный диапазон от 0 до 255. Таймер 1 - 16-битный, его счётный диапазон от 0 до 65535 [4]. Для последовательной выборки с определенным шагом используется Таймер 1 микроконтроллера. Он генерирует прерывание через определенный период. Этот параметр измеряется в микросекундах и указывается в специальной функции `Timer1.initialize`. После этого происходит запуск таймера и каждый раз, когда он

переполняется, осуществляется прерывание. Непосредственно в обработчике прерывания происходит считывание аналогового сигнала с входа и его преобразование посредством стандартной функции `analogRead`. В качестве порта указывается номер входа на плате Arduino. После этого выполняется запись считанного значения в память и контроль количества выполненных операций.

После записи данных в память по готовности выполняется побайтная передача на персональный компьютер через последовательный порт. Предварительно полученное значение `int` преобразовывается в два байтовых значения типа `char`. Функция `Serial.flush` выполняет проверку окончания передачи байта данных функцией `Serial.Print`.

Фрагмент листинга тестовой программы:

```
//.....//
pinMode(A3, INPUT);           // инициализируем пин как вход
Serial.begin(9600);           // инициализируем порт, скорость 9600
//.....//
Timer1.initialize(clk_int);    // установка переполнения таймера
Timer1.start();                // старт таймера 1
Timer1.attachInterrupt(Timer1_action); // привязка функции прерывания
//.....//
// Обработчик прерывания таймера 1
void Timer1_action()
{
inputCod= analogRead(A3);      // чтение напряжения на входе A3
DATA[j]= inputCod;            // запись данных в память
j++;
if (j==n) {                    // проверка количества считанных данных
Timer1.stop();                 // остановка таймера
flag=1;                         // флаг готовности данных
}
}
//.....//
for (int i=0; i<n; i++)
{ char LOWBYTE = (byte)(DATA[i] & 0xff); // младший байт
char HIGHBYTE = (byte)(DATA[i] >> 8); // старший байт = 0
Serial.print(LOWBYTE);
Serial.flush();
Serial.print(HIGHBYTE);
Serial.flush();
}
```

Структурная схема разработанного аппаратного блока

Структурная схема показана на рис.2. С установки на плату Arduino идут управляющий и информационный сигналы. По срабатыванию управляющего сигнала происходит считывание данных с аналогового входа. Плата Arduino подключается к персональному компьютеру, в котором осуществляется основная обработка сигнала, формирование и отображение графика зависимости оптической плотности от времени. Существует возможность передачи сигнала в систему моделирования MATLAB.

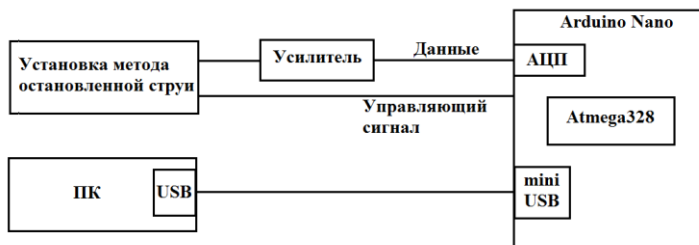


Рисунок 2 – Структурная схема разработанного аппаратного блока

В среде MATLAB был разработан пользовательский интерфейс, который позволяет задавать начальные параметры. Пользователь имеет возможность задать шаг измерения в микросекундах и время длительности процесса, также измеряемое в микросекундах.

Выводы

Была рассмотрена организация аппаратного и программного обеспечения струевых методов исследования быстрых химических реакций.

Предложена структура разрабатываемого устройства и разработан пользовательский интерфейс, позволяющий взаимодействовать с системой. Проведенные исследования позволяют утверждать, что Arduino Nano с микроконтроллером Atmega328 удовлетворяют требованиям и могут использоваться для дальнейшей реализации поставленной задачи.

Список литературы

1. Мецлер Д. Биохимия. Химические реакции в живой клетке. Том 1 / Мецлер Д. - М.: Мир, 1980. - 609 с.
2. Arduino Nano v3.0 - [Электронный ресурс].- Режим доступа: http://radiodetalki.narod.ru/pribory/arduino_nano_v3.pdf
3. ATmega328P- [Электронный ресурс].- Режим доступа: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
4. Таймеры-счётчики Arduino - [Электронный ресурс].- Режим доступа: <https://sites.google.com/site/vanyambauselinux/arduino/tajmery-sceetcki-arduino>

УДК 004.42

ИССЛЕДОВАНИЕ И РАЗРАБОТКА ОПТИМАЛЬНЫХ ЗАПРОСОВ В РАМКАХ РЕШЕНИЯ ЗАДАЧИ ПО АВТОМАТИЗАЦИИ ПРОЦЕССА ФОРМИРОВАНИЯ НАГРУЗКИ КАФЕДРЫ

В.Е. Скиба, В.А. Краснокутский, Л.Л. Меренкова

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
skiba1993victor@gmail.com

Аннотация

Скиба В.Е., Краснокутский В.А., Меренкова Л.Л. Исследование и разработка оптимальных запросов в рамках решения задачи по автоматизации процесса формирования нагрузки кафедры. Исследован процесс формирования нагрузки кафедры. Разработан план по его автоматизации, построены и оптимизированы необходимые запросы для решения данной задачи. Разработана компьютерная программа выполняющая алгоритм формирования нагрузки.

***Ключевые слова:** автоматизированная система управления, нагрузка кафедры, SQL запрос, оптимизация, индекс, оптимизатор, план выполнения, компьютерная программа.*

Постановка проблемы. Процесс формирования нагрузки является фундаментом, на основе которого строится весь учебный процесс в ВУЗе. Автоматизация данного процесса является актуальной и может способствовать повышению эффективности работы кафедр и уменьшить количество используемых человеческих ресурсов. Для решения этой задачи необходимо выполнить следующие действия:

- исследовать алгоритм формирования нагрузки на кафедру в настоящее время;
- на основе проведенного анализа составить план по автоматизации процесса формирования нагрузки;
- разработать и оптимизировать необходимые запросы;
- разработать компьютерную программу, выполняющую все необходимые этапы формирования нагрузки;

Цель статьи – провести анализ процесса формирования нагрузки на кафедру в настоящее время, выделить основные его этапы, составить и оптимизировать необходимые запросы, которые будут использованы в разрабатываемом программном продукте.

Анализ процесса формирования нагрузки. Процесс формирования нагрузки кафедры начинается прежде всего с формирования учебных планов каждым факультетом. *Учебный план* — это документ, определяющий состав учебных предметов, последовательность их изучения и общий объем отводимого на это времени. Каждый учебный план содержит перечень дисциплин, читаемых в течении обучения и учебный график, который показывает перечень учебных недель, модульные контроли, сессии, экзамены и практики.

Далее, на основе учебных планов факультета, формируется учебная нагрузка на данный факультет. Нагрузка факультета содержит в себе сводную нагрузку по всем специальностям за определенный семестр. Нагрузка факультета сводится в один файл формата EXCEL и отправляется на центральный сервер ДонНТУ в общедоступную для всех факультетов папку. В данном документе, каждый лист является нагрузкой определенной кафедры, если дисциплины, читаемые этой кафедрой, присутствуют в учебном плане.

Чтобы составить сводную нагрузку по кафедре, необходимо выбрать полную коллекцию документов с нагрузками всех действующих факультетов с центрального сервера и из каждого документа выбрать нагрузку кафедры и свести все в один документ. Далее выполняется сведение групп в лекционные потоки, проводится расчет часов на все виды нагрузки относительно сформированных потоков и определяется общее количество часов и ставок на кафедру. После выполнения всех шагов нагрузка проверяется и утверждается учебным отделом, если она соответствует всем критериям.

По проведенным исследованиям, процесс формирования нагрузки на кафедру можно свести к следующим шагам:

- 1) Выборка данных из сформированных и подписанных учебных планов за определенный семестр;
- 2) Формирование нагрузки факультета;
- 3) Выборка коллекции нагрузок факультетов и формирование нагрузки кафедры с сведением групп в потоки и последующим расчетом часов;
- 4) Формирование сводной нагрузки кафедры за осенний и весенний семестр;
- 5) Утверждение нагрузки учебным отделом.

В действующей АСУ ВУЗа процесс разработки учебных планов уже автоматизирован, и полная коллекция актуальных планов хранится в базе данных. Исходя из этого, можно сделать вывод, что нет необходимости анализировать всю коллекцию нагрузок факультетов, так как всю первоначальную информацию о нагрузке кафедры можно получить из коллекции учебных планов.

План по автоматизации процесса формирования нагрузки можно свести к следующим этапам:

- 1) Выборка необходимых данных из коллекции учебных планов по заданной кафедре;
- 2) Автоматическое формирование лекционных потоков с возможностью ручного редактирования;
- 3) Расчет кафедральной нагрузки согласно требованиям учебного отдела;

Разработка и оптимизация необходимых запросов. На основе проведенного анализа процесса формирования нагрузки на кафедру, можно составить запрос, позволяющий получить полный набор данных, который необходим для дальнейшей разработки автоматизированной системы.

После разработки запроса необходимо провести его анализ, выявить наиболее затратные операции при выполнении и применить различные стратегии оптимизации с целью уменьшения времени выполнения запроса. К наиболее распространенным стратегиям оптимизации можно отнести:

- Лексическая оптимизация запроса, при которой анализируется часть запроса с предикатами выборки в инструкции WHERE и условия объединения таблиц в инструкции FROM [1];
- Использование индексов. *Индекс* – объект базы данных, который служит для увеличения производительности при поиске данных. Если таблица имеет большое количество строк, то поиск по определенному значению полным перебором может занять много времени. Индекс формируется из значений определенных столбцов с указателем на них, по которому потом эти данные могут быть извлечены напрямую без необходимости полного перебора всех записей [2];
- Использование подсказок оптимизатору запросов. Подсказки оптимизации являются частью запроса SELECT, которые указывают оптимизатору, что инструкцию необходимо выполнять определенным, необходимым для пользователя способом [2].

Поскольку при оптимизации запроса нет необходимости рассматривать селективную часть запроса, то ниже будет представлена часть запроса с условиями FROM и WHERE (рис. 1), к которым будут применены различные средства оптимизации с целью повышения производительности запроса и уменьшения потребляемых ресурсов сервера.

Для начала рассмотрим время выполнения неоптимизированного запроса. Чтобы получить точные данные, запрос необходимо выполнить не менее 5 раз, результаты свести в таблицу и определить среднее время выполнения. Результаты выполнения первоначального запроса сведены в табл. 1.

```

from
(
select u.god_nabora, up.n_sem, up.k_discip, up.npp_g, up.npp, up.k_fak, up.k_kaf,
isnull(of_n.name_p_of_n.name_p_u) as kaf_name, isnull(of_n.fak.name, of_n.fak.name_u) as fak_name,
up.kurs, up.lekcii lekc_1, up.laborat labor_1, up.prakt prakt_1, up.samost samost_1, up.ekzamen ekzamen_1,
up.zachet zachet_1, isnull(up.k_proekt, 0) k_proekt_1, isnull(up.k_rab, 0) k_rab_1,
isnull(up.kont_rab, 0) kont_rab_1, isnull(up.not_kont, 0) not_kont_1, up.dzachet dzachet_1,
isnull(discip_donntu.name_rus, discip_donntu.name_u) name_u, isnull(vid_podg.snam, vid_podg.snam_u) name,
isnull(priz_disc.name, priz_disc.name_u) prizn1, u.spec, u.potok, u.vid_podg, up.k_cikl, spc.name name_cikl,
up.vsego, up.vsego_sem, isnull(up.chasov_kp_kr, 0) chasov_kp_kr, priz_disc.npp priz_disc
from uch_plan up, uch_gr u, PERSONAL.dbo.OFFICE_N of_n,
PERSONAL.dbo.OFFICE_N of_n_fak, dbo.discip_donntu, sp_vid_podg vid_podg, sp_prizn_disc priz_disc, sp_cikl_donntu spc
where u.npp_g=up.npp_g and up.k_fak = u.k_fak and u.aktual=1 and isnull(u.nostandard, 0)=@nostandard
and up.k_kaf=of_n.cod_podr and of_n.aktual = 1 and
up.k_fak=of_n_fak.cod_podr and of_n_fak.aktual = 1 and
up.k_discip=discip_donntu.cod_disc and
vid_podg.k_vid_podg=up.vid_podg and
priz_disc.npp=up.prizn_dis
and up.k_cikl=spc.cod_cikl and
(up.k_fak = @k_fak or 0 = @k_fak) and (up.k_kaf = @k_kaf or 0 = @k_kaf)
and ((cast(up.god_nabor as numeric)+(up.n_sem-1)/2)=(@uch_god) and
up.n_sem%2 = @ost and
case when isnull(u.nostandard, 0)=0 and up.n_sem
in ((@uch_god - u.god_nabora)*2 + 1, (@uch_god - u.god_nabora)*2 + 2) then 1
when u.nostandard=1 then
case
when (u.god_nabora-@uch_god)=1 then 1
when up.n_sem in (((@uch_god - u.god_nabora) + 1)*2, ((@uch_god - u.god_nabora) + 1)*2 + 1)
then 1 else 0 end else 0
end = 1) or (cast(up.god_nabor as numeric)+(up.n_sem-1)/2=@uch_god) and
up.n_sem%2 = @ost and priz_disc.npp in (2601, 2804, 2805))))
rasp1
, dbo.get_grupps_for_nagruzka_tests(@uch_god, @sem, @nostandard, @plan_fakt) studs ,
vw_spec spe
where rasp1.prizn_disc not in (1256, 1254) and rasp1.spec=studs.kod_spec
and rasp1.god_nabora=studs.god_nabora and rasp1.potok=studs.n_potok
and rasp1.vid_podg=studs.vid_podg and studs.K_FAK = rasp1.k_fak and spe.cod_sp=studs.kod_spec
order by spe.COD_FO, rasp1.name_u, rasp1.K_FAK, spe.K_NAPRAV, rasp1.vid_podg, rasp1.kurs, studs.NAME GRUP

```

Рисунок 1 – Условная часть запроса до оптимизации

Таблица 1 – Результаты выполнения запроса до оптимизации

Итерация	№1	№2	№3	№4	№5
Время	4456 мс	4372 мс	4391 мс	4593 мс	4437 мс

Среднее время выполнения запроса составило **4450 мс**. Данное значение будет эталонным, с которым будут сравниваться результаты в дальнейших исследованиях.

Первым этапом в оптимизации запроса является применение алгоритмов лексической оптимизации. Из рис. 1 видно, что таблицы в запросе связаны с помощью простого перечисления, и серверу будет необходимо осуществить полный перебор всех таблиц для выполнения запроса. В качестве оптимизации

необходимо для соединения таблиц применить операции слияния с инструкцией JOIN. При наличии данного условия полный перебор таблиц уже не требуется, а будут выбраны только те записи, для которых выполняется условие, указанное в слиянии. После выполнения лексической оптимизации, среднее время выполнения запроса за 5 итераций составило **4289 мс**, что меньше на **160 мс** относительно первоначального запроса и дает прирост производительности всего **3.5%**.

Следующим шагом оптимизации является создание индексов. Индексы необходимо создавать на те поля, по которым связываются таблицы в инструкции FROM или на поля, которые участвуют в условии выборки WHERE. В текущем запросе наибольшая часть данных извлекается из таблиц UCH_PLAN, UCH_GR и STUDENT (во вложенной функции get_grupps_for_nagruzka_tests). Назначим индексы на ключевые поля в данных таблицах и проверим как это повлияет на время выполнения. Результаты выполнения запроса после создания индексов сведены в табл. 2.

Таблица 2 – Результаты выполнения запроса после создания индексов.

Итерация	№1	№2	№3	№4	№5
Время	1278 мс	1176 мс	1134 мс	1139 мс	1171 мс

Среднее время выполнения составило **1179 мс**. По сравнению с первоначальным запросом, время выполнения уменьшилось на **3271 мс**. Благодаря созданию индексов на трех используемых таблицах был получен прирост производительности на **73.5%**.

Последним этапом оптимизации является использование подсказок оптимизатору запросов. Наиболее распространенными являются подсказки в отношении связывания таблиц. Существует несколько подсказок соединения [2]:

- FORCE ORDER – указывает оптимизатору на то, что таблицы необходимо объединять в том порядке, в каком они следуют в предложении FROM запроса SELECT;
- Подсказки LOOP, MERGE и HASH указывают оптимизатору использовать методы вложенного цикла, соединения слиянием и соединения хешированием соответственно.

Проведя эксперименты с каждой подсказкой было выявлено, что наиболее эффективной является подсказка Force Order. Результаты выполнения запроса с ее использованием сведены в таблицу 3.

Таблица 3 – Результаты выполнения запроса с опцией Force Order

Итерация	№1	№2	№3	№4	№5
Время	315 мс	310 мс	316 мс	313 мс	306 мс

Среднее время выполнения запроса составляет **312мс**, что на 4138 мс быстрее чем в первоначальном варианте запроса. Прирост производительности с использованием операции Force Order составляем **93.2%**.

После проведенных исследований в области оптимизации запросов можно сделать вывод, что наиболее эффективными способами оптимизации являются назначение индексов и использование подсказок оптимизатору.

Разработка программы автоматизированного формирования нагрузки.

Проанализировав необходимые требования и критерии был спроектирован следующий набор таблиц для автоматизированной системы:

- NAGRUZKA_KAF_REG – регистрационная таблица, где собрана коллекция записей, идентифицирующих каждую созданную нагрузку;
- NAGRUZKA_BASIC – таблица с первоначальными данными, полученными из учебных планов. Также в данной таблице записывается информация о принадлежности группы к лекционному, практическому и лабораторному потоку, информация о разбиениях групп на подгруппы;
- NAGRUZKA_RESULT – таблица, хранящая данные о рассчитанной нагрузке;
- NAGRUZKA_ITOG – таблица, хранящая сводную информацию по нагрузке;

Структура спроектированных таблиц и связи между ними показаны на рис.

2.

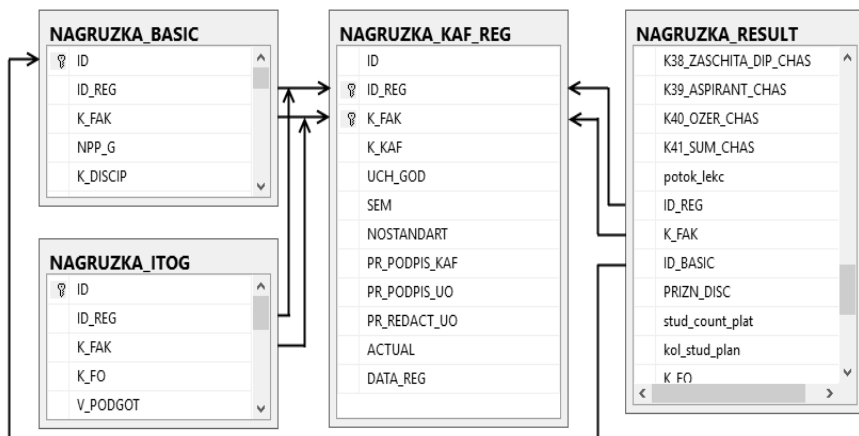


Рисунок 2 – Структура спроектированных таблиц

На основе всех полученных данных и требований был реализован программный продукт, в котором выполняются следующие функции:

- 1) Авторизация и определение программной роли пользователя и доступных ей прав;
- 2) Выборка необходимых дисциплин в целом по ВУЗу по каждой кафедре из сборника учебных планов, хранящихся на Центральном сервере, согласно заданным параметрам (код кафедры, учебный год, семестр, и т.д.);
- 3) Формирование базовой нагрузки;
- 4) Выполнение функций объединения групп в потоки согласно установленных методик в вузе;
- 5) Расчет планируемых часов проведения занятий на кафедру согласно нормативам;
- 6) Редактирование нагрузки кафедрой до подписания УО;
- 7) Редактирование нагрузки УО с отправкой изменений на сервер факультета;
- 8) Реализация цифровой подписи кафедры и подписи учебного отдела с блокировкой к изменениям;
- 9) Удаление и восстановление нагрузки;
- 10) Сохранение результатов работы в базе данных;
- 11) Формирование выходных итогов по нагрузке и сводных отчетов за семестр по кафедре;

На текущий момент, разработанный программный продукт находится на стадии тестирования и внедрения в действующую АСУ ВУЗа.

Выводы. Проведен анализ процесса формирования нагрузки на кафедру в настоящее время. Выявлены основные этапы и сформирован план по автоматизации данного процесса. Разработан и оптимизирован запрос на выборку необходимых данных из коллекции учебных планов. На основе проведенных исследований в области оптимизации запросов было выявлено, что наиболее эффективной стратегией оптимизации является назначение создание индексов и использование подсказок оптимизатору. Спроектирована оптимальная структура необходимых таблиц для автоматизированной системы. На основе требований и указаний учебного отдела был разработан программный продукт, выполняющий все этапы формирования кафедральной нагрузки.

Список литературы

1. Обзор развития методов лексической оптимизации запросов. Труды Института системного программирования РАН Том 23. 2012 г. Стр. 195-214.
2. Душан Петкович, Microsoft SQL Server 2012. Руководство для начинающих: Пер. с англ. – СПб.: БХВ-Петербург, 2013, – 816 с.: ил.

УДК 004.92

**ЭФФЕКТИВНОСТЬ ИСПОЛЬЗОВАНИЯ МНОГОЯДЕРНЫХ СИСТЕМ
ДЛЯ ПРОГНОЗИРОВАНИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ ПОДВИЖНОГО
ОБЪЕКТА РАСПРЕДЕЛЕННОГО СИМУЛЯТОРА ТЯЖЕЛОЙ
ИНЖЕНЕРНОЙ ТЕХНИКИ**

Койбаш А.А., Кравченко А.Г.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
alexandr.koybash@gmail.com

Аннотация

Койбаш А.А., Кравченко А.Г. Эффективность использования многоядерных систем для прогнозирования траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники. Разработаны пути оптимизации алгоритма поиска кратчайшего пути A^ для подвижных объектов на многоядерных системах. Исследована эффективность методов ускорения вычисления маршрута, определены достоинства и недостатки каждого метода. Определено поведение тяжелой техники для следования по заданной траектории с использованием метода мультифиниша.*

Ключевые слова: *прогнозирование поведения объекта, поиск кратчайшего пути, многопоточность, мультифиниш.*

С развитием науки значительно возросла роль средств передвижения. Инженерная техника используется повсеместно, однако на практике маршрут далек от оптимального. Современные компьютерные технологии дают возможность эффективного решения проблемы оптимизации траектории движения. Это позволит получить значительный выигрыш по времени и, как следствие, затратам топлива, а также уменьшит износ техники и позволит смягчить влияние человеческого фактора.

Для решения этой задачи местность передвижения представляется в виде рабочего дискретного поля. Это поле представлено мультиграфом, клетки поля - вершинами, а расстояния между ними – весами рёбер. Таким образом, появляется возможность использовать математические методы просчета кратчайшего маршрута для графов. Так как вес рёбер неотрицательный, использован эвристический подход к алгоритму Дейкстры – алгоритм A^* [1].

Для каждой вершины стоимость пути вычисляется по формуле:

$$F = G + H, \quad (1)$$

где: G – стоимость передвижения к текущей вершине из начальной;

H – эвристическая стоимость передвижения от текущей вершины до конечной.

Следующей вершиной выбирается точка с минимальным параметром F . Таким образом, для каждой новой позиции алгоритм вынужден просчитывать по формуле (1) параметры 8 соседних вершин. Последовательный обход является крайне ресурсоёмким. Длительные вычисления негативно влияют на возможность актуального просчета пути тяжелой техники, так как увеличивается задержка просчета нового маршрута при отклонении от определенной ранее оптимальной траектории. Многопоточная реализация позволит ускорить алгоритм и повысить аппроксимирующую дискретизацию рабочего поля.

Алгоритм реализован средствами языка $C\#$ среды разработки Microsoft Visual Studio, так как платформа .NET позволяет реализацию кроссплатформенных приложений [2]. Одним из вариантов оптимизации является распределение просчета параметров соседних вершин по потокам. Данный способ также является вариативным и есть несколько путей реализации: распределение на 2 потока, в котором первый обрабатывает 1 – 4 вершины, второй вершины 5 – 8; распределение на 4 потока, обрабатывающих по 2 соседа; распределение на 8 потоков, каждый из которых обрабатывает по одной вершине. Все варианты данного метода редактируют общие списки, поэтому доступ к памяти синхронизируется мьютексами [3].

Альтернативным методом является разделение единой траектории на несколько вложенных маршрутов. Каждому вложенному маршруту соответствует свой поток. Синхронизация не требуется, так как каждый поток имеет собственные локальные переменные. Сравнительный анализ двух методов проведен на процессоре Intel Pentium CPU B970 @ 2.30 GHz, 2.30 GHz. Время выполнения отображено на рисунке 1.

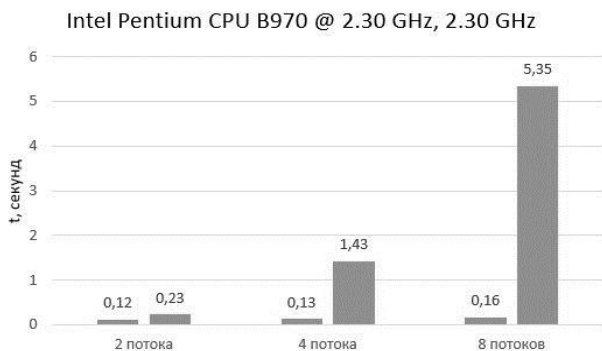


Рисунок 1 – Время выполнения методов

В левом столбце отображено время вычисления одновременно нескольких маршрутов, в правом – распределение просчета вершин единого пути по потокам. Время выполнения однопоточной реализации составляет 0,12 мс. Как видно из графика, метод распараллеливания просчета вершин одного пути показывает существенное падение производительности. Средствами профилировщика Microsoft Visual Studio определены множественные конкуренции потоков за доступ к общей памяти, а также циклическое ожидание потоками завершения друг друга в месте синхронизации, которые представлены на рисунке 2 [1].

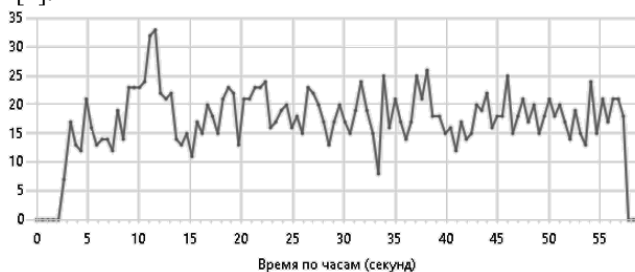


Рисунок 2 – Диаграмма конкуренции потоков

Альтернативный вариант, в котором каждый поток просчитывает свой маршрут, напротив показал существенный прирост во времени выполнения. Это связано с наличием раздельной памяти у каждого потока и отсутствием необходимости синхронизации потоков в цикле. Средства профилировщика показали отсутствие состязаний за исключением синхронизации в главном потоке.

Просчет маршрута необходимо выполнять во время передвижения, чтобы сообщать объекту действия согласно вычисляемой траектории. Это необходимо для оптимального и своевременного достижения конечной цели. Пример маршрута между точками А и В показан на рисунке 3.

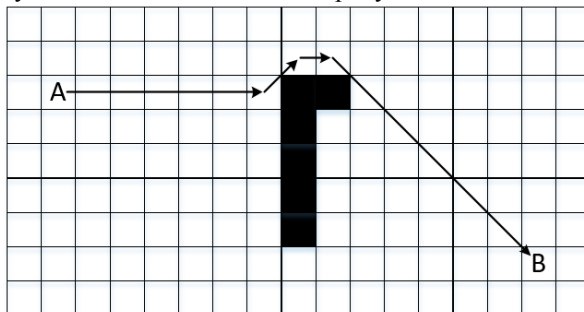


Рисунок 3 – Оптимальный маршрут передвижения

Единица тяжелой инженерной гусеничной техники может менять направление движения, а также регулировать скорость. Необходима возможность поворота на 45° и на 90° , а также ускорение или торможение. Поворот на 45° осуществляется отключением от трансмиссии той гусеницы, в направлении которой необходимо повернуть. Для крутых поворотов на 90° требуется дополнительное торможение гусеницы. Скорость изменяется переключением передачи и с помощью педали [4]. Таким образом, возникает необходимость составить очередность действий для достижения цели. Список действий формируется с использованием метода мультифиниша. При продолжительном однонаправленном отрезке появляется возможность ускорения для улучшения временной составляющей движения. После достижения максимальной скорости давление на педаль не меняется, передвижение остаётся равномерным. Перед поворотами на 45° отключается одна гусеница, для 90° гусеница тормозится. Если дальнейший путь прямолинейный, возобновляется прежняя скорость движения. Перед финишем необходимо торможение.

Таким образом, разработаны и протестированы методы многоядерного ускорения алгоритма A^* . Метод, в котором потоки циклически просчитывают параметры соседних вершин, показал падение производительности. Это связано с конкуренцией за общие участки памяти, а также из-за синхронизации завершения потоков после просчета соседей каждой вершины. Метод распределения траектории на несколько маршрутов показал улучшение производительности из-за разделяемой памяти, однако такая оптимизация имеет недостаток в сходимости краёв маршрутов. Для следования траектории разработан алгоритм определения последовательности действий, в котором обосновано изменение скорости и определены действия для поворотов.

Список литературы

1. Койбаш А.А., Кривошеев С.В. Прогнозирование траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники. В кн.: Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ – 2016): материалы VII междунар. науч.-техн. конф., Донецк, 2016. / редкол. А.Ю. Харитонов и др. Донецк: ДонНТУ, 2016. С. 343-346.
2. Кристиан Нейгел и др. C# 5.0 и платформа .NET 4.5 для профессионалов. – М.: «Диалектика», 2013. – 1440 с.
3. Hughes C., Hughes T. Professional Multicore Programming: Design and Implementation for C++ Developers. – Indianapolis: Wiley Publishing, Inc., 2008. – 648 p.
4. Анилович В. Я., Водолажченко Ю. Т. Конструирование и расчет сельскохозяйственных тракторов. Справочное пособие. Изд. 2 переработанное и доп. – М.: «Машиностроение», 1976. – 456 с.

УДК 004.77, 004.73

ИССЛЕДОВАНИЕ ПЕРСОНАЛЬНЫХ СЕТЕЙ ПРИ ПРОЕКТИРОВАНИИ КОМПЬЮТЕРНЫХ СИСТЕМ

Р.Ю. Габибов, Д.В. Николаенко

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
beanruslan@gmail.com

Аннотация

Габибов Р.Ю., Николаенко Д.В. Исследование персональных сетей при проектировании компьютерных систем. Выполнен анализ области применения технологии беспроводной передачи данных Bluetooth. Определены аспекты практического применения при проектировании КС.

Ключевые слова: компьютерная система, персональная сеть, технология беспроводной передачи данных

Постановка проблемы

В процессе проектирования компьютерных систем разработчики сталкиваются с проблемой приема-передачи различной информации. Таким образом, возникает необходимость исследования и анализа современных технологий для последующего применения в области компьютерной инженерии. Для реализации этого необходимо выполнить следующее:

- проанализировать распространенные технологии беспроводной передачи данных;
- определить правила и требования, помогающие выбрать наиболее подходящий стандарт при проектировании КС;

Цель статьи – провести анализ современной технологии беспроводной передачи информации Bluetooth, и определить преимущества и недостатки исходя их способов последующего использования при проектировании компьютерных систем.

Анализ концепции беспроводной передачи данных

Сейчас можно говорить о том, что все распространенные технологии персональных сетей прошли испытание временем, и каждая из них обладает характерными незаменимыми свойствами. Типичным примером PAN (Personal Area Network) является беспроводное соединение вычислительного устройства с периферийными, такими как принтер, наушники, мышь или клавиатура. В свою очередь мобильные телефоны также используют технологию PAN для

соединения со своей периферией (чаще всего это наушники), а также с компьютером своего владельца. Последние годы некоторые марки наручных часов также стали поддерживать технологию PAN, превращаясь в универсальные устройства с функциями PDA (Personal Digital Assistant).

Разработкой стандартов для беспроводных персональных сетей и координацией их с другими стандартами занимается рабочая группа IEEE по стандартам серии 802.15, которая в свою очередь состоит из следующих групп:

- 802.15.1. Рабочая группа 1, определяет стандарты на беспроводную персональную сеть, основанную на использовании спецификаций Bluetooth версии 1.1. В основе этой технологии с расширением спектра лежит скачкообразное изменение частоты (FHSS), скорость передачи не превышает 1 Мбит/с. Группа 802.15 опубликовала стандарт 802.15.1 в июне 2002 года; на его основе и осуществляется разработка Bluetooth-устройств.

- 802.15.2. Группа, отвечающая за разработку этого стандарта, Рабочая группа 2, дает практические рекомендации, способствующие сосуществованию сетей стандартов 802.15 и 802.11. Проблема состоит в том, что обе сети работают в одном и том же диапазоне 2,4 ГГц, поэтому координация между их работой жизненно необходима. Группа проводит оценку возможных помех и предлагает методы противодействия им.

- 802.15.3. Рабочая группа 3 работает над проектами новых стандартов на высокоскоростные беспроводные персональные сети. Скорость передачи может составлять 11, 22, 33, 44 и 55 Мбит/с. Наряду с этими повышенными скоростями передачи стандарты регламентируют применение механизмов обеспечения качества связи (QoS), что делает их хорошей основой для обеспечения работы мультимедийных приложений. Эта группа занимается также снижением стоимости и потребляемой мощности. Проект стандарта 802.15.3 уже готов для распространения.

- 802.15.4. Рабочая группа 4 работает над стандартами, предусматривающими низкую скорость передачи данных, но зато при экстремально низком энергопотреблении для малогабаритных устройств, рассчитанных на работу без замены батареи питания в течение месяцев и даже лет. Кандидатами на применение этой технологии могут быть всевозможные датчики, "умные" идентификационные жетоны и системы бытовой автоматизации. Скорость передачи данных составляет 20, 40 и 250 Кбит/с. Проект стандарта 802.15.4 уже можно приобрести [1].

Определение правил и требований предъявляемых к персональным сетям

Персональные сети должны обеспечивать как фиксированный доступ, например, в пределах дома, так и мобильный, когда владелец устройств PAN перемещается вместе с ними между помещениями или городами.

Конечно, персональные сети во многом имеют общие свойства и частично схожи на локальные, но тем не менее у них есть и свои особенности:

- Многие из устройств, которые могут входить в персональную сеть, гораздо проще, чем традиционный узел LAN — компьютер. Кроме того, такие устройства обычно имеют небольшие габариты и стоимость. Поэтому стандарты PAN должны учитывать, что их реализация должна приводить к недорогим решениям, потребляющим небольшую энергию.

- Область покрытия PAN меньше области покрытия LAN, узлы PAN часто находятся на расстоянии нескольких метров друг от друга.

- Высокие требования к безопасности. Персональные устройства, путешествуя вместе со своим владельцем, попадают в различное окружение. Иногда они должны взаимодействовать с устройствами других персональных сетей, например, если их владелец встретил на улице своего знакомого и решил переписать из его устройства PDA в свое несколько адресов общих знакомых. В других случаях такое взаимодействие явно нежелательно, так как может привести к утечке конфиденциальной информации. Поэтому протоколы PAN должны обеспечивать разнообразные методы аутентификации устройств и шифрования данных в мобильной обстановке.

- При соединении малогабаритных устройств между собой желание избавиться от кабелей проявляется гораздо сильнее, чем при соединении компьютера с принтером или концентратором. Из-за этого персональные сети в гораздо большей степени, чем локальные, тяготеют к беспроводным решениям.

- Если человек постоянно носит устройство PAN с собой и на себе, то оно не должно причинять вред его здоровью. Поэтому такое устройство должно излучать сигналы небольшой мощности, желательно не более 100 мВт (обычный сотовый телефон излучает сигналы мощностью от 600 мВт до 3 Вт).

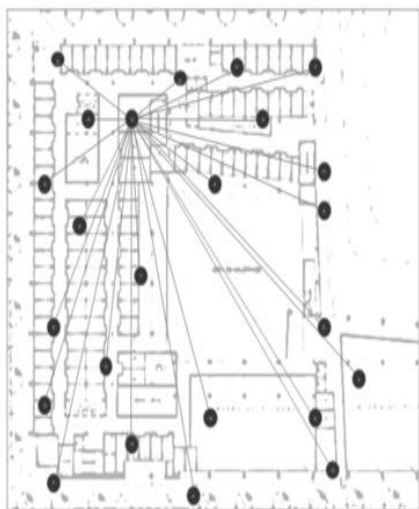
Сегодня самой популярной технологией PAN является Bluetooth, которая обеспечивает взаимодействие 8 устройств в разделяемой среде диапазона 2,4 МГц со скоростью передачи данных до 723 Кбит/с [2].

Также стоит уделить внимание особенностям топологий организации сети, которые используются в современных стандартах беспроводной передачи информации. Так в сетях с применением технологий Bluetooth и Wi-Fi сетевое взаимодействие идет через центральный шлюз, и в случае его выхода из строя обмен данными станет невозможным. Кроме этого отдельные узлы могут остаться без связи, если неожиданно возникнет какая-то преграда на пути следования радиосигнала.

Что же касается сетей ZigBee и Thread, то в данных технологиях надежность связи повышается за счет наличия избыточных связей между устройствами. Все устройства, которые не уходят в спящий режим, выполняют роль роутеров, которые ответственны за маршрутизацию сетевого трафика, выбора оптимального маршрута следования и ретрансляцию пакетов. Даже если

из строя выйдет устройство, которое выступало в качестве организатора сети, ZigBee-сеть продолжит функционировать дальше. Возникновение помехи или преграды, а также выход какого-либо из роутеров из строя не является критичным за счет наличия избыточных связей. Поэтому с введением дополнительных узлов, которые имеют стационарное питание и могут выполнять задачи роутера, сеть становится надежнее. На рисунке 1 наглядно продемонстрировано применения различных топологий персональных сетей в различных технологиях [3].

Wi-Fi, Bluetooth



ZigBee, Thread

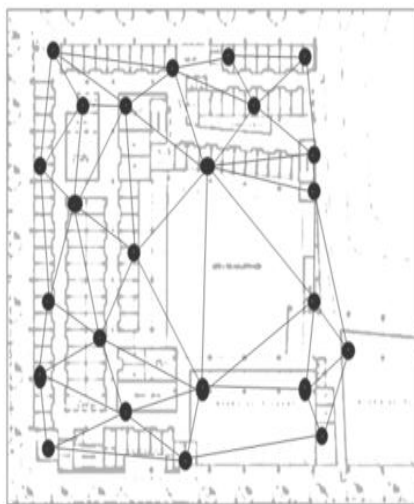


Рисунок 1 – Сетевые топологии WPAN

Достоинства и недостатки современных стандартов персональных сетей

Технология Bluetooth обеспечивает рабочие характеристики, сходные с таковыми беспроводных локальных сетей. За счет использования высокомошной версии Bluetooth производители в дальнейшем смогут предлагать точки доступа и маршрутизаторы Bluetooth с радиусом действия таким же, какой обеспечивают сети стандарта 802.11. Однако предлагаемые в настоящее время изделия Bluetooth имеют намного меньшую потребляемую мощность и ориентированы на выполнение функций, характерных для беспроводных персональных сетей. Кроме того, компьютерным системам, поддерживающим технологию Bluetooth, будет трудно завоевать существенную долю рынка, поскольку системы с использованием стандарта 802.11 уже достаточно широко распространены.

В чем уступает технология Bluetooth изделиям стандарта 802.11, так это в радиусе действия и производительности. Компоненты сетей 802.11 могут обеспечивать скорость передачи до 54 Мбит/с, Bluetooth – в лучшем случае 1 Мбит/с. Этого может оказаться вполне достаточно в большинстве случаев замены кабельных соединений, но для просмотра Web-технологий с использованием широкополосного соединения или создания корпоративной сети нужны более высокие характеристики. Кроме того, радиус действия устройств стандарта 802.11 в условиях ЛВС обычно составляет примерно 90 м (300 футов), что намного превышает возможности Bluetooth. Для организации беспроводной сети на основе технологии Bluetooth на достаточно обширной площади разработчикам КС необходимо будет развернуть слишком много точек доступа.

Выводы

Произведен анализ распространенных стандартов персональных сетей. Определены правила выбора подходящей технологии, в зависимости от предназначения разрабатываемой компьютерной системы. Проведена оценка качества и сравнение современных стандартов PAN.

Результаты показали, что при проектировании КС с высоким уровнем надежности беспроводной передачи информации имеет смысл использовать технологию ZigBee или Thread. Если же более приоритетным фактором является скорость передачи данных, компактность и масштабируемость системы, то стоит прибегнуть к технологиям WiFi или Bluetooth.

В результате, при наложении объектов, данные о полупрозрачности были утеряны.

Список литературы

1. Джим Гейер. Беспроводные сети. Первый шаг : Пер. с англ. – М.: Издательский дом "Вильямс", 2005. – 192 с.: ил.
2. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы, 4-е издание. СПб.: Питер, 2010, – 943 с.
3. Блог компании «ЭФО» [электронный ресурс] // Хабрахабр: [сайт]. [2006-2017]. URL: <https://habrahabr.ru/company/efo/blog/281048/>

УДК 004.4'2

ОСОБЕННОСТИ РАЗРАБОТКИ ПРИЛОЖЕНИЙ ДЛЯ УНИВЕРСАЛЬНОЙ ПЛАТФОРМЫ WINDOWS

А.Ю. Игнатов (4 курс, каф. КИ), С.В. Теплинский, к.т.н., доц.
Донецкий национальный технический университет, г. Донецк
кафедра «Компьютерная инженерия», 3nix@mail.ru

Аннотация

Игнатов А.Ю., Теплинский С.В. Особенности разработки приложений для универсальной платформы Windows. Проанализированы средства для разработки приложений для универсальной платформы Windows. Проведена сравнительная характеристика с предшествующими технологиями.

Ключевые слова: Универсальная платформа Windows, разработка приложений.

Постановка проблемы. С каждым днем появляются новые технологии программирования, существующие – развиваются или устаревают. Вместе с тем, при разработке современных приложений также уделяется внимание совместимости с различными платформами. Эти проблемы коснулись и разработки приложений для операционных систем семейства Windows.

Цель статьи – проанализировать особенности разработки приложений для универсальной платформы Windows.

Универсальная платформа Windows. Универсальная платформа Windows была представлена после появления Windows 10. Для всех платформ Windows 10 было разработано общее ядро Windows, один стек ввода-вывода файла и одна модель приложения. В результате, приложения, написанные с помощью этой технологии, работают на ПК, смартфонах, планшетах или Xbox One, при чем, не указывается, для какого устройства оно разработано.

Кроме того, такие технологии, как Windows Forms и WPF, прекратили свое развитие, хотя и поддерживаются, что выводит Универсальную платформу Windows на первый план.

Требования при разработке приложений для Универсальной платформы Windows. Разработка приложений для Универсальной платформы Windows осуществляется в операционной системе Windows 10 с использованием Visual Studio 2015 или более новой версии. При этом, нужно дополнительно установить средства разработки приложений Windows. Также, нужно включить устройство Windows 10 для разработки. Возможна разработка на платформах Windows 7 и Windows 8.1 с рядом ограничений: приложения будет невозможно запустить под управлением этих платформ. В Windows 8.1 есть возможность использовать эмулятор.

Для разработки можно использовать следующие языки программирования: C#, Visual Basic, C++(C++/CX), JavaScript. Интерфейс описывается с помощью языков XAML или HTML[1].

Особенности процесса разработки. В целом, процесс разработки приложений для Универсальной платформы Windows схож с процессом разработки WPF-приложений. Для разработки интерфейса присутствуют конструктор и XAML-редактор. Интерфейс универсальных приложений оформляется в стиле Metro – стиль оформления графического интерфейса, который изначально был применен в Windows Phone, в котором основной акцент делается на текст, графический минимализм и плавную анимацию (один из наиболее отличительных элементов стиля – использование прямоугольных «плиток»). Этот стиль сейчас используется во всех платформах компании Microsoft. Это вносит некоторые изменения в процесс разработки. В стиле Metro появились некоторые новые элементы управления (например, CommandBar), а старые удалены (например, StripMenu). Также, элементы не имеют координат, а располагаются относительно соседних элементов или с помощью выравнивания (хотя присутствует элемент Canvas, в пределах которого можно использовать жесткие координаты). Кроме того, был изменен ряд других свойств элементов управления[2].

Появилась возможность посмотреть, как будет выглядеть приложение на том или ином устройстве при разных ориентациях экрана.

Код, написанный для Универсальной платформы Windows, будет одинаково выполняться на разных устройствах. Не смотря на это, можно добавить код для определенных платформ.

Приложения, разработанные для Универсальной платформы Windows, имеют ряд ограничений на доступ к файлам. Каждое новое приложение по умолчанию получает доступ к папке, в которую оно устанавливается, к папке, в которой расположены данные этого приложения, и к съемным устройствам. Доступ к дополнительным расположениям можно получить с помощью файла манифеста (файла, описывающего свойства программы или используемого ресурса) или с помощью диалога выбора файла.

Разработка функционала тестового приложения. Для разработки функционала тестового приложения был выбран язык C++ (точнее, C++/CX – расширение языка для платформы WinRT). Платформенезависимая составляющая C++/CX полностью совместима со стандартом C++11.

Тестовое приложение осуществляет открытие и сохранение файлов с помощью диалогов, и их обработку.

Большое количество методов, реализованных для Универсальной платформы Windows, являются асинхронными (особо сильно это коснулось работы с файлами). Такой подход является более сложным для программирования, но, при этом, более эффективным. Это привело к широкому распространению лямбда-выражений – специального синтаксиса, искользуемого для объявления анонимных методов.

Применение асинхронного программирования привело к необходимости контроля последовательности выполнения кода и гонки потоков. Например, в универсальных приложениях может быть отображено только одно диалоговое окно одновременно, а отображение выполняется асинхронно. Для отображения нескольких диалоговых окон по-очереди, создается очередь этих окон, и по событию закрытия диалогового окна отображается следующее в очереди:

```
list<MSG> q;
```

Метод отображения нового диалогового окна:

```
void TestTXTFileUWP::MainPage::ShowDialog(String^ Title, String^ Content,
String^ Button){
```

```
    //Добавление параметров окна в очередь
```

```
    q.push_back(MSG(Title, Content, Button));
```

```
    //Если в очереди более одного элемента, то окно уже отображается
    if (q.size() > 1) return;
```

```
    //Создание объекта диалогового окна
```

```
    ContentDialog^ WorkDialog = ref new ContentDialog;
```

```
    WorkDialog->Title = Title;
```

```
    WorkDialog->Content = Content;
```

```
    WorkDialog->CloseButtonText = Button;
```

```
    //Добавление обработчика события после закрытия диалогового окна
```

```
    WorkDialog->Closed += ref new TypedEventHandler< ContentDialog^,
```

```
ContentDialogClosedEventArgs ^>(this,
```

```
&TestTXTFileUWP::MainPage::DialogClosed);
```

```
    //Отображение окна
```

```
    WorkDialog->ShowAsync();
```

```
}
```

Обработчик события после закрытия диалогового окна:

```
void TestTXTFileUWP::MainPage::DialogClosed( ContentDialog ^sender,
```

```
ContentDialogClosedEventArgs ^e){
```

```
    //удаление из очереди текущего окна
```

```
    q.erase(q.begin());
```

```
    //Если очередь не пустая, то отображаем следующее окно
```

```
    if (q.size() > 0){
```

```
ContentDialog^ WorkDialog = ref new ContentDialog;
```

```
    WorkDialog->Title = q.begin()->Title;
```

```
    WorkDialog->Content = q.begin()->Content;
```

```
    WorkDialog->CloseButtonText = q.begin()->Button;
```

```
    WorkDialog->Closed += ref new
```

```
TypedEventHandler<ContentDialog ^, ContentDialogClosedEventArgs ^>(this,
```

```
&TestTXTFileUWP::MainPage::DialogClosed);
```

```
    WorkDialog->ShowAsync(); } }
```

Если же попытаться отобразить несколько диалоговых окон без создания очереди, будет вызвано исключение.

Разработка интерфейса тестового приложения. Для разработки интерфейса тестового приложения был использован язык разметки XAML. Тестовое приложение имеет следующий вид:

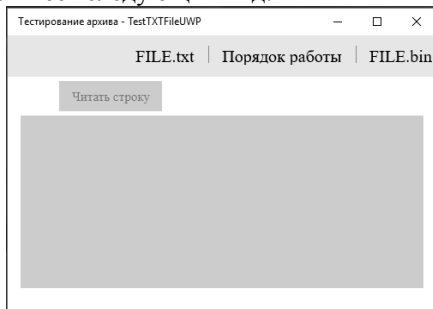


Рисунок 1 – Пример тестового приложения

В верхней части располагается CommandBar, на котором располагаются все команды. Этот элемент привязан к верхнему краю страницы. CommandBar состоит из трёх кнопок AppBarButton, Одна из них ведет себя схожим образом с обычной кнопкой. Две другие содержат всплывающее меню (MenuFlyout). Таким образом, реализован элемент, схожий с MenuStrip.

Дальнейшая разметка организована с помощью StackPanel – панель, располагающая дочерние элементы в одном столбце. Первым дочерним элементом является кнопка(Button), привязанная к левому краю с постоянным отступом. Вторым элементом является TextBox, привязанный к обоим краям.

Обработка событий осуществляется так же, как и в WPF.

Вывод. В результате проделанной работы был проведен обзор Универсальной платформы Windows и технологий для разработки универсальных приложений. Изучены особенности разработки интерфейса и функционала приложений для данной платформы. Разработана программа для Универсальной платформы Windows. Функционал реализован на языке C++, интерфейс – на языке XAML. Приложение было протестировано под управлением операционной системы Windows 10.

Список литературы

1. MSDN – сеть разработчиков Microsoft [Электронный ресурс]. – URL: <https://msdn.microsoft.com/ru-ru>
2. Moemeka Edward, Moemeka Elizabeth, Real World Windows 10 Development: Apress, 2015. – 607с.

УДК 004.94

РЕАЛИЗАЦИЯ ПОШАГОВОГО ОНЛАЙН-УМНОЖИТЕЛЯ ДВОИЧНЫХ ЧИСЕЛ

С.Н. Омелянченко, Р.В. Мальчева

Донецкий национальный технический университет, г.Донецк,
кафедра компьютерной инженерии

Аннотация

Омелянченко С.Н., Мальчева Р.В. Реализация пошагового онлайн-умножителя двоичных чисел. Реализация пошагового онлайн-умножителя двоичных чисел в разных форматах, разными способами на основе методических материалов учебного курса архитектуры компьютера.

Ключевые слова: алгоритмы умножения, онлайн-умножитель, PHP, javascript, учебный курс архитектуры компьютера.

Постановка проблемы. С изучением архитектуры компьютеров многие сталкиваются с проблемой досконального понимания работы алгоритмов умножения и неверного изучения, а в дальнейшем, допущения ошибок при выполнении задачи умножения и проектирования различных схем, один из процессов который основан на умножении двоичных чисел в различных форматах. При разных схемах удобны различные алгоритмы умножения, потому понимание и правильное выполнение очень важно. Возможным решением является подробное наглядное объяснения с возможностью быстрой проверки, что облегчит задачу изучения.

Цель статьи – разработка сайта со встроенным онлайн-умножителем, отражающим все пошаговые действия каждого из алгоритмов.

Анализ существующих решений

При анализе сервисов умножения чисел в машинном коде [1] можно столкнуться с тем, что не совсем наглядно или же не до конца понятно как правильно работает алгоритм. Теоретическая часть присутствует, а практической нет, или же практическая часть не до конца понятна. Наглядный пример имеет некоторые недостатки, что легко устраняются перефразировкой картины пошагового алгоритма умножения. Так же возможность выбора формата или же способа умножения отсутствует.

Описание задачи реализации алгоритмов умножения двоичных чисел

Суть задачи заключается в том, что существует различные форматы представления чисел в машинном формате [2]:

- ПК(прямой код);
- ДК(дополнительный код);

Умножение двоичных чисел

езде записываем нули, если 1 - то переписываем первое число.

					1	0	1	0	1
				1	1	1	1	0	1
=	=	=	=	=	=	=	=	=	=
					1	0	1	0	1
				0	0	0	0	0	
			1	0	1	0	1		
		1	0	1	0	1			
	1	0	1	0	1				

Рисунок 1 – Пример онлайн умножителя чисел в машинном коде

- ОК(отрицательный код);
- ОН(отрицательный ноль);
- ПН(положительный ноль)
- и др.

Разные алгоритмы умножения:

- Самые простые 4 варианта умножения;
- Несколько вариантов ускоренного умножения;

На основании этого реализация пошагового онлайн-умножителя должна состоять из двух частей. Программный код перевода из 10го формата чисел в двоичный, затем в один из форматов необходимого кода. А вторая часть состоит из реализации алгоритмов умножения с демонстрацией пошагового умножения чисел в нужных форматах.

Реализация пошагового онлайн умножителя

Реализация пошагового онлайн умножителя основана на создании web-странички на языке разметки HTML[3], подключения функций обработки за счет java-script и осуществление алгоритмических функций на языке PHP[4].

PHP это скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов. На PHP проще всего работать с числами, «переменными», массивами и осуществлять такие вещи как сдвиги, разделение и склеивание строк, переводы данных из

одного в другие форматы за счет встроенных функций языка, что облегчает реализацию заданной задачи.

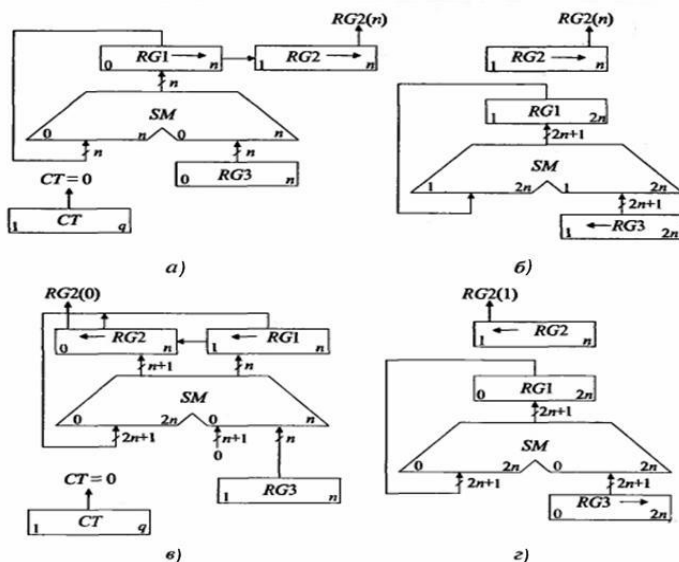


Рисунок 2 - Операционные схемы устройств для умножения чисел:
а) первый способ; б) второй способ; в) третий способ; г) четвертый способ

Ниже приведен фрагмент кода простейшего преобразования числа в машинный код.

```

$a = 23;
$a_b = toBinary($a);
print_r( $a_b);

function toBinary($val){
    $res = array();
    while($val != 0){
        $res[] = $val % 2;
        $val = (integer)( $val / 2 );
    }
    $res=array_reverse($res);
    return $res;
}

```

Рисунок 3 – Пример кода простейшего преобразования числа в машинный код

Так же реализация использования базы данных при внедрения в веб-сайт регистрационной формы, а значит использование одной из самых удобных и эффективных систем управления базами данных MySQL.

Проблемы реализации и пути их решения

Основной проблемой реализации веб-сайта с пошаговым онлайн множителем двоичных чисел в различных форматах разными способами умножения является то, что необходимо достаточно подробно и наглядно показать варианты умножения в двоичном коде. Важно преподнести информацию достаточно понятно, чтобы изучение стало более продуктивным, приятным и эффективным.

Возможным вариантом решения проблемы может являться поиск некоторых деталей, что помогают облегчить изучение алгоритмов умножения за счет большого количества совершенно простых задач. При разбиении большой сложной задачи на более маленькие реализация и понимание обычно проходит гораздо проще и продуктивнее. Следует отметить, что облегчение задачи позволяет качественнее изучать предмет, а значит и дальнейшая работа студента будет более продуктивной.

Моя дальнейшая работа будет посвящена реализации в первую очередь самого онлайн множителя, а так же подробного описания алгоритмов умножения для более качественного изучения материала.

Выводы. В ходе работы было выполнено изучение алгоритмов умножения двоичных чисел в разных форматах и различными способами, а так же изучение языков разметки веб-странички, скриптовых языков и их возможности работы с серверами. Воссоздание алгоритмов умножения на бумаге для точной реализации их в практической части дипломной работы.

Список литературы

1. <http://math.semestr.ru/inf/multiplication.php>
2. Мальчева Р.В. Разработка виртуальной лаборатории для изучения и моделирования архитектур процессорных элементов / Р.В. Мальчева, О.А. Авксентьева // Программная инженерия: методы и технологии разработки информационно- вычислительных систем (ПИИВС-2016): сборник научных трудов I научно-практической конференции. 16-17 ноября 2016 г. –Донецк, ГОУ ВПО «Донецкий национальный технический университет», 2016. –С. 102-108.
3. <http://htmlbook.ru/>
4. <http://www.php-book.ru/>
5. <https://ospanel.io/>

УДК 004.42

ИССЛЕДОВАНИЕ МЕТОДОВ ФОРМИРОВАНИЯ АВТОМАТИЗИРОВАННОГО РАСПИСАНИЯ

Иванов В.Ю., Завадская Т. В., Меренкова Л.Л.

ГОУВПО "Донецкий национальный технический университет"
кафедра компьютерной инженерии.

E-mail: ivanov.valerii.yuriiovych@mail.ru

Аннотация

Иванов В.Ю., Завадская Т.В., Меренкова Л.Л. Исследование методов формирования автоматизированного расписания. Выполнен анализ метода формирования автоматизированного расписания. Определены основные критерии для формирования расписания и алгоритм. Построен ряд функций для достижения качества расписания, а так же спроектированы этапы формирования расписания.

***Ключевые слова:** расписание, этапы расписания, критерии расписания, алгоритм формирования расписания, генетический алгоритм.*

Постановка проблемы. Работа по составлению расписания требует значительных трудовых и временных затрат, так как на данный момент, выполняется вручную дважды в год и корректируется в течение семестра. Поэтому, автоматизация наиболее трудоемких процессов может существенно облегчить ее выполнение и повысить качество управления учебным процессом. Форма составления расписания должна представлять собой фактически те бумажные формы, которые используются при составлении расписания на текущий момент, но с автоматизацией тех процессов, которые не требуют от человека принятия решения, однако процесс принятия решений по всем спорным моментам в составлении расписания должен остаться точно такой же, как и при ручной работе.

Цель работы – основной задачей является исследование алгоритмов формирования расписания в университете. В результате, на базе информации из учебных планов и нагрузки преподавателей, необходимо разработать программу, в которой будет заложен алгоритм автоматического формирования базового расписания в рамках всего ВУЗа.

Анализ метода формирования расписания. В результате обзора и исследования существующих методов и моделей построения расписаний

занятий в ВУЗе были отобраны 2 модели «Генетического алгоритма», опирающегося на систему критериев оптимизации расписания.

Расписание формируется в 2 этапа:

1. Построение обезличенного расписания с учетом критериев оптимизации только для групп;
2. Оптимизационный этап построение оптимального расписания с учетом критериев оптимизации по преподавателям.

Формирование расписания предусматривает создание коллекции расписаний с учетом настроечных коэффициентов и критериев оценки качества. Каждый экземпляр расписания проверяется на его эффективность.

В результате сравнения экземпляров производится выбор наиболее эффективной модели, которая будет выступать в качестве «БАЗОВОГО» расписания. В него позже могут вноситься текущие изменения диспетчерами расписания ВУЗа с помощью отдельного программного интерфейса.

В качестве базовых данных для построения расписания используются данные нагрузки преподавателей, полученные с помощью программ «нагрузка кафедры» и «нагрузка ППС (Профессорско-преподавательский Состав)».

При разработке расписания используется «Генетический алгоритм». В нем предлагается использовать в качестве генов настроечные коэффициенты w_j алгоритма составления расписания. В результате мутаций будут получены новые наборы настроечных коэффициентов и как следствие – результаты работы алгоритма будут различны. При этом оценку качества составленного расписания следует проводить с использованием значений w_j заданных диспетчером.

Таким образом, учитывается возможность генерации расписания, наиболее удовлетворяющего оценкам качества с точки зрения диспетчера, при использовании несколько измененных оценок качества в процессе его генерации.

Хромосома такого генетического алгоритма будет представлять собой набор действительных чисел: $\{w_1, w_2, w_3, \dots, w_m\}$, где $w_1 \dots w_m$ – гены хромосомы, настроечные коэффициенты алгоритма составления расписания;

m – количество настроечных коэффициентов алгоритма.

Целью работы генетического алгоритма является достижение максимума функционала R :

$$R_{il} = \sum_{j=1}^m w_j k_{jl} \quad (1)$$

$$R_i = \max_l (R_{il}), l = 1, \dots, h \quad (2)$$

$$R = \sum_{i=1}^n R_i \rightarrow \max, \quad (3)$$

где Ril – качество расположения i -го занятия на l -й позиции в расписании;
 K_{jl} – значение, полученное по j -му критерию оценки качества
расположения занятия на l -й позиции в расписании;

w_j – весовой коэффициент j -го критерия оценки качества;

m – количество критериев оценки качества;

l – возможная позиция i -го занятия в расписании;

Ri – качество расположения i -го занятия в расписании;

h – количество возможных вариантов расположения занятия в расписании,

R – Качество составленного расписания [1].

В качестве основных настроечных коэффициентов и критериев качества
выбраны следующие данные:

1. Проверка на вместимость аудиторий
2. Проверка назначения аудиторий
3. Очередность занятий (лекции в начале дня - 1, 2 пара)
4. Наличие окон в занятиях группы
5. Наличие окон в занятиях преподавателя
6. Наличие свободных дней у преподавателя (лектора/ ассистента)
7. Нагрузка в пятницу должна быть меньше
8. Первое занятие в понедельник должно быть простым (1 группа циклов)
9. Учет райдера ППС с учетом ранга их должностей
10. Количество лекций в день (максимум)
11. Количество лабораторных занятий в день (макс)
12. Количество практических занятий в день (макс)
13. Количество перемещений между корпусами группы
14. Количество перемещений между корпусами ППС

Для каждого критерия определен свой приоритет и значимость (W_j), а
также определена формула расчета коэффициента качества (K_j)

На рис. 1 представлены таблицы, которые были спроектированы в БД для
данной задачи. Они были спроектированы так чтобы могли взаимодействовать с
таблицами задач «нагрузка кафедры» и «нагрузка ППС». Все эти таблицы
участвуют при формировании расписания. Основная таблица, которая, по сути, и
будет тем самым расписанием – «RASP_GRUPPA_DISCIP_AUD_PPS». Данная
таблица содержит в себе внешние ключевые поля, по которым связывается с
остальными таблицами. Кроме этого содержит в себе такие поля как «Код
дисциплины», «День недели», «Параметр недели» (верхняя/нижняя), «Код
аудитории», «Табельный номер преподавателя»

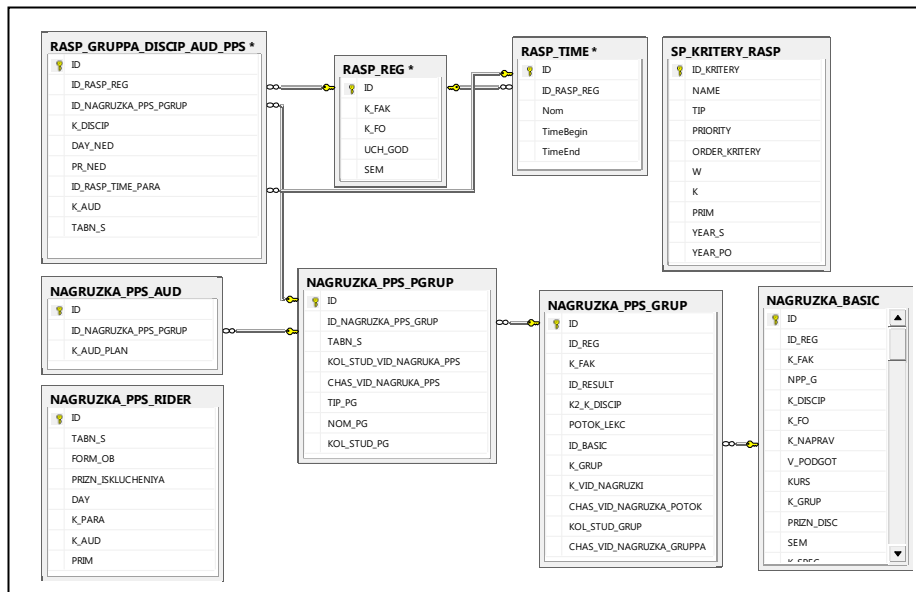


Рисунок. 1 Диаграмма связей основных таблиц БД, используемых для построения расписания

Этапы формирования расписания.

Проанализировав имеющиеся данные, в результате были разработаны этапы формирования расписания:

1. Формируем расписание поочередно по каждому факультету вуза
2. Установка главных параметров регистрации расписания:
 - 2.1 Факультет
 - 2.2 Учебный год
 - 2.3 Семестр
 - 2.4 Форма обучения
3. Формирование справочника расписания учебных дней и пар (количество и временной диапазон) для каждого зарегистрированного расписания
4. Выборка «базовой» для расписания информации из нагрузки по кафедрам для заданных параметров расписания (п.2);
5. Сортировка базовой информации по:
 - 5.1. Виду подготовки, курсу, направлению подготовки (бакалавры, магистры) / специальности (специалисты);
 - 5.2. По группе;
 - 5.3. По дисциплине;
 - 5.4. По виду нагрузки (лекции, практические, лабораторные).

6. Построение 1 модели «обезличенного» расписания с программным учетом критериев оптимизации для групп. Организация циклов программируется согласно параметрам сортировки «базовой информации»;

7. В качестве основного объекта для формирования расписания выступает объект - «Поток групп/группа/ подгруппа».

8. Расчет качества модели Ri .

9. Расчет модели с поочередным учетом критериев качества для группы. Расчет нового качества модели Ri .

10. Выбор модели с максимальным Ri .

11. Построение 2 модели «генетического» расписания с программным учетом критериев оптимизации для ППС. Организация циклов программируется согласно параметрам сортировки «базовой информации».

12. Расчет качества модели Ri .

13. Расчет модели с поочередным учетом критериев качества для ППС. Расчет нового качества модели Ri .

14. Выбор модели с максимальным Ri .

15. Сравнение всех полученных моделей и выбор оптимального варианта расписания $max (Ri)$.

Выводы. Проведен анализ методов формирования расписания. Был определен алгоритм, с помощью которого будет улучшаться качество базового расписания путем достижения максимума построенных функций. В результате были определены основные критерии, которые будут участвовать в этих функциях как настроечные коэффициенты. Были спроектированы таблицы в базе данных для достижения программной реализации этой проблемы, так же были сформированы этапы формирования расписания.

Список литературы

1. Береговых Ю.В. Алгоритм составления расписания занятий / Ю.В. Береговых, Б.А. Васильев, Н.А. Володин // Искусственный интеллект. 2009. № 2. С. 50-57.

УДК 519.725, 625.7, 681.3

СТРУКТУРА СИСТЕМЫ ЗОНДОВОГО ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ ПЛАТ

Чаган А.Н., Ярошенко Николай Александрович

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

Empty2410@gmail.com

Аннотация

Чаган А.Н., Ярошенко Н.А. Структура системы зондового диагностирования цифровых плат. Выполняется разработка и обоснование структуры технических средств системы зондовой диагностики цифровых устройств, реализованных в виде типовых элементов замены (ТЭЗ), собранных из ИМС малой, средней и большой степени интеграции, до выводов которых есть доступ с помощью ручного либо роботизированного зонда. Описывается состав и назначение компонентов системы.

***Ключевые слова:** генератор тестов, РСЛОС, ГПСТ, сигнатурный анализатор, пост контроля*

Введение. В настоящее время на территории стран СНГ и в частности Украины России, используется большое количество традиционной радиоэлектронной и электронно-вычислительной аппаратуры [1,2]. Ее полная замена новой техникой является дорогой, поэтому возникает необходимость в ее диагностировании и ремонте. Анализ отечественного рынка и стран СНГ показывает отсутствие действующих систем диагностирования с функцией локализации неисправностей. На мировом рынке такие системы есть, однако их стоимость составляет \$500 тыс. и выше, что очень дорого для отечественного потребителя. В свою очередь анализ отечественных и зарубежных литературных источников показывает практическое отсутствие оптимальных алгоритмов поиска неисправностей. С одной стороны, это объясняется высокой трудоемкостью и сложностью разработки таких алгоритмов. С другой стороны, алгоритмы, которые решают задачу поиска неисправностей, скорее всего, есть, но фирмы-разработчики их не раскрывают [3].

Разработка и обоснование структуры системы ЗД

Целью статьи является разработка структуры технических средств системы ЗД ЦУ, реализованных в виде ТЭЗ (плат), собранных из ИМС различной степени интеграции, до выводов которых есть доступ с помощью ручного либо роботизированного зонда.

Вычислительный комплекс системы ЗД представлен на рис.1.

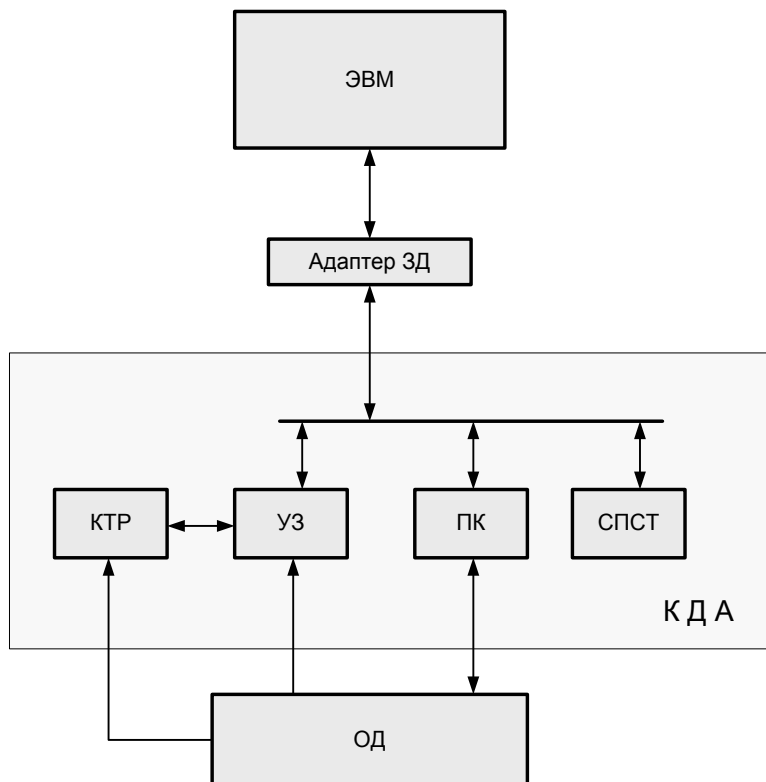


Рисунок 1 – Вычислительный комплекс системы ЗД

Основу комплекса составляет инструментальная ЭВМ, в качестве которой используется персональный компьютер (ПЭВМ). Контрольно-диагностической аппаратуры (КДА) состоит из специализированных устройств ЗД ОД. ЭВМ комплекса выполняет "интеллектуальные" функции системы ЗД: подготовку и обработку информационного обеспечения ОД, ведение баз данных, генерацию детерминированных (ДТ) и псевдослучайных тестов (ПСТ), реализацию алгоритмов ЗД, диалог диагноста, ведущего поиск, с ПЭВМ и другое.

На КДА возлагаются следующие функции:

- генерация ДТ и ПСТ;
- съем и обработка тестовых реакций (ТР);
- управление синхронизацией ОД;

- сопряжение с органами управления и индикации зонда;
- сопряжение ОД с ПЭВМ;

Реализацию указанных функций КДА осуществляет с помощью следующих составляющих ее устройств:

- одного или нескольких постов контроля (ПК);
- сопроцессора псевдослучайных тестов (СПСТ);
- устройства зондирования (УЗ);
- компаратора тестовых реакций (КТР);
- адаптера ЗД.

Пост контроля вычислительного комплекса системы ЗД предназначен для генерации ТП по внешнему интерфейсу ОД. Он выполняет следующие основные функции:

- выдачу ТП на входы ОД в старт-стопном режиме;
- генерацию ПСТ и выдачу их на рабочей частоте на входы ОД;
- съем (без преобразования) с внешних разъемов ТР в старт-стопном режиме и передачу их в ЭВМ;
- сжатие внешних ТР по методу сигнатурного анализа (СА) на рабочей частоте ОД;
- сопряжение на физическом уровне с ОД и ПЭВМ;
- обмен информацией с ЭВМ.

Тестирование ОД в проектируемой системе ЗД может выполняться на ДТ, ПСТ или комбинированных тестах, вырабатываемых программным или аппаратным способом [1,2]. В первом случае тесты вырабатываются программой генерации тестов, во втором - встроенным в ПК генератором псевдослучайных тестов (ГПСТ) либо с помощью СПСТ. ДТ вырабатываются только программным способом. Для генерации ПСТ может применяться как программный, так и аппаратный способы. Комбинирование тестов предполагает сочетание "во времени" и "в пространстве" ДТ и ПСТ.

Реализацию описанных функций ПК осуществляет с помощью следующих основных узлов:

- памяти ТР;
- СА;
- простейшего ГПСТ (РСЛОС);
- смесителя тестов;
- селектора выводов;
- интерфейсного узла.

Память ТР выполняет следующие функции: прием от ЭВМ, временное хранение и выдачу на рабочей частоте ОД фрагмента теста; съем с внешних выходов ОД, временное хранение и передачу в ЭВМ фрагмента ТР.

Сигнатурный анализатор выполняет сжатие ТР ОД по методу СА. Он является многофункциональным узлом [4].

Задачей ГПСТ является генерирование ПСТ на рабочей частоте ОД без приостановки процесса генерации. Для обеспечения такой возможности генератор строится на базе РСЛОС, который вырабатывает М-последовательности [2].

Сочетание ДТ с ПСТ выполняется с СП. Селектор выводов предназначен для настройки выводов ПК на ввод, вывод либо на двунаправленный обмен тестов. Интерфейсный узел выполняет задачу сопряжения узлов ПК с ПЭВМ на физическом уровне.

Задачей СПСТ является выработка за один машинный цикл 16 разрядного псевдослучайного кода, который через ПЭВМ передается в каналную память ПК. Благодаря этому ценой небольших аппаратурных затрат удастся увеличить на два порядка время формирования ПСТ по сравнению с программным способом.

Сопроцессор состоит из следующих основных узлов: РСЛОС, преобразователя вероятностей (ПВ), задатчика вероятности (ЗВ), регистра кода ПСТ, узла сопряжения с ПЭВМ.

Устройство зондирования (УЗ) обеспечивает следующие основные функции: съём ТР без преобразования и съём ТР со сжатием. При выполнении первой функции ТР заносятся без преобразования во внутреннюю память УЗ для последующей передачи реакции в ПЭВМ, где над ней будет произведен далее логический анализ. Так как память устройства ограничена по емкости, то съём полных ТР производится в старт/стопном режиме.

Сжатие ТР выполняется по одному из следующих методов: сигнатурный анализ, счет уровней и счет перепадов.

Выполнение основных функций обеспечивается рядом дополнительных функций:

- синхронизаций процесса съёма ТР как от ОД, так и от внутреннего генератора УЗ;
- отображением служебной информации для диагноста, ведущего зондирование ОД;
- реакцией на управляющие воздействия;
- программно-управляемым способом настройки режимов и параметров функционирования УЗ.

УЗ состоит из следующих основных узлов: памяти ТР, компрессора ТР, схемы синхронизации, схемы сопряжения с органами управления и индикации зонда, схемы управления процессом зондирования, узла сопряжения с ПЭВМ. КТР состоит из усилителя сигналов, АЦП, компаратора логических уровней, задатчика логических уровней.

Выводы

Выполнена разработка и дано обоснование структуры технических средств системы зондовой диагностики цифровых устройств, реализованных в виде типовых элементов замены, собранных из ИМС малой, средней и большой степени интеграции, до выводов которых есть доступ с помощью ручного либо роботизированного зонда. Описывается состав и назначение компонентов системы. Полученные результаты могут найти применение при проектировании внешнего тестового оборудования систем ЗД, НИРС, дипломного проектирования и при выполнении магистерских работ.

Список литературы

1. Ярмолик В.Н. Контроль и диагностика цифровых узлов ЭВМ. – Мн.: Наука и техника, 1988. – 240 с.
2. В.Ю. Скобцов, Ю.А. Скобцов. Логическое моделирование и тестировании цифровых устройств. - Донецк: ИПММ НАНУ, 2005.-436 с.
3. Сперанский Д.В., Черевко Н.В. О сокращении числа зондовых проб при диагностировании цифровых устройств // Электронное моделирование. - 1992, N 3., С. 56 - 62.
4. Дяченко О.Н. Метод аналитического вычисления сигнатур // Сб. тр. ФВТИ ДонГТУ. Вып. 1. - Донецк: ДонГТУ, 1996.- С.97-102.

УДК 004.457

ОСОБЕННОСТИ ОПРЕДЕЛЕНИЯ КОДИРОВКИ СИМВОЛОВ В С#

**К.С. Захарченко (3 курс, каф. КИ), С.В. Теплинский, к.т.н., доц.
Л.И. Дорожко, к.т.н., доц,**

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии

Аннотация

Захарченко К.С., Теплинский С.В., Дорожко Л.И. Особенности определения кодировки символов в С#. Выполнен анализ методики определения кодировки символов в С#. Определены основные особенности определения кодировки символов в С#.

Ключевые слова: Юникод, кодовая страница, кодировка символов.

Постановка проблемы. Иногда бывает так, что текст, состоящий из букв русского алфавита, полученный с другого компьютера, невозможно прочитать. Это происходит оттого, что на компьютерах применяется разная кодировка символов русского языка. Таким образом, возникает необходимость определить, как с этой проблемой справляется С#.

Цель статьи – провести анализ методики определения кодировок символов строки, отличных от используемой по умолчанию, в С#. Показать механизм ее работы.

О кодировках UTF. Большинство кодировок работают с набором символов, которые являются лишь малой частью UCS(Universal Coded Character Set). Это становится проблемой для многоязычных данных, вот по этому необходима кодировка, которая использует все символы UCS. Кодировка 8-битных символов очень простая, так как вы получаете один символ из одного байта, но UCS использует 31 бит и вам необходимо 4 байта на символ. Появляется проблема порядка байтов так как некоторые системы используют порядок от старшего к младшему, другие - наоборот. Так же часть байтов всегда будут пустыми, это пустая трата памяти. Правильная кодировка должна использовать различное количество байтов для различных символов, но такая кодировка будет эффективная в одних случаях и не эффективна в других.

Решение этой головоломки – использовать несколько кодировок из которых вы можете выбрать подходящую. Они называются Unicode Transformation Formats, или UTF.

UTF-8 – самая распространенная кодировка в Интернете. Она использует для ASCII символов один байт, а для всех остальных символов UCS 2 или 4

байта. Это очень эффективно для языков использующих латинские буквы, так как они все входят в ASCII, достаточно эффективно для Греческого, Кириллицы, Японского, Армянского, Сирийского, Арабского и др., так как для них используется 2 байта на символ. Но это не эффективно для всех остальных языков из BMP (Basic Multilingual Plane), так как будет использоваться 3 байта на символ, а для всех остальных символов UCS, например Готическое письмо, будет использоваться 4 байта.

UTF-16 использует для всех символов из BMP одно 16-битное слово и два 16-битных слова для всех остальных символов. По этому, если вы не работаете с одним из упомянутых выше языков, вам лучше использовать UTF-16. Так как UTF-16 использует 16-битные слова, мы получаем проблему порядка байтов. Она решена наличием трех вариантов: UTF-16BE для порядка байтов от старшего к младшему, UTF-16LE - от младшего к старшему, и просто UTF-16, который может быть UTF-16BE или UTF-16LE, при кодировании в начале используется маркер, который указывает порядок байтов. Этот маркер называется "byte order mark", или "BOM".

Так же существует UTF-32, который может быть в двух вариантах BE и LE как и UTF-16, и хранит символ Юникод как 32-битное целое число. Это не эффективно почти для всех символов, кроме тех, которые требуют 4 байта для хранения. Но при этом очень легко обрабатывать такие данные, так как у вас всегда 4 байта на символ.

Важно разделять закодированные данные от данных Юникод. Поэтому не думайте о UTF-8/16/32 данных как о Юникод. Таким образом, хотя кодировки UTF и определены в стандарте Unicode, в рамках модели понимания Юникода считаем что UTF - это не Юникод [1].

Определение кодировки символов строки в C#. В C# для кодировки символов по умолчанию используется кодировка Unicode (UTF-8), стоит проверить как он поведет себя с другими кодировками.

Итак, имеем файлы с разными кодировками (ANSI, UTF-8, Unicode и Unicode BE, в частности) со следующим текстом: «Hello! Привет!».

В языке C# для чтения текстового файла воспользуемся классом StreamReader. Перегрузим конструктор двумя параметрами: путь к файлу и значение «истина» для параметра автоматического определения кодировки по метке порядка следования байтов [2].

```
using (var sr = new StreamReader(of.FileName, true))
{
    textBox.Text = sr.ReadToEnd()+" " sr.CurrentEncoding.EncodingName;
}
```

Теперь поочередно передаем классу файлы в разных кодировках и следим за результатом. Файл с кодировками из семейства Unicode были прочитаны

корректно (рис. 1-3). Однако, из файл с кодировкой символов ANSI была получена лишь половина символов (рис.4).

```
--UTF-8.txt--Unicode (UTF-8)  
Hello! Привет Unicode (UTF-8)
```

Рисунок 1 – Файл с кодировкой UTF-8

```
--Unicode.txt--Unicode  
Hello! Привет Unicode
```

Рисунок 2 – Файл с кодировкой Юникод

```
--Unicode BE.txt--Юникод (Big-Endian)  
Hello! Привет Юникод (Big-Endian)
```

Рисунок 3 – Файл с кодировкой Юникод Big Endian

```
--ANSI.txt--US-ASCII  
Hello! ❖❖❖❖ Unicode (UTF-8)
```

Рисунок 4 – Файл с кодировкой ANSI прочитан не верно

Теперь вызовем другой конструктор класса StreamReader перегрузим его, указав вторым параметром кодовую страницу 1251.

```
using (var sr = new StreamReader(of.FileName, Encoding.GetEncoding(1251)))  
{  
    textBox.Text = sr.ReadToEnd() + " " + sr.CurrentEncoding.EncodingName;  
}
```

В результате получаем полный текст файла с кодировкой ANSI (рис.5). На правильность чтения файлов с кодировками Юникод изменение кодовой страницы не повлияло.

```
--ANSI.txt--US-ASCII  
Hello! Привет Кириллица (Windows)
```

Рисунок 5 – Файл с кодировкой ANSI прочитан верно

Аналогичный эксперимент был проведен с файлом в кодировке Mac. Используя конструктор класса StreamReader без указания кодировки – файл читается не верно (рис.6), при использовании же метода с указанием кодовой страницы Mac (10007) – результат полный и правильный (рис.7).

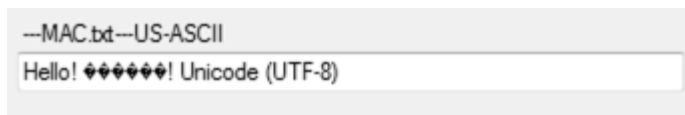


Рисунок 6 – Файл с кодировкой Mac прочитан не верно

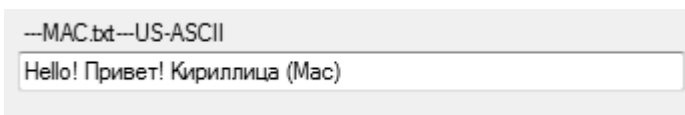


Рисунок 7 – Файл с кодировкой Mac прочитан верно

Как показали проведенные эксперименты – надежного способа определения произвольной кодовой страницы не существует, хотя и проводились попытки создать его, основываясь на вероятности обнаружения определенных последовательностей байтов в тексте. В StreamReader это не реализовано [3,4].

Вывод. На основе изложенной выше информации была разработана программа, которая определяет кодировку текстового файла, отображает кодировку на основании BOM(маркер порядка следования байт) выбранного текстового файла и его текст.

Результаты проведенных экспериментов показали, что C# успешно справляется с определением кодировки текстовых файлов, однако все же требует предварительной подстройки.

Список литературы

1. Unconfusing Unicode: What is Unicode? [электронный ресурс] // Lennart Regebro: Python, Plone, Web [сайт]. [2015]. URL: <https://regebro.wordpress.com/2011/03/23/unconfusing-unicode-what-is-unicode/>
2. Юникод [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2017]. URL: [www.wikipedia.org/wiki/ Юникод](http://www.wikipedia.org/wiki/Юникод)
3. Microsoft Developer Network. [сайт]. URL: <https://msdn.microsoft.com/>
4. Гордеев А.В, Молчанов А.Ю. Системное программное обеспечение. – СПб.: Питер, 2002. – 736 с.: ил.

УДК 004.457

ОСОБЕННОСТИ ПОИСКА УДАЛЕННЫХ ФАЙЛОВ И КАТАЛОГОВ В ФАЙЛОВОЙ СИСТЕМЕ NTFS

М.А. Седневц (4 курс, каф. КИ), С.В. Теплинский, к.т.н., доц.
Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии

Аннотация

Седневц М.А., Теплинский С.В. Особенности поиска удаленных файлов и каталогов в файловой системе NTFS. Выполнен анализ методики поиска файлов и каталогов пригодных для восстановления в файловой системе NTFS. Определены основные особенности поиска файлов и каталогов.

***Ключевые слова:** файловая система NTFS, поиск удаленных файлов.*

Постановка проблемы. Потеря информации – Довольно частое хоть и неприятное явление особенно среди неопытных пользователей в зависимости от важности данных их утрата может привести к разного рода проблемам в том числе финансовым, да и потеря личных данных событие неприятное.

На данный момент проблема поиска и восстановления удаленных данных в файловой системе NTFS решена. На просторах интернета можно найти множество программ однако информации о самих принципах поиска удаленных файлов и их восстановления не так много из за чего тема не перестает быть интересной для исследования. Различные утилиты используют разные подходы следовательно и результаты их работы отличаются. В данной статье мы проведем исследование одного из алгоритмов «ручного» восстановления данных и разработаем свой алгоритм поиска удаленных файлов.

Цель статьи – провести анализ методики поиска удаленных файлов и каталогов в файловой системе NTFS. Разработать алгоритм поиска удаленных файлов и каталогов пригодных для дальнейшего восстановления.

Общие сведения. Файловая система NTFS была разработана компанией Microsoft для использования в новой на тот момент операционной системе Windows NT. NTFS должна была заменить устаревшую файловую систему FAT имеющую множество ограничений которые не позволяли на основе этой системы создать полноценно многопользовательскую и многозадачную ОС [1].

Целью было разработать безопасную, надежную, расширяемую систему которая была бы способна не только полноценно заменить FAT но обладать рядом преимуществ. Как мы видим разработка была удачной на данный момент Windows одна из самых популярных операционных систем по всему миру.

Файлы. В архитектуре файловой системы NTFS абсолютно все данные, в том числе и разнообразные служебные структуры самой файловой системы, представлены в виде файлов. И даже сверх этого сама файловая система представлена в виде файлов.

Также в виде файлов представлены и административные данные файловой системы, в иных операционных системах подобные данные хранятся в скрытых областях с фиксированным адресом. Как раз из-за того что в NTFS к примеру таблица разделов хранится в виде файла. NTFS нет нужды резервировать какие-либо области т.к. файлы могут располагаться (физически) в любом месте тома.

В случае необходимости файлы могут менять свой размер к примеру файловые таблицы заметно прибавляют в весе при хранении на томе большого числа фалов. При всем при этом используются стандартные средства работы с файлами. Также не является проблемой и отсутствие непрерывного участка для хранения данных системы если такого не найдено то файл будет фрагментирован и все равно сохранен.

В отличии от других систем в NTFS нет жесткой структуры. В NTFS вся файловая система рассматривается как область данных а значит и любой файл может быть сохранен где угодно. Исключениями являются загрузочный сектор и загрузочный код расположенные в первых секторах тома.

Файловая таблица MFT. Информация о фалах и каталогах содержится в главной файловой таблице MFT. Даная таблица содержит информацию обо всех файлах и каталогах. Каждому файлу или каталогу соответствует как минимум одна запись.

Формат записи MFT. Формально размер записи MFT содержится в одной из переменных в загрузочном секторе. Но на практике все версии Windows используют фиксированный размер в 1024 байта. Из которых заголовок расположенный в начале файла размером 42 байта и содержащий 12 полей. И 982 байта без четкой структуры который используются для хранения атрибутов.

Адреса записей MFT. Адресация записей MFT производится в 48-битной системе начиная с 0, то есть первая запись будет находиться по 0 адресу. Адрес же последней записи будет изменяться в процессе увеличения MFT. При желании адрес можно узнать разделив размер файла \$MFT на 1 КБ.

В таблице MFT записи пронумерованы. Каждая запись имеет порядковый номер (16 разрядов). Номер автоматически увеличивается при добавлении новой записи.

Адрес файла формируется по такому принципу. Адрес записи в таблице объединяется с порядковым номером (старшие 16 бит) на выходе получаем уникальный базовый адрес файла (64- разряда)

Для обращения к записям используют порядковый номер который как очевидно уникален для каждой записи. Использования такого обращения имеет ряд преимуществ к примеру помогает выявить поврежденный файла рассмотрим

небольшой пример. Если ошибка произошла в момент выделения структур данных для нового файла то по уникальному номеру возможно определить какому файлу принадлежит запись новому или предыдущему. Следовательно данный метод используют программы для восстановления данных [2].

Атрибуты записей MFT. Как уже говорилось в NTFS в отличии от FAT нет жесткой структуры записей. Отдельная запись имеет минимальное структурирование. Иметься заголовок и место для хранения атрибутов. При этом атрибутом может являться что угодно от даты и времени до самого файла.

В атрибутах может храниться самая разнообразная информация и естественно разные типы информации могут записываться в разных форматах и занимать разное количество места в записи.

Алгоритм ручного восстановления с использованием Disk Editor и CHKDSK:

1. Извлекаем из boot-сектора указатель на MFT
2. Читаем из MFT первую запись (данная запись описывает \$MFT)
3. Находим атрибут \$DATA располагающийся по смещению(80h)
4. Декодируем список отрезков (data runs)
5. Последовательно читаем все записи в MFT
6. Анализируем содержимое атрибута \$FILE_NAME(30h) (имя файла) в данном атрибуте также храниться ссылку на материнскую директорию
7. Имена файлов хранятся по смещению EAh от начала сектора в начале которого расположена сигнатура FILE используя любой дисковый редактор с помощью функции поиска вводим имя восстанавливаемого файла в юникоде и ищем его.
8. Когда искомое значение будем найдено проверяем находится ли в начале сектора сигнатура FILE если-же она не найдена продолжаем поиск (Флаги: 00h – запись не используется или была удалена, 01h – запись используется и описывает каталог, 02h – запись используется и описывает директорию)
9. Исправляем 00h на 01h и записываем изменения
10. Вносим изменения в файл /\$MFT: \$BITMAP чтобы данная запись вновь использовалась
11. Удаляем из /\$BITMAP номера кластеров используемых восстановленным файлом
12. Перестраиваем двоичное дерево индексов хранящие содержимое каталога

Последние три пункта удобнее выполнять с помощью CHKDSK [3]

Представленный алгоритм прекрасно демонстрирует через что предстоит пройти пользователю при ручном восстановлении файла и необходимость в средстве сканирования для определения возможности восстановления файла. На основе ручного алгоритма был составлен алгоритм поиска удаленных файлов в системе NTFS представленный на рисунке 1.

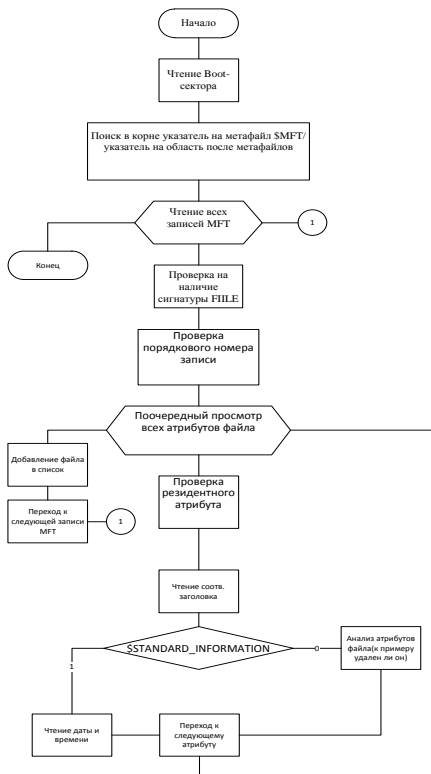


Рисунок 1 – Алгоритм поиска удаленных данных

Вывод. В результате проделанной работы разработан алгоритм поиска удаленных файлов в файловой системе NTFS пригодных для дальнейшего восстановления. Файлы могут быть любых форматов и размеров.

Список литературы

1. <https://ru.wikipedia.org/wiki/NTFS> – свободная энциклопедия (NTFS)
2. https://hetmanrecovery.com/ru/recovery_news/ntfs-file-system-analysis.htm – Анализ структуры данных файловой системы NTFS Автор-*Michael Miroshnichenko*
3. <http://samag.ru/archive/article/414> – Восстановление NTFS – undelete своими руками Автор Крис Касперски.

УДК 004.93'1

МЕТОДЫ ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЯ ПРИ ПРОЦЕДУРЕ ДИНАМИЧЕСКОГО РАСПОЗНАВАНИЯ ЛИЦ

Р. В. Лычагин, А. Я. Аноприенко, С. В. Иваница

Донецкий национальный технический университет, г. Донецк,
кафедра компьютерной инженерии
roma.l.work@gmail.com

Аннотация

Лычагин Р. В., Аноприенко А. Я., Иваница С. В. Анализ существующих методов выравнивания изображения при решении проблемы распознавания лиц. Выполнен анализ основных методов преобразования (выравнивания) изображений. Определены условия, в которых они могут быть применены. Также выведены наиболее полезные из них методы.

***Ключевые слова:** нормализация, выравнивание, гистограмма, фильтр Гауса, оператор Лапласа, масштабирование, идентификация, метод Виолы-Джонсона, признаки, изображение, кадр.*

Постановка проблемы. Интерес к процессам идентификации и аутентификации личности в современном мире постоянно повышается благодаря возрастающей практической потребности в их использовании во все большем количестве применений: от систем охраны и верификации банковских карточек до криминалистической экспертизы и идентификации преступников при рецидиве. Поскольку одним из основных и однозначно различимых критериев идентификации личности является человеческое лицо, то в данном случае речь идет, фактически, о процедуре динамического распознавания лиц.

Продуктивность динамического распознавания основывается на получении, в конечном счете, качественных статических изображений (кадров). При этом каждый варьирующийся параметр кадра усредняется, выравнивается, приводится к общему виду и на основе преобразованного изображения выполняется глубокий анализ с целью идентификации личности.

Цель данной работы заключается в исследовании и вычленении наиболее значимого способа преобразования изображения, а именно выравнивания изображения по отношению к изображенному на нем лицу, что позволяет реализовать методы распознавания лиц путем нормализации кадра.

Общий алгоритм распознавания. На рис. 1 представлен фрагмент общей схемы обработки изображения в разрабатываемой системе распознавания лиц. На схеме выделено 2 блока, где выполняется нормализация изображения.

Работа данного фрагмента заключается в следующем. Камера делает захват кадра. Программный компонент системы распознавания переводит в понятный для текущей среды кадр, который может быть получен как путем захвата видеопотока (кадр или несколько кадров видеокамеры), так и в виде графического файла на локальном физическом носителе.

На полученном кадре необходимо выполнить определённые модификации с целью нормализации кадра. Подобная нормализация может быть полезна при неоднородности внешних условий, а также – при неоднородности самих камер.

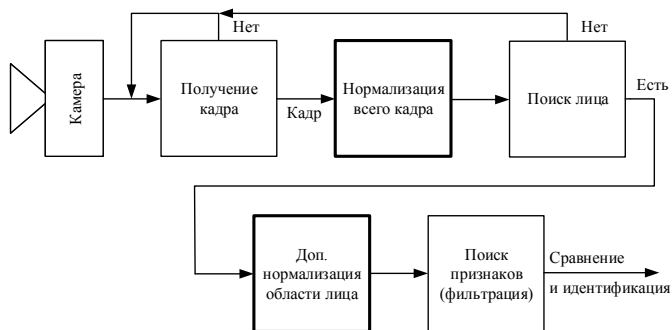


Рисунок 1 – Структурная схема последовательно используемых процедур системы распознавания лиц

Так как конечной целью является распознавание лиц, то необходимо определить все лица на кадре (при множественном распознавании), чтобы уменьшить рабочую область для анализа и минимизировать влияние внешних факторов, например, неоднородного фона. Обнаружение лиц на кадре осуществляется с помощью метода Виолы-Джонсона.

На найденной области проводятся дополнительные нормализации с целью облегчить сравнения приводя объект к некоторой общей форме, например, поворот и масштабирование лица. Эти преобразования возможно провести только при обнаружении лица, поэтому в схеме предусмотрено дублирование блока нормализации кадра.

После последней нормализации можно приступить к вычислению признаков, которые в дальнейшем будут использованы в сравнении объектов.

Предобработка изображения

Предобработка изображения направлена, прежде всего, на исправление труднораспознаваемых кадров и служит повышением стабильности работы программы в плохих условиях ведения видеонаблюдения. Данная процедура весьма важна в подобных системах. Все рассмотренные методы в данном разделе могут не только улучшить кадр, но и испортить его, поэтому

необходимо использовались эти методы с тщательным подбором набора параметров.

Простейшее преобразование изображения сводится к оптимизации яркости и контрастности [1]. Применяя формулу (1) к каждому пикселю изображения можно повысить или понизить показатели яркости и контрастности.

$$g(i, j) = \alpha \cdot f(i, j) + \beta, \quad (1)$$

где α – параметр контрастности [0, 3], значение по умолчанию 1;

β – параметр яркости [-100, 100], значение по умолчанию 0.

Для подавления шумов на изображениях используется фильтр Гаусса [2], являющийся выражением (2). Фильтр размывает изображение и тем самым устраняет лишние помехи. Данный фильтр не нашел активного применения на практике, но тем не менее был реализован в целях дополнительного тестирования устойчивости метода.

$$G_0(x, y) = A e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}}, \quad (2)$$

где μ – значение (пик); σ – дельта (каждого значения x и y)

В тех случаях, когда необходимо повышение резкости изображения используется фильтр на основе оператора Лапласа [3], являющийся выражением (3). Принцип использования фильтра заключается в том, что к изображению применяется оператором Лапласа и затем от исходного изображение вычитается полученная матрица. И этот механизм в сущности применения не нашел, но тем не менее остается в целях дополнительного тестирования устойчивости метода.

$$\text{Лаплас}(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3)$$

Гистограмма изображения [4], в данном случае, представляет собой набор значений пикселей и их частоту появления. При выравнивании гистограммы часто встречающиеся пиксели меняются в сторону реже встречающихся. Таким образом, в первую очередь выравнивается яркость и повышается контрастность изображения. Это компенсирует недостатки изображения, связанные, например, с условиями плохого освещения, а также позволяет различать больше деталей. Данный метод наиболее эффективно выравнивает кадр и повышает эффективность используемого после него метода Виолы-Джонсона по поиску лиц в кадре (изображении). Результаты приведённых методов продемонстрированы на рис. 2.

Дополнительная нормализация лица. Первая предобработка выполняется в первую очередь для выполнения последующего анализа кадра, поиска лица. И нет возможности на этом этапе выполнять все возможные изменения кадра. Но тем не менее перед вычислением признаков с известной позицией лица можно привести кадр к определенным условиям сравнения.

Результатом нахождения лица является прямоугольник, указывающий на расположение лица. Полученная фигура может иметь различные параметры

ширины и высоты. Это связано с различными размерами головы, расстоянием от камеры и возможными неточностями используемого метода Виолы-Джонсона. Но не смотря на все эти проблемы на этапе сравнения признаков желательно иметь 2 объекта в относительно одинаковых условиях. В связи с этим выполняется масштабирование данной области, приведение ее к одному размеру (200x200 пикселей). Масштабирование или изменение размера кадра — это частный случай геометрического преобразования [5] описанной формулой (4). Функция изменения размеров решает 2 проблемы экстраполяции несуществующих пикселей и интерполяции значений пикселей.

$$dst(x, y) = src(f_x(x, y), f_y(x, y)) \quad (4)$$



Рисунок 2 – Пример результатов обработки изображения

(*a* – исходное изображение, *б* – преобразование контраста и яркости, *в* – фильтр Гауса, *г* – аилтер на основе оператора Лапласа, *д* – выравнивание гистограмм)

Однако данный подход может и ухудшить качество кадра при условии, если область лица была изначально меньше чем та, к которой ее необходимо привести перед распознаванием. Но такой подход значительно упрощает сравнение полученных признаков. Также упрощается подбор параметров фильтра Габора, делая вычисление признаков более стандартизированным.

Есть и вторая проблема масштабирования – уменьшение количества потенциальных признаков, приводящее к уменьшению числа лиц, входящих в базу данных.

На выделенном участке лица можно также провести повторную нормализацию гистограммы, так как количество пикселей изменилось, во-первых, из-за выполненного масштабирования, а во-вторых вследствие вырезания участка с лицом из всего кадра.

Горизонтальное выравнивание. Метод Виолы-Джонсона весьма гибок и позволяет находить лица в различных вариациях, в различных углах наклона как головы, так и камеры. В результате на изображении лицо будет повернуто набор, что создает дополнительные проблемы в процессе сравнения. Один из способов это на этапе сравнения делать допущение на возможный угол наклона, но это сильно усложнит и затормозит весь процесс, а также уменьшит точность метода.

В связи с этими проблемами целесообразно выполнять определенный разворот кадра, с целью его горизонтального выравнивания. Для выполнения

такого разворота необходимо иметь некоторую угловую константу, полученную экспериментальным путем. Если взять, например, контуры губ, то из-за различного выражения лица, в результате мимических трансформаций, они не являются устойчивым параметром. И форма носа является недостаточным параметром, характеризующая угол наклона лица. Поэтому наиболее подходящим объектом в данном методе являются глаза. Таким образом, выполнив поиск глаз, например, методом Виолы-Джонсона, по найденным позициям их расположения можно вычислить угол наклона лица путем проведения параллели к линии пересечения центров глаз. По окончании расчетов выполняется поворот лица на полученный угол.

Выводы. В данной работе были рассмотрены основные особенности анализа и обработки изображения в ходе распознавания лица. Были определены условия, при которых наиболее эффективно будет применена нормализация кадров. В ходе анализа методов выравнивания были рассмотрены основные варианты нормализации изображения и выбран наиболее эффективный метод – метод выравнивания гистограмм, который делает кадр более разборчивым в условиях плохого освещения, а также способствует уменьшению разницы в яркостях на отдельных участках кадра. Был также предложен метод горизонтального выравнивания с использованием дополнительного сканирования лица методом Виолы-Джонсона. Данный метод находится на стадии дальнейшей разработки и требует дополнительных настроек.

Список литературы

1. Изменение контраста и яркости изображения [электронный ресурс] [сайт]. [2017]. URL: http://docs.opencv.org/2.4/doc/tutorials/core/basic_linear_transform/basic_linear_transform.html
2. Сглаживание изображения [электронный ресурс] [сайт]. [2017]. URL: http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html
3. Оператор Лапласа [электронный ресурс] [сайт]. [2017]. URL: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html
4. Высчитывание гистограммы [электронный ресурс] [сайт]. [2017]. URL: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_calculation/histogram_calculation.html
5. Геометрическое преобразование изображения [электронный ресурс] [сайт]. [2017]. URL: http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html

УДК 004.92

ИССЛЕДОВАНИЕ СТРУКТУРЫ ГРАФИЧЕСКОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ НА ПРИМЕРЕ ИГРЫ

А.А. Лада, Р.В. Мальчева

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
allada96@mail.ru

Аннотация

Лада А.А., Мальчева Р.В. Исследование структуры графической обучающей системы на примере игры. Выполнен анализ образовательных задач и определены основные подходы, способствующие более качественному восприятию и усвоению выдаваемого учебного материала.

***Ключевые слова:** графические обучающие системы, система контроля, система самопроверки.*

Постановка проблемы. Использование графического материала повышает наглядность УМ и тем самым ускоряет его восприятие.

При создании и внедрении в состав ГОС графики следует помнить, что:

- нежелательно, когда в тексте рябит от подвижных и ярких картинок, которые ничего не дают для понимания УМ, а только отвлекают внимание студентов;

- доступ к УМ может происходить при низких скоростях сетевых каналов, малой пропускной способности, поэтому значительное внимание стоит уделить оптимизации графического материала и выбора рационального формата его представления.

Цель статьи – провести анализ образовательных задач и определить основные подходы, способствующие более качественному восприятию и усвоению выдаваемого учебного материала.

Организация графики и анимации. Использование графического материала повышает наглядность УМ и тем самым ускоряет его восприятие.

При создании и внедрении в состав ГОС графики следует помнить, что:

- нежелательно, когда в тексте рябит от подвижных и ярких картинок, которые ничего не дают для понимания УМ, а только отвлекают внимание студентов;

- доступ к УМ может происходить при низких скоростях сетевых каналов, малой пропускной способности, поэтому значительное внимание стоит уделить оптимизации графического материала и выбора рационального формата его представления.

Для создания и редактирования иллюстраций могут быть использованы различные графические редакторы, начиная от программы MSPaint, что идет в

поставке с операционной системой Windows, и заканчивая последними графическими пакетами. В частности для редактирования и оптимизации иллюстративного материала используется Adobe Photoshop CS (команда Save for Web позволяет уменьшить цветопередачу файла без видимых потерь). Как форматы представления графики используется формат GIF, как наиболее эффективный для изображений с простыми линиями и областями чистого цвета (для иллюстрирования УМ) и формат JPEG для иллюстраций с плавными переходами тонов (для оформления интерфейса системы).

Для создания анимационных роликов используется пакет Macromedia Flash 8, который позволяет создавать разнообразные анимации и оптимизировать их качество. Для сравнения использовалась программа Microsoft GIF Animator, однако анимационные файлы, созданные с помощью этой программы имели гораздо больший объем и менее гибкий интерфейс использования. Учитывая, что разрабатываемая учебная система все-таки является продуктом учебного назначения, то из всех возможностей Flash используется "порционная" подача текстовой информации (эффект "электронного лектора") [1].

Создание видеоматериалов. Как видеоматериалы выступают инструкции по использованию предлагаемой системы. В данном случае для реализации поставленной задачи подходит программный продукт CamStudio 2.1. Критерием выбора именно этого ПО стала простота в использовании, доступный интерфейс и бесплатное распространение.

Организация системы самопроверки. Данная система представляет собой набор тестовых заданий с проверкой пользовательской ответа. Для реализации системы лучше всего использовать язык сценариев Javascript, как, встроенный в состав html страниц язык программирования для увеличения возможностей взаимодействия с пользователями. Для написания кода на JavaScript подходит любой текстовый редактор. Данный язык выбран из-за его простоты, распространенность и наличие необходимых средств для реализации поставленных задач.

Организация системы контроля. Предполагается реализация системы контроля как внешней программы, запуск которой выполняется из оболочки системы. Для реализации системы можно выбрать среду C ++ Borland Builder 6.0 по следующим причинам:

- языком написания программы является C ++, которая изучается и используется в течение нескольких лет обучения в университете;
- наличие в библиотеке объектов для реализации обмена информацией между клиентом и сервером;
- возможность организации программного доступа к базам данных и т.д [2].

При замене вопроса (кнопка "REPLACE") из базы вопросов будет выбрано задание по теме, выбранной при старте.

Количество допустимых пропусков определяется номером раунда. В игре всего предусмотрено 5 раундов. На первом раунде игрок должен ответить на один вопрос и может пропустить четыре вопроса. На втором раунде обязательно нужно ответить на два вопроса, разрешается пропустить три задания и т.д.

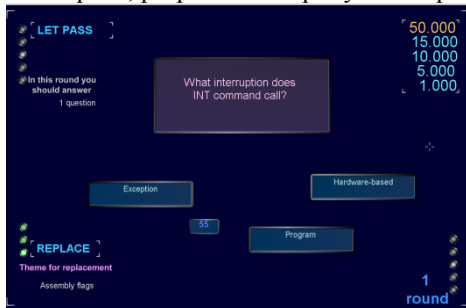


Рисунок 1 – Главное окно игры

При прохождении каждого раунда игроку начисляется определенная сумма. Участник может забрать эту сумму в процессе игры, нажав кнопку "MONEY".

Если участник успешно проходит все раунды, ему сообщается о максимальном выигрыше. При этом он может продолжить игру при нажатии кнопки "PLAY", имея в наличии постоянную сумму, выигранную на предыдущем этапе (рис. 2).

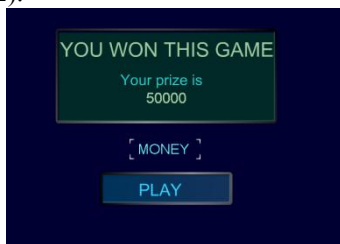


Рисунок 2 – Главное окно игры

Игра заканчивается без выигрыша при первой неправильном ответе.

Выводы. провести анализ образовательных задач и определить основные подходы, способствующие более качественному восприятию и усвоению выдаваемого учебного материала. Проанализирована структура графической обучающей системы на примере игры

Список литературы

1. Macromedia [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2012]. URL: <https://ru.wikipedia.org/wiki/Macromedia>
2. C ++ Borland Builder [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2012]. URL: https://ru.wikipedia.org/wiki/C%2B%2B_Builder

УДК 004.92

РАЗРАБОТКА ОНЛАЙН СИСТЕМЫ ТЕСТИРОВАНИЯ ЗНАНИЙ В ОБЛАСТИ FPGA ТЕХНОЛОГИЙ

А.С. Марченко, Ю.Е. Зинченко

Донецкий национальный технический университет
кафедра компьютерной инженерии
sashamarhenko@gmail.com

Аннотация

Марченко А.С., Зинченко Ю.Е. Разработка онлайн системы тестирования знаний в области FPGA-технологий. Выполнен анализ существующих разработок в сфере дистанционного оценивания знаний. Изучены аспекты функционирования подобных систем. Предложена модель для реализации онлайн-системы для проверки знаний.

Ключевые слова: *FPGA-технологии, оценивание знаний, система оценивания.*

Постановка проблемы. В процессе изучения предметов в современное время студенты часто сталкиваются с недостаточным качеством знаний из-за неполного погружения в материал. Решить проблему можно за счёт регулярного проведения тестирования, но процесс обязательно должен быть автоматизирован, поскольку преподавателю будет крайне сложно проверять огромный поток работ регулярно.

Реализации таких систем уже уделялось внимание в различных работах [1-3], однако авторам не удалось создать готовый продукт, который отвечал бы современным требованиям и мог бы быть внедрён в учебный процесс. Результаты ограничивались отдельными исследованиями и наработками в сфере безопасности систем удалённого оценивания знаний [2-3] или созданием локальной системы тестирования знаний с ограниченным объёмом вопросов и невозможностью расширения базы. В данной статье описывается вариант структуры онлайн-системы проверки знаний в сфере FPGA-технологий и возможный механизм её реализации.

Пользователи на данный момент могут выбрать огромный ассортимент курсов на разных образовательных площадках в сети интернет, на которых они получают и материалы по теме, и тесты, позволяющие закрепить усвоение вопроса.

Цель статьи – описать основные принципы функционирования системы, позволяющей проводить тестирование знаний студентов, а также создать алгоритм тестирования.

Организация системы онлайн-тестирования. Интернет-системы для проверки знаний получили широкое распространение. В их числе имеются разработки, предназначенные для тестирования знаний по языкам описания (программирования) аппаратуры, среди которых наибольшее распространение получили VHDL и VERILOG. Анализ этих систем показал их общие недостатки: ограниченность типов и отсутствие возможности редактирования задаваемых вопросов, а также сложность адаптации к другим компонентам методического обеспечения и к учебному процессу в целом.

В данной статье предлагается онлайн-система тестирования знаний, названная ATP (Advanced Testing Package). Система ATP представляет собой WEB-сайт с соответствующим информационным и программным обеспечением. Краткая схема представлена на рисунке 1.

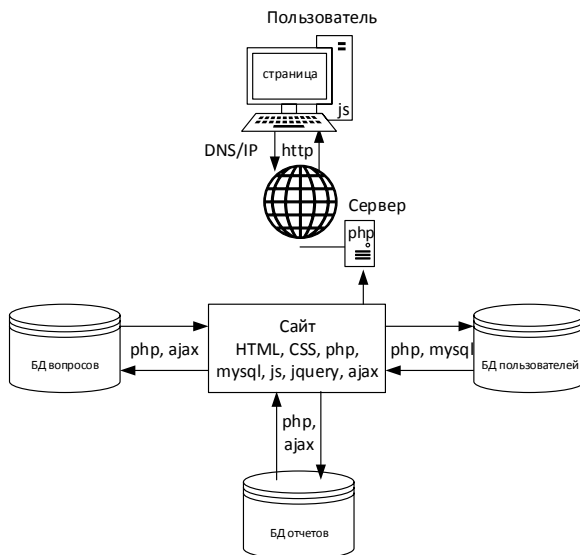


Рисунок 1 – Структура проекта

Более подробная структура сайта системы онлайн-тестирования приведена на рисунке 2.

Работу с системой пользователь начинает с главной Web-страницы, на которой предоставляется возможность выбора дальнейших действий: регистрация, вход в систему для прохождения теста, просмотр журнала или изменение настроек системы (последняя функция доступна только администратору системы).

Пользователь “администратор” запрограммирован в систему и не имеет возможности пройти тестирование. Данный пользователь попадает сразу на

страницу управления учетными записями и допусками пользователей "user" к тестам. Войдя в систему, пользователь "user" имеет возможность выбрать нужный ему тест и, если допуск для прохождения установлен, то он попадает непосредственно в зону тестирования. По окончании тестирования системой выдается отчет о прохождении тестирования, который фиксируется в базе данных системы. При оценке результата тестирования учитывается также время, потраченное пользователем и сложность каждого вопроса. Результаты тестирований может просмотреть любой пользователь (для этого не обязательна регистрация). Результаты можно просмотреть как в сжатой форме, так и в развернутом виде (по каждому вопросу в отдельности).

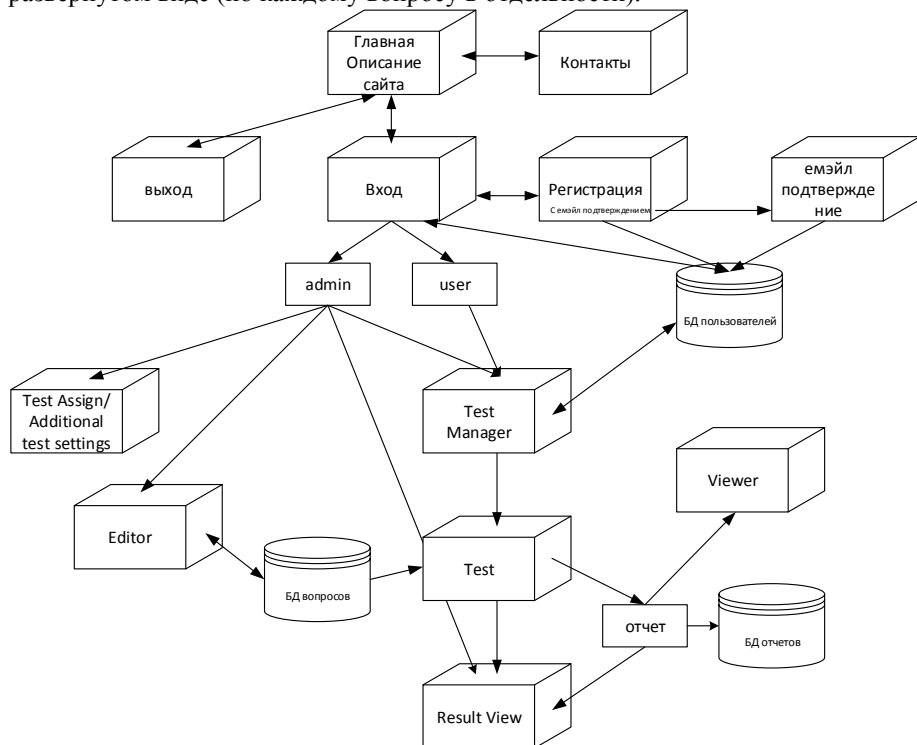


Рисунок 2 – Структура сайта системы тестирования

Структура сайта системы тестирования включает следующие основные компоненты:

1. Внешняя оболочка сайта – это визуальная навигационная система проекта, включающая в себя такие основные компоненты:

- главная страница, содержащая описание сайта;
- контакты - информация о разработчиках проекта;

- возможность авторизации и выход из профиля пользователя;
 - регистрация - добавление нового пользователя к проекту с учетом действующего e-mail.
2. СУБД тестов - система управления базами вопросов и ответов тестов:
- база тестов – содержит вопросы и ответы, укомплектованные в отдельные пакеты по дисциплинам;
 - редактор тестов (editor) – подсистема построения и редактирования баз тестов;
 - менеджер тестов (test manager) - система распределения тестов между пользователями (доступна администратору);
 - тест (test) - содержит определенное количество вопросов и учитывает дополнительно заданные параметры;
 - база отчетов - содержит отчеты с результатами пройденных тестов с возможностью подробного просмотра ответов, данных пользователем;
 - таблица результатов (result view) - информация по результатам пройденных тестов.
3. СУБД пользователя - система управления информацией обо всех пользователях сайта:
- admin - пользователь, отвечающий за администрирование всего сайта, а также за распределение, создание, назначение всех тестов, содержащихся в базе тестов, и контроля полученных результатов.
 - user - пользователь, который может пройти контроль своих знаний.

Алгоритм тестирования. Тестирование имеет следующие основные действия: ответить на вопрос, пропустить вопрос, сдать тест. При чем пропустить вопрос может не отображаться если это было задано в настройках теста администратором.

Основные события в тестировании представлены на рисунке 3. Видно, что в начале теста выводится вопрос. После учитывается нажатие кнопки сдать тест – если определяется нажатие кнопки, все действия останавливаются для предотвращения нажатий после сдачи теста. Если тест не сдан, то следующее возможное действие – это пропуск вопроса. Действие сразу переводит пользователя на функцию перехода на следующий вопрос.

Если же пропуска не было, то остается событие ответа, которое является ключевым в проведении теста. Оно выполняет сбор данных и отправляет их на сервер, где происходит проверка ответа и сохранение действий пользователя. После чего выполняется переход на следующий вопрос и алгоритм повторяется, начиная с вывода текущего вопроса.

На приведенной блок схеме не отображена работа таймера, потому что он запущен параллельно в отдельном потоке и никак не зависит от работы данных элементов. Также на сайте реализованы некоторые алгоритмы защиты от

жульничества. Так, например, нельзя перезапустить тест вернувшись на страницу выбора теста. Правильный ответ в исходном коде «подсмотреть также нельзя» – все переданные ответы сравниваются с правильным на сервере, поэтому клиенту не передаются данные с верными ответами.

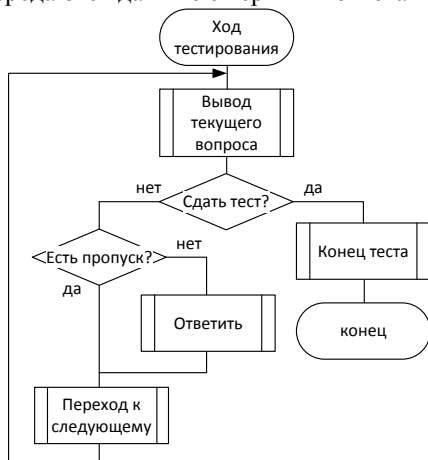


Рисунок 3 – Основные события хода тестирования

Выводы. Названы недостатки существующих систем тестирования знаний в требуемой сфере. Предложена модель полноценной автономной системы тестирования знаний, которая позволяет легко контролировать результаты каждого студента и назначать ему необходимые для более качественного изучения предмета тесты. Предложен алгоритм проведения тестирования и составные элементы онлайн-системы тестирования. В результате разработки проекта по созданной модели должен получиться сайт, который может использоваться для дистанционного обучения основам FPGA-технологии с возможностью проверки знаний всех обучаемых по любой теме.

Список литературы

1. Зинченко Ю. Е., Корченко А. А., Масюк А. Л. ADVANCED TESTING PACKAGE 4.8 – Методическое пособие по применению системы автоматизированного интерактивного контроля знаний – 40с.
2. Политика защиты систем дистанционного обучения [Электронный ресурс] // Дейкун О.В., Губенко Н.Е. URL: <http://masters.donntu.org/2009/fvti/daikun/library/tez3.htm>
3. Защита информации при разработке обучающих систем [Электронный ресурс] //Черепкова Е.В. URL: <http://masters.donntu.org/2006/fvti/cherepkova/diss/index.htm>

УДК 681.3

МОДЕРНИЗАЦИЯ АЛГОРИТМОВ ОПРЕДЕЛЕНИЯ РУК ПРИ ПОМОЩИ KINECT И LEAP MOTION

Лупашевский В.А., Аноприенко А.Я.

Донецкий национальный технический университет
кафедра компьютерной инженерии

E-mail: vlad@lupashevskiy.com, anoprien@gmail.com

Аннотация

Лупашевский В.А., Аноприенко А.Я. Модернизация алгоритмов определения рук при помощи Kinect и Leap Motion. В данной статье представлено описание модернизированного алгоритма для определения рук при помощи Microsoft Kinect в системах, где используется комбинация Microsoft Kinect и Leap Motion. В результате удалось повысить частоту обработки информации из Microsoft Kinect с 10 до 40 кадров в секунду.

Ключевые слова: *Kinect и Leap Motion, обработка информации.*

Актуальность

В настоящее время, благодаря развитию технологий в области компьютерного зрения, актуализировался, в числе прочих, вопрос достаточного уверенного распознавания рук человека и жестов, в том числе при помощи комбинации различных методов. Примером актуального использования таких технологий является контроль корректности выполняемых операций на производствах, где сборка агрегатов реализуется по-прежнему руками человека. Не стоит также забывать и об особой популярности данной темы в игровой индустрии, и в приложениях виртуальной реальности.

В 2009 году компанией Microsoft был представлен сенсор Microsoft Kinect, способный распознавать движения тела человека на расстоянии от 0.5 до 2 м. Позднее, в 2012 году, компания, Leap Motion выпустила сенсор, который достаточно точно определяет положение отслеживает движения рук и отдельных пальцев в пределах прилегающего пространства радиусом порядка 60 см. Ввиду наличия у обоих сенсоров своих специфических достоинств и недостатков представляется целесообразным исследовать возможности комбинирования данных сенсоров для получения более точных результатов отслеживания рук путем компенсации недостатков одного сенсора достоинствами второго.

Цель исследования

Целью данного исследования является улучшение показателей качества, а также ускорение процесса определения рук при помощи использования сенсора

Microsoft Kinect путем модернизации существующих разработок в области использования комбинации Microsoft Kinect и Leap Motion для определения рук в реальном времени. Помимо этого, стоит задача экспериментальным путем доказать возможность эффективного использования языка программирования C# для реализации данных алгоритмов.

Обзор используемых устройств

Современные модификации Microsoft Kinect представляют собой сенсор, который способен определять движения тела человека на расстоянии от одного до двух и более метров. Информация при этом представляется трехмерными координатами определенных точек скелета человека. Помимо этого имеется возможность получить цветное и инфракрасное изображения, а также – данные с камеры-глубины. Что касается непосредственно кистей рук, то Microsoft Kinect предоставляет информацию только о центре ладони (сенсор второй версии способен определять также и положение большого пальца) [2], что является недостаточным для точного определения движения рук.

Leap Motion, в свою очередь, предназначен для определения только рук на небольшом расстоянии, однако является более информативным, так как определяет координаты не только центра ладони, но также и координаты всех пальцев руки.

Исследование и основные результаты

За основу было взято исследование [1]. Алгоритм метода определения рук с использованием комбинации Leap Motion и Kinect можно разделить на 8 этапов (рис. 1).

В вышеупомянутом исследовании для работы с изображениями применялась библиотека OpenCV 2. Для уменьшения времени обработки данных, авторами текущей статьи было принято решение использовать оптимизированную и улучшенную третью версию данной библиотеки [3, 4].



Рисунок 1 – Порядок действий для определения рук

Для дальнейшей обработки данных, получаемых с сенсора Microsoft Kinect, необходимо их преобразовать в понятный для OpenCV формат, которым является матрица. В качестве метода для уменьшения времени преобразования используются средства языка C# для работы с неуправляемой памятью, а именно класс Marshal и его метод Copy.

Для улучшения качества определения рук на изображении необходимо произвести калибровку – определить пороговое значение глубины нахождения рук. Это позволит убрать лишние фрагменты изображения используя информацию камеры-глубины, т.е. если глубина пикселя превышает пороговое значение найденное в результате калибровки, то данный пиксель закрашивается черным цветом и не будет оказывать воздействия на определение контура руки.

Калибровка выполняется однажды, при запуске программы. Первые 5 секунд ведется обратный отсчет перед началом калибровки. Последующие 5 секунд выполняется сам процесс. Он представляет собой сбор данных о удалении центра ладони от сенсора Kinect (координата Z). По истечению времени сбора данных производится кластеризация данных с использованием плотностного алгоритма кластеризации пространственных данных с учетом шума – DBSCAN (Density Based Spatial Clustering of Applications with Noise) [5]. Данный алгоритм позволяет разделить места скоплений точек на группы. В нашем случае данными группами будут являться положения центров левой и правой рук. Помимо этого, данный алгоритм способен определять шум, что делает его применение полезным ввиду того, что данные приходящие с сенсора Kinect являются зашумленными.

Было проведено также улучшение метода маскирования изображения по глубине. В оригинальном методе данные в матрице обнулялись каждый проход цикла, в случае неудовлетворения условиям. Так как данный алгоритм отсекает области изображения, которые представляют информацию о руках, большая часть изображения должна быть окрашена в черный цвет. Было выяснено, что при окрашивании только одного пикселя затрачивается больше времени, чем при закрашивании одновременно всех пикселей. Исходя из этого было принято решение закрашивать все пиксели – заполнять матрицу нулями. Для этого была использована функция RtlZeroMemory из динамической библиотеки kernel32.dll, которая принимает два параметра: указатель на начало области памяти и размер области памяти, которую нужно заполнить нулями.

Вдобавок, еще одним предложенным решением является проверка принадлежности пикселя к скелету человека в дополнение к проверке значения его удаления в сравнении с пороговым значением, полученным в результате калибровки. Эта методика позволяет отсеять посторонние предметы, находящиеся примерно на одном удалении с руками, что обеспечивает значительное уменьшение времени обработки, так как в маскированное изображение попадут только руки.

Этап определения рук заключается в нахождении контура руки на полученном ранее маскированном изображении. Для определения контура используются методы нахождения руки по цвету кожи. В качестве цветовой модели, которая способна описать точным образом цвет кожи используется модель HSV. После нахождения контура, исходные изображения руки обрабатываются при помощи алгоритма построения выпуклой оболочки на плоскости. Для этого используются функции OpenCV: ConvexHull и ConvexityDefects. Пример выполнения этих функций показан на рис. 2. Красными точками показаны координаты оболочки, а зелеными – дефекты.

Для нахождения центра ладони находится вписанная в контур ладони окружность, центр которой будет представлять центр ладони. На рис. 3 синей линией показана вписанная в руку окружность, а красной точкой показан ее центр.

В ходе работы этап определения рук на маскированном изображении был также подвержен модернизации, суть которой заключается в предварительной и последующей обработке данных до и после вызова функции нахождения контура. После нахождения маски, производятся операции эрозии и расширения, которые сглаживают конур маски, а также уменьшают количество шумов в ней.

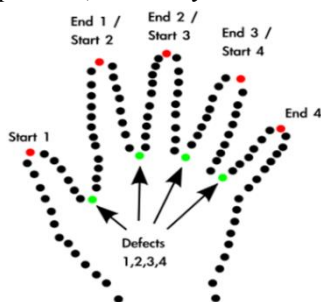


Рисунок 2 – Результаты работы функций ConvexHull и ConvexityDefects

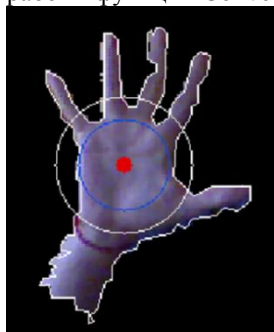


Рисунок 3 – Результат работы алгоритма определения центра руки

Смысл постобработки заключается в уменьшении количества точек в сегментах путем использования функции ApproxPolyDP. Итогом данных операций является более сглаженный контур с меньшим количеством точек, что приводит к уменьшению времени дальнейшей обработки. Результат работы данного этапа показан на рис.4.

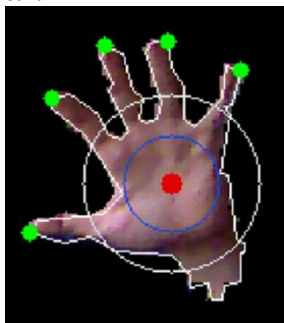


Рисунок 4 – Определенные позиции кончиков пальцев (зеленые точки) и центр руки (красная точка)

Выводы

В ходе модернизации существующего метода, который входит в систему определения рук на базе комбинированного использования Motion и Microsoft Kinect, а именно в части обработки данных последнего сенсора, была увеличена частота обработки информации с 10-15 до 30-40 кадров в секунду, что означает увеличение производительности в 2-3 раза.

Использование языка программирования C# с версией .NET Framework 4.5 для реализации данного алгоритма позволило обеспечить обработку изображений в режиме реального времени.

Список литературы

1. Natural Human-Computer Interaction with Kinect and Leap Motion [Электронный ресурс].– URL: https://cs.anu.edu.au/courses/csprojects/14S1/Final_presentations/Presentations%20by%20student/Xietong_Lu_Final.pdf
2. Kinect for Windows SDK. [Электронный ресурс].– URL: <https://msdn.microsoft.com/library/dn799271.aspx>
3. Emgu CV: OpenCV in .NET. [Электронный ресурс].– URL: http://www.emgu.com/wiki/index.php/Main_Page
4. OpenCV library. [Электронный ресурс].– URL: <http://opencv.org/>
5. Плотностный алгоритм кластеризации пространственных данных с присутствием шума — DBSCAN. [Электронный ресурс].– URL: <https://habrahabr.ru/post/143151/>

УДК 004.42

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ МЕТОДОВ ОПТИМИЗАЦИИ БАЗ ДАННЫХ В РАМКАХ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПРИЕМНОЙ КОМИССИИ ДОННТУ

Дерксен Е.А., Завадская Т.В., Меренкова Л.Л.
ГОУВПО "Донецкий национальный технический университет"
кафедра компьютерной инженерии
E-mail: derksen.evgenij@yandex.ua

Аннотация

Дерксен Е.А., Завадская Т.В., Меренкова Л.Л. Исследование эффективности методов оптимизации баз данных в рамках автоматизированной системы приемной комиссии ДонНТУ. Проведен анализ существующих методов оптимизации реляционных баз данных. Определена эффективность каждого метода при использовании в рамках автоматизированной информационной системы приемной комиссии ДонНТУ.

Ключевые слова: база данных, оптимизация, денормализация, индекс.

Постановка проблемы. В разнообразных информационных системах в качестве основного хранилища данных используются реляционные базы данных (БД). Существуют методы оптимизации и повышения производительности таких БД, однако их эффективность в конкретной задаче заранее неизвестна.

Цель статьи – провести анализ существующих методов оптимизации структуры баз данных и определить их эффективность при использовании в рамках автоматизированной информационной системы приемной комиссии ДонНТУ.

Анализ существующих методов оптимизации. Для системы управления базами данных (СУБД) Microsoft SQL Server, используемой в качестве сервера БД в автоматизированной системе (АС) приемной комиссии ДонНТУ, существуют следующие методы оптимизации структуры БД [1]:

- 1) Индексирование.
- 2) Денормализация.
- 3) Использование столбцов постоянной и переменной длины.

Рассмотрим подробнее каждый из этих способов.

Индексирование. Индекс – объект базы данных, создаваемый с целью повышения производительности поиска данных. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и

их поиск по заданному критерию путём последовательного просмотра таблицы строка за строкой может занимать много времени. Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет искать строки, удовлетворяющие критерию поиска. Ускорение работы с использованием индексов достигается в первую очередь за счёт того, что индекс имеет структуру, оптимизированную под поиск – в случае СУБД Microsoft SQL Server – сбалансированного дерева.

Как и большинство других методов оптимизации, данный не лишен недостатков. Например, увеличение количества индексов может ускорить выполнение запросов, содержащих команду SELECT. Однако команды управления данными (INSERT, UPDATE и DELETE) будут выполняться при этом значительно медленнее, поскольку при каждой операции необходимо будет изменять большее число индексов. Поэтому в тех случаях, когда большая часть запросов содержит инструкцию SELECT, создание дополнительных индексов может повысить скорость обработки запросов. Однако, если приложение выполняет большое количество операций по управлению данными, количество индексов не следует увеличивать без необходимости.

Денормализация – намеренное приведение структуры базы данных в состояние, не соответствующее критериям нормализации, обычно проводимое с целью ускорения операций чтения за счет добавления избыточных данных.

Нормализованная база данных предотвращает появление функциональных зависимостей в данных, что упрощает обновление базы данных и делает его более эффективным. Однако при выполнении запросов к базе данных для получения результирующих сведений может потребоваться объединение большого количества таблиц. С увеличением числа объединяемых таблиц время выполнения запроса значительно возрастает. Поэтому в некоторых случаях не рекомендуется использовать нормализованные базы данных. Как правило, в качестве компромиссного варианта используется база данных с определенным уровнем денормализации. Это несколько повышает сложность процесса обновления, но позволяет уменьшить число таблиц, которые необходимо объединять.

Использование столбцов постоянной и переменной длины. При проектировании таблиц необходимо знать достоинства и недостатки применения столбцов постоянной и переменной длины. Использование столбцов переменной длины уменьшает размер базы данных, поскольку они занимают столько места, сколько необходимо для размещения хранящихся в них данных. Столбцы постоянной длины всегда занимают максимально возможный объем места, указанный в схеме (даже если данные в соответствующем столбце отсутствуют). Недостатком столбцов переменной длины является то, что некоторые операции выполняются для них менее эффективно, чем для столбцов постоянной длины.

Например, если столбец переменной длины изначально имел небольшой размер, а после выполнения операции UPDATE его размер значительно вырос, может потребоваться перемещение соответствующей записи. Кроме того, частые обновления постепенно приводят к повышению фрагментации страниц данных. Поэтому в тех случаях, когда размер содержащихся в столбце данных будет изменяться незначительно и когда часто выполняются обновления данных, рекомендуется использовать столбцы постоянной длины.

Исследование эффективности методов оптимизации. В рамках исследования АС приемной комиссии ДонНТУ были найдены следующие часто используемые, наиболее ресурсоемкие транзакции:

- 1) Выборка всех данных по всем абитуриентам.
- 2) Выборка всех данных по абитуриентам в рамках одного факультета, одной формы обучения, одного образовательного уровня.
- 3) Выборка всех данных по одному абитуриенту.
- 4) Обновление конкурсных баллов для всех абитуриентов.

На основе времени данных транзакций исследована эффективность методов оптимизации структуры базы данных. Результаты представлены в таблице 1.

Таблица 1 – Время выполнения транзакций

Транзакция	Время выполнения, мс				
	Исходн	Индексир	Денормализ	Поля стат. длины	Поля динам. длины
1	1950	1910	1811	1965	1759
2	7292	6667	7141	7342	6996
3	8	4	6	8	8
4	131	150	145	101	143

Выводы. На основе полученных результатов можно сделать выводы:

- для ускорения выборки данных по установленным условиям наиболее эффективным способом оптимизации является индексирование, однако индексирование замедляет операции обновления данных;
- для ускорения выборки больших объемов данных без условий наиболее эффективным способом является использование полей динамической длины.
- денормализация дает незначительный прирост производительности при выборке данных, но замедляет операции обновления.
- использование полей статической длины значительно ускоряет операции обновления данных для большого количества записей.

Список литературы

1. Методы оптимизации SQL Server 2005 // Microsoft MSDN [Электронный ресурс] – Режим доступа: [https://technet.microsoft.com/ru-ru/library/ms172984\(v=sql.90\).aspx](https://technet.microsoft.com/ru-ru/library/ms172984(v=sql.90).aspx)

УДК 004.5

ЭМУЛЯЦИЯ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ ДЛЯ УДАЛЕННОГО ДОСТУПА К FPGA-КОМПЛЕКСАМ

Азаров А.Б., Зубков Д.А., Константинов В.С., Зинченко Ю.Е., Коваль А.В.

Донецкий национальный технический университет
кафедра компьютерной инженерии
E-mail: azarovantony@gmail.com

Аннотация

Азаров А.Б., Зубков Д.А. Константинов В.С., Зинченко Ю.Е., Коваль А.В. Эмуляция периферийных устройств для удаленного доступа к FPGA-комплексам. Описаны методы организации дистанционного доступа периферийных устройств к отладочному FPGA-стенду, методы организации передачи, ввода и вывода информации. Также рассмотрена схема взаимодействия пользователя с FPGA-лабораторией и проблемы многопользовательского управления комплексом в целом.

***Ключевые слова:** дистанционный доступ, FPGA-комплекс, Web-интерфейс, эмуляторы периферийных устройств, COM-порт, Websocket.*

Введение

Одним из важных навыков, получаемых студентами на специальностях, связанных с компьютерной инженерией, являются навыки работы с вычислительными устройствами типа FPGA.

FPGA (Field Programmable Gate Arrays) широко применяются для моделирования микроэлектронных устройств и их детального исследования и отладки [1-4].

Для работы с FPGA необходима инфраструктура. К сожалению, существует ряд проблем, которые мешают студентам получать необходимый опыт в данной области. Главная проблема это высокая стоимость отладочных стендов. Платы являются достаточно дорогими и часто выходят новые, более функциональные и быстродействующие платы, поэтому лаборатории учебных заведений не могут обеспечить каждого студента доступом к платам в учебные часы. Также существует проблема с доставкой стендов в наш регион. Решением данной проблемы является создание удаленного доступа к FPGA лаборатории (FPGA-стендам) при помощи технологии Интернет. Данный ресурс поможет обеспечить студентам доступ к FPGA-стендам и будет способствовать их более углубленному изучению.

Данная статья посвящена описанию решения проблемы доступа к периферийным устройствам ввода/вывода а также кнопчным переключателям в условиях дистанционного доступа.

Постановка задачи

Вопрос обеспечения дистанционного доступа к FPGA-комплексам неоднократно поднимался в различных работах [5-7]. Авторы данных работ добились некоторого успеха в этой области, однако нигде даже не обсуждался вопрос имитации периферийных устройств, которые могут подключаться к FPGA-комплексам. Результаты ограничивались загрузкой прошивки в FPGA и в некоторых случаях управлением внешними переключателями. Попытки удаленного использования периферии предпринимались студентами ДонНТУ [8,9], но серьезных результатов достигнуто не было.

Данная статья описывает решение проблемы эмуляции периферийных устройств, а также использования LCD-монитора для обеспечения управления FPGA-комплексом в процессе тестирования проектов, ориентированных на реализацию на FPGA.

На данный момент пользователи, при помощи среды разработки Xilinx ISE Design Studio могут создавать «системы на кристалле (СнК)», добавляя в них периферийные устройства. Далее программа загружается в FPGA и отлаживается при помощи подключения реальных физических устройств непосредственно в лаборатории. В ядре СнК в FPGA-комплексах компании Xilinx располагаются стандартное IP-ядро микропроцессора Microblaze или микроконтроллера Picoblaze. Целью работы является возможность погружать, отлаживать и выполнять FPGA-проекты как удаленно, так и непосредственно на FPGA-комплексе, без внесения изменений в программный код проекта удаленного разработчика.

Организация удаленного доступа

Структуру системы удаленного доступа к FPGA-комплексу можно представить в виде совокупности трех основных частей: клиентская, серверная и собственно FPGA-комплекс. Структура системы представлена на рисунке 1.



Рисунок 1 Структура системы удаленного доступа к FPGA-комплексу

Процесс управления платой организован при помощи Web-интерфейса. Клиентская (визуальная) часть создается при помощи технологий HTML, CSS и JavaScript. Она состоит из главной страницы, которая должна презентовать

виртуальную FPGA-лабораторию, регистрации пользователей и самого кабинета виртуальной лаборатории. Доступ к виртуальному кабинету могут иметь только зарегистрированные пользователи с особыми статусом, которыми может наделить только администратор сайта. Интерфейс кабинета является максимально простым и легким для понимания. Все запросы к серверной части в виртуальном кабинете должны производиться без перезагрузки страницы, это возможно реализовать при помощи технологии AJAX (Asynchronous Javascript and XML). Также для работы с комплексом необходима видеотрансляция, которая будет отображать все изменения на комплексе в реальном времени.

В личном кабинете также присутствует простейший файловый менеджер, необходимый для удобной загрузки файлов на сервер. В нем также представлены все ранее загруженные файлы с возможностью их повторной загрузки в FPGA.

Лаборатория является многопользовательским ресурсом, однако обеспечение одновременного доступа к одному комплексу нескольких пользователей невозможно. Для того, чтобы во время работы с платой другие пользователи не могли загружать в нее свои проекты, необходимо обеспечивать контроль над сессией. В случае занятости комплекса, сервер запрещает к ней доступ для других пользователей до ее освобождения. Также при продолжительном бездействии пользователя плата освобождается автоматически.

Для организации удаленного управления и отладки подключена к серверу через COM-порт, USB-jtag для программирования стенда, также присутствует видеочамера, для отображения состояния стенда в реальном времени. Через интерфейс COM-порта производится обмен данными. Загрузка прошивки на плату организована с помощью batch-скрипта, который запускается по запросу пользователя. На FPGA-стенде все операции ввода/вывода должны перенаправляться на порт RS232. Для этого необходимо разработать эмуляторы периферийных устройств, которые должны принимать данные с порта RS232 и перенаправлять их на шину process local bus (PLB), а также ответ с шины передавать в порт RS232.

Эмуляция периферийного устройства

Эмуляция периферийных устройств должна обеспечивать постоянную связь пользователя с программным кодом, находящимся в плате. Исходя из этого, эмуляция периферии должна быть задействована во всех уровнях структуры виртуальной лаборатории. (см. рис. 2).

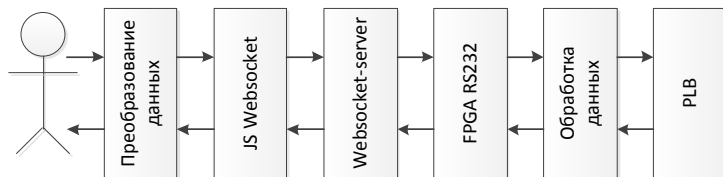


Рисунок 2 – Жизненный цикл одного действия эмулятора периферийного устройства

При эмуляции устройства ввода, из данных, которые вводит пользователь (нажатие клавиши или изменение положения мыши), необходимо сформировать пакеты данных, аналогичные тем, которые посылает физическое устройство. Данные пакеты погружаются в информационную часть пакета, структура которого представлена на рисунке 3. Эти пакеты посредством Websocket передаются на сервер. Сервер отправляет полученный пакет через COM-порт на стенд. Встроенный коммутатор, анализируя код устройства, определяет от какого устройства пришел пакет и отправляет его на соответствующий модуль эмулятора. Пользовательский код, находящийся в устройстве, принимает данные от модуля имитатора, как от физического устройства. Данные обрабатываются пользовательским кодом, который находится в стенде, и эмулятор формирует выходной пакет, структура которого аналогична входному пакету. Через COM-порт пакет передается обратно клиенту через сервер, где уже javascript-сценарий для конкретного периферийного устройства должен принять решения о последующих действиях.



Рисунок 3 – Структура входного/выходного пакета

Также, одной из особенностей использования периферийных устройств является их инициализация. При прохождении инициализации устройство последовательно посылает и принимает серии байтов, для того, чтобы оно было идентифицировано и были определены его параметры.

Заключение

В данной статье была в общем рассмотрена технология дистанционного доступа к FPGA-комплексам и имитация периферийных устройств. В частности для написания статьи в качестве периферийных устройств рассматривались клавиатура и мышь. В качестве направления дальнейшей работы можно рассмотреть дополнительно имитацию других устройств. Результаты работы будут использованы при проектировании системы дистанционного доступа к FPGA-комплексам лаборатории ДонНТУ, а также могут использоваться при

выполнении лабораторных работ, НИРС, дипломном проектировании и при выполнении магистерских работ.

Список литературы

1. Зинченко Ю., Калашников В., Хайдук С. и др. FPGA-технологии проектирования и диагностика компьютерных систем / Сборник научных трудов VI Межд. научн.-практ. конф. «Современные информационные технологии и ИТ-образование». - Москва: МГУ, 2011. - Т. 1. 787 С. 422-429.
2. Зинченко Ю., Дяченко О. и др. Новые технологии проектирования в ДонГТУ /Межд. науч.-техн. конф. ”Новые информационные технологии в САПР и АСУ”. Киев.- 2001.- С 15-19.
3. Медведев К.В., Зинченко Ю.Е. Навигационная система на базе FPGA-технологий с возможностью самотестирования // Сб. тр. VII международной научно-технической конференции студентов, аспирантов и молодых ученых «Информатика и компьютерные технологии» – 2011. – Донецк: ДонНТУ – 2011.
4. Волошин Д.Н., Зинченко Ю.Е. Разработка системы реального времени на базе Xilinx XC3S500E Spartan-3E FPGA // Сб. тр. VII межд. науч.-техн. конф. студентов, аспирантов и молодых ученых «Информатика и компьютерные технологии», 22-23 ноября 2011 г., Донецк: ДонНТУ. – 2011. Т. 1. – с. 44-47.
5. Rusen J., Kolberg S. Online FPGA laboratory for interactive digital design //International Conference on Engineering Education 2004 (ICEE2004). – 2004. – С. 16-21.
6. A Remote FPGA Laboratory for Digital Design Students. Авторы: Joao Soares, Jorge Lobo. Режим доступа: <http://ap.isr.uc.pt/archive/jlobo-REC2011.PDF>
7. Лаборатория Электронных Средств Обучения (ЛЭСО) СибГУТИ. Интернет ресурс. Режим доступа: <http://www.labfor.ru/online/fpga>
8. Зинченко Ю.Е., Зинченко Т.А., Асланов И.Г. Виртуальная FPGA-лаборатория с дистанционным доступом // III Всеукр. школа-семинар молодых ученых и студентов Современные компьютерные информационные технологии (АСИТ'2013). – Тернополь: ТНЕУ, 2013
9. Беседа Д.Г., Зинченко Ю.Е. Разработка метода эмуляции работы FPGA-платы с клавиатурой через порт PS/2 // Сб. тр. V Всеукр. науч.-техн. конф. студентов, аспирантов и молодых ученых «Информационные управляющие системы и компьютерный мониторинг (ИУС КМ-2014)». – Донецк: ДонНТУ, 2014 - т.1 - С. 157-162

УДК 004.72

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ SD-WAN СЕТЕЙ

Зимцев П.П., Коротач А.С., Чередникова О.Ю.,

Донецкий национальный технический университет

кафедра компьютерной инженерии

E-mail: akorotach@yandex.ru

Аннотация

Зимцев П.П., Коротач А.С., Чередникова О.Ю. Исследование возможностей SD-WAN сетей. Описаны возможности SD-WAN сетей, преимущества, концепция работы SD-WAN, возможный технический функционал, задачи, решаемые с помощью SD-WAN.

Ключевые слова: SD-WAN, сети, сетевое оборудование, разработка, каналы связи, сетевые технологии.

Введение

Одним из важных навыков, получаемых студентами на специальностях, связанных с компьютерной инженерией, являются навыки работы в сфере компьютерных сетей.

Компьютерная сеть (вычислительная сеть) – система, обеспечивающая обмен данными между вычислительными устройствами (компьютеры, серверы, маршрутизаторы и другое оборудование).

Для работы с компьютерными сетями необходимо разное оборудование, а также разные архитектурные решения, которые позволяли бы выполнять задачи разной сложности. В проектировании сетей все больше начинает набирать популярность новая технология SD-WAN. По заявлению разных производителей данная технология является отказоустойчивой и более надежной. По оценкам аналитиков, в мире растет потребность в новых и более совершенных подходах к построению сетей. По данным Gartner, к концу 2019 г. распространенность технологии SD-WAN в организациях должна вырасти до 30% (сейчас – менее 1%). По прогнозам IDC, мировой рынок SD-WAN вырастет с 225 млн. долл. в 2015 г. до более чем 6 млрд. долл. в 2020-м.

Следует заметить что данная технология является экономически выгодной, а архитектор ИТ-сети, как утверждается, сможет легко решать глобальные или локальные вопросы, например, перенаправить трафик всех видеоприложений по сетям с максимальной емкостью; предоставлять обновления программного обеспечения по интернет-каналам, а бизнес-трафик – по защищенным VPN-сетям. Раньше операторам сети приходилось вводить сотни и тысячи строк кода CLI (через интерфейс командной строки), чтобы изменить конфигурацию сети и приложений. Этот неэффективный способ был к тому же не застрахован от человеческих ошибок [1].

Возможности SD-WAN

Software Defined WAN (SD-WAN) – концепция на основе архитектуры SDN, цель которой предоставить новый подход к управлению корпоративной WAN-сетью.

Данная технология является новым решением в сфере построения компьютерных сетей, которая, как утверждают производители ведущих фирм как Cisco, Glue Networks, Tata Communications, имеет ряд преимуществ, которые позволяют обезопасить сеть от разного вида ущерба, как например –это отказоустойчивость и нарушение работы проводных соединений.

Суть работы SD-WAN заключается в создании единой виртуальной сетевой фабрики, состоящей из защищенных overlay-соединений поверх физических подключений, которыми могут быть выделенные каналы, недорогие Интернет-каналы, или даже LTE. Посредством мониторинга производительности WAN-линков и инспектирования пакетов, система принимает решения о направлении трафика через то или иное подключение, тем самым оптимизируя использование каналов. При этом одним из главных требований к платформе — возможность гарантирования качества сервиса при использовании публичных каналов связи[4].

Главное преимущество SD-WAN заключается в возможности управления трафиком на уровне приложений. Решение может/должно адаптировать ресурсы сети под требования конкретного приложения, предоставляя статистические данные в режиме реального времени. Это позволяет администраторам идентифицировать и устранять проблемы с качеством сервиса.

Важно не путать понятие SD-WAN с трактовкой SDN от организации ONF, предполагающей использование протокола OpenFlow как одного из основных компонентов платформы. Реализация SD-WAN возможна с использованием существующих технологий и протоколов (например, в решении Cisco IWAN задействованы широко известные технологии и протоколы – DMVPN, PFR, BGP, EIGRP, NBAR, WAAS) [2].

Концепция работы SD-WAN сетей

SD-WAN призвана решить распространенные задачи WAN, сложно решаемые с помощью традиционных подходов.

Сложность управления. Гибридная WAN-сеть – типичный случай для территориально распределенной корпоративной сети. Различные типы подключения филиалов, используемые протоколы и методы шифрования вызывают головную боль у сетевых инженеров, ответственных за администрирование такой сети. К этому еще можно добавить процедуру взаимодействия с каждым оператором связи.

Компании, использующие публичные облачные сервисы для хостинга инфраструктуры, в значительной мере теряют полный контроль над управлением сетевой инфраструктурой за пределами корпоративной сети [3].

SD-WAN призвана решить вышеописанные задачи.

Скептики могут возразить, что SD-WAN звучит как очередной трюк маркетологов, которые пытаются «наложить» уже хорошо известный термин «SDN» на традиционные WAN-сети. Но пока маркетологи раскручивают очередную «Software Defined» концепцию, сам подход действительно отличается от традиционных WAN-сетей.

Преимущества SD-WAN сетей

Технология SD-WAN имеет много новых функций, использование которых в сценарии распределенных сетей может сильно сократить операционные расходы и повысить эффективность. Ее гибкость, масштабируемость и простота внедрения дает весомые преимущества перед традиционными выделенными глобальными сетями или другими технологиями виртуальных частных сетей.

Традиционные WAN-сети обеспечивают очень высокую скорость, надежность и устойчивость, однако к их минусам можно отнести очень высокую стоимость (прокладка кабеля от 150 до 500 рублей за метр) и долгий срок выполнения работы (от недели до нескольких месяцев). Более экономичный вариант – виртуальная частная сеть (VPN) точка-точка между двумя объектами через сеть Интернет, но он делает неудобным связь и управление большим числом удаленных филиалов.

Архитектура SD-WAN стремится решить многие проблемы описанных технологий WAN, делая процесс создания такой сети более гибким. Также к ее преимуществу можно отнести возможность построения поверх любого канала, будь то выделенная линия или Интернет, централизованное управление и настройка с помощью контроллера. Даже если ваша WAN-сеть использует MPLS, SD-WAN поможет задействовать ее ресурсы с наибольшим эффектом [1].

Что делает архитектуру SD-WAN серьезным конкурентом другим способам – это его функции, направленные на агрегацию, выбор и настройку пути следования трафика. Какие преимущества дает такая технология:

- В традиционных WAN-сетях динамические протоколы маршрутизации довольно грубо управляют трафиком, направляя его как поток воды по заданному руслу. SD-WAN абстрагирует технологии маршрутизации, безопасности и физические устройства, создавая логическое подключение, в котором можно как угодно управлять трафиком, гарантировать доставку пакетов, балансировать нагрузку и оптимизировать линки.
- Для использования SD-WAN нет необходимости наполнять стойку дорогостоящим сетевым оборудованием. В большинстве случаев

необходимы лишь небольшие SD-WAN-устройства, коммутаторы доступа или беспроводные точки доступа.

- SD-WAN-сеть не только легко развернуть, но также легко ею управлять, осуществлять поиск и устранение неисправностей за счет наличия контроллера, который берет эти задачи на себя. За счет агрегации нескольких каналов, достигается высокая отказоустойчивость, неисправность можно устранять в рабочем порядке без падения производительности или отключения каких-либо сервисов.

- Еще одно преимущество SD-WAN – применение виртуализированных WAN, которые позволяют легко сегментировать пользователей внутри WAN. Раньше было невозможно сделать это без использования Dynamic Multipoint VPN (DMVPN) поверх существующей WAN-сети. Теперь SD-WAN позволяет предоставить такую услугу клиентам, которые хотят изолировать трафик от дата-центра до подключенных к WAN филиалов.

Конечно, у SD-WAN есть и минусы:

- Во-первых, это использование Интернета для передачи данных. Ни одна агрегация нескольких интернет соединений не сможет обеспечить такой же уровень надежности и безопасности, как выделенная линия.

- Во-вторых, оборудование SD-WAN принимает только ethernet-соединения, исключая некоторые схемы подключения.

- В-третьих, решения разных поставщиков SD-WAN несовместимы, а значит, свободно менять их по необходимости не такая простая задача[5].

Заключение

Технологии меняются и со временем появляются все более совершенные способы коммутации и управления трафиком. Как системы DPI для операторов связи, так и SD-WAN для компаний с распределенными сетями дают возможность использовать каналы связи с максимальной эффективностью и надежностью.

Фундаментальная разница между мешаниной традиционных протоколов, насильно втиснутых в архитектуру гибридных WAN и SD-WAN – архитекторы продуктов SD-WAN не имеют проблем с унаследованными реализациями, не должны повторно использовать код, который был разработан для решения совсем иных задач, могут не использовать не оптимальные для задачи протоколы.

Архитектура большинства продуктов SD-WAN за счёт этого гораздо проще и легче конфигурируется, чем традиционные гибридные сети. Однако не следует забывать, что большинство из них использует проприетарные протоколы.

После проведенных разными изданиями экспериментов, можно сделать вывод что SD-WAN – это новый виток развития компьютерных сетей, который

не только впечатляет своими решениями в области проектирования сети, но и предоставляет преимущества. Технология призвана чтобы увеличить отказоустойчивость сети, уменьшить к минимуму потери пакетов. Ведь даже при нарушении работы проводной системы для передачи пакетов, пакеты начинают передаваться по без проводной системе.

Список литературы

1. Open Networking Use Group Интернет ресурс. Режим доступа: <http://opennetworkingusergroup.com/onug-spring-2014-use-cases/software-defined-wide-area-network-sd-wan/>
2. SDNBLOGGER Что такое SD-WAN? Интернет ресурс. Режим доступа: <https://sdnblog.ru/what-is-sdwan/>
3. Citrix Скажи нет сбоям в сети: почему SD-WAN, Интернет ресурс. Режим доступа: <http://emeahub.citrix.com/ru/no-more-network-failure-why-sd-wan-is-a-game-changer/>
4. Борчев Д., Программно-определяемые WAN: хорошо сделанная изолянта? Интернет ресурс. Режим доступа: <https://habrahabr.ru/post/260977/>
5. Хазов В., SD-WAN как конкурент традиционному MPLS Интернет ресурс. Режим доступа: <http://blog.vasexperts.ru/?p=88>
6. Vas Experts Tata принесет SD-WAN в 130 стран Интернет ресурс. Режим доступа: <https://habrahabr.ru/company/vasexperts/blog/316278/>
7. Бубнов И., SDN,NFV,SD-WAN и тенденции современного интернета Интернет ресурс. Режим доступа: https://geekbrains.ru/posts/route_future

УДК 004.921

ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ АРХИТЕКТУР ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ ДЛЯ ВИЗУАЛИЗАЦИИ 3D-МОДЕЛИ

Крахмаль М.В., Кривошеев С.В.

Донецкий национальный технический университет
кафедра компьютерной инженерии
E-mail: marry_krahmal@mail.ru

Аннотация

Крахмаль М.В., Кривошеев С.В. Применение параллельных архитектур вычислительных систем для визуализации 3d-модели. В работе рассматриваются классификации параллельных вычислительных систем, особенности в использовании CPU и GPU, возможности синтеза изображений на параллельных архитектурах специализированных графических вычислительных систем методом трассировки лучей.

Ключевые слова: 3D-визуализация, параллельные архитектуры, GPU, CUDA, метод трассировки лучей.

Введение

Применение параллельных вычислительных систем (ПВС) является одним из основных направлений вычислительной техники. Это связано с постоянным наличием вычислительных задач, для решения которых уже существующих средств стало недостаточно.

На сегодняшний день параллельным вычислениям уделяется огромное внимание, что связано с определенными причинами. Главным фактором является научно-технический прогресс, где большинство задач необходимо решать с помощью различных методов математического моделирования, которые возможно реализовать на базе только высокопроизводительной техники. Вторым фактором является развитие и распространение параллельных компьютеров [1].

Построение параллельности вычислений, при одновременном выполнении множества операций, осуществляется за счет многопроцессорности. Есть возможность ускорения процесса решения вычислительных задач за счет осуществления разделения алгоритма на информационно независимые части.

ПВС широко применяются в сферах, где необходимо проводить большие расчеты. В системе автоматизированного проектирования выполняются проектные работы с применением компьютерной техники для решения конструкторских задач. Удастся значительно ускорить процесс проектирования за счет применения параллельных информационных технологий, что снижает временные и трудовые затраты на разработку и изготовление новой модели.

Классификации архитектур ПВС

Первый вариант классификации архитектур вычислительных систем (ВС) был предложен М.Флинном в 1966 году. За основу было взято понятие потока. Подразумевается последовательность элементов, данных или команд, которые обрабатываются на процессоре. Основной классификацией при функциональной точке зрения является соотношение между потоком команд и потоком данных, на основании чего было определено 4 типа архитектур [2].

Под архитектурой ВС понимается работа нескольких АЛУ под управлением одного устройства управления. То есть способность обрабатываться большому количеству данных по одной программе.

Архитектура SISD (Single Instruction Single Data) – одиночный поток команд и данных (рис. 1 – а). Обеспечивается параллельная работа устройств системы, процессор занимается обработкой информации только по данной команде. Выполнение параллельно разных частей программы или нескольких программ в такой системе невозможно.

Архитектура SIMD (Single Instruction Multiple Data) – одиночный поток команд и множественный поток данных (рис. 1 – б). В этой системе используется большое количество процессорных элементов, которые выполняют одну и ту же функцию, но над разными данными.

Архитектура MISD (Multiple Instruction Single Data) – множественный поток команд и одиночный поток данных (рис. 1 – в). В такой системе реализован принцип конвейерной обработки. Такая организация представляет собой регулярную цепочку последовательно соединенных элементов, которые образуют процессорный конвейер.

Архитектура MIMD (Multiple Instruction Multiple Data) – множественный поток команд и данных (рис.1 – г). В такой ВС каждый процессорный элемент выполняет свою задачу над своим потоком данных. Реализуются мультипроцессорные и многомашинные системы [3].

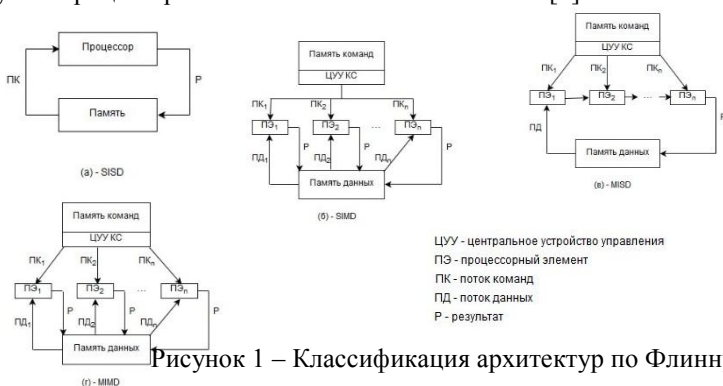


Рисунок 1 – Классификация архитектур по Флинну

Использование архитектур графических систем для визуализации

Новые технологии позволяют создавать визуализацию проектируемого объекта для наглядной оценки, которая максимально приближена к реальному устройству. Трехмерная модель будущего механизма ускоряет и облегчает работу, так как позволяет еще на этапе проектирования вычислить оптимальные параметры устройства и определить недостатки.

Одним из распространенных способов реализации аппаратного ускорения на ПВС является использование графических процессоров (GPU). Такая технология называется GPGPU (GPU общего назначения).

Главной функцией GPU является получение изображения по модели с помощью программы. То есть необходимо принять на входе полигоны, произвести над ними все необходимые математические и логические операции, и на выходе выдать координаты пикселей. Вся работа GPU заключается в обработке большого количества независимых между собой задач.

На рисунке 2 показана структура и занимаемое пространство CPU и GPU.

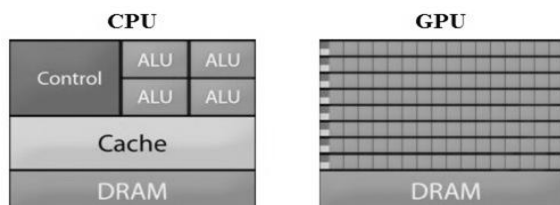


Рисунок 2 – Структура CPU и GPU

Используя эту структуру можно сделать вывод, что в CPU большую часть занимает кэш-память и устройства управления. В структуре GPU все по-другому, вычислительные модули занимают практически все пространство кристалла. Благодаря большому количеству ядер можно получить высокую производительность при вычислениях на GPU. Из этого следует, что GPU целесообразней использовать для параллельных вычислений. Использование нескольких GPU дает намного больше вычислительных мощностей.

В качестве ПВС можно использовать графический мультипроцессор видеокарт NVidia, основанный на архитектуре CUDA. CUDA - программно-аппаратная архитектура, которая позволяет производить вычисления с использованием графических процессоров NVidia, поддерживающих технологию GPGPU. Метод прямой и обратной трассировки лучей хорошо масштабируется по числу процессоров. Так как лучи могут трассироваться независимо друг от друга, то возможно каждому процессору ВС выделить определенную часть изображения для дальнейшей обработки [4].

Расчет изображений может осуществляться независимо для правого и левого изображений. Процесс расчета может быть распараллелен во времени и отображен на параллельные архитектуры ВС - CUDA. Из этого формируется следующий процесс компьютерного синтеза:

- 1) Параллельная независимая реализация синтеза «левый канал» - «правый канал» на мультипроцессоре;
- 2) Параллельная «внутриканальная» реализация визуализации методом трассировки лучей на ядрах мультипроцессора, выделенных под каждый канал.

После того, как будет выполнен расчет изображения, с помощью GPU можно выполнить процесс постобработки кадров и визуализацию (рис. 3).

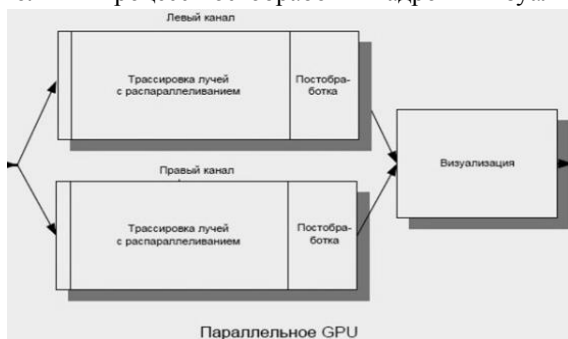


Рисунок 3 – Процесс синтеза изображения с использованием GPU-CUDA

Выводы. В данной работе были рассмотрены классификации архитектур вычислительных систем, методы повышения производительности вычислительных систем за счет параллельных вычислений, реализация подхода 3D-визуализации с использованием графических процессоров GPU (технология CUDA).

Список литературы

- 1) Параллельное программирование с использованием технологии CUDA / [Электронный ресурс] – Режим доступа: <http://edu.chpc.ru/cuda/main.html>
- 2) Головкин Б.А. Параллельные вычислительные системы. М.: Наука, 1980. - 520 с.
- 3) Таненбаум Э. Архитектура компьютера, 4-е изд. – Спб.: Питер, 2003. – 700 с.
- 4) Основы работы с технологией CUDA / Боресков А. В., Харламов А. А. – ДМК Пресс, 2010. – 232 с.
- 5) Дёмин А.Ю. Основы компьютерной графики: учебное пособие – Томск: Изд-во ТПУ, 2011. – 191 с.

УДК 004.942

ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ АРХИТЕКТУР ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ ДЛЯ РАСЧЕТА ПАРАМЕТРОВ ДВИЖЕНИЯ ТРАНСПОРТНОГО СРЕДСТВА

М. К. Кравченко, С. В. Кривошеев

Донецкий Национальный Технический университет, факультет компьютерных наук и технологий, кафедра компьютерной инженерии.

E-mail: kravchenko_misha_1996@mail.ru

Аннотация

Кравченко М.К., Кривошеев С.В. Применение параллельных архитектур вычислительных систем для расчета параметров движения транспортного средства. В данной статье рассмотрены проблемы, связанные с автоматизацией движения транспортных средств. Определены способы ускорения вычислений, для расчетов параметров движения. Выполнен анализ современных параллельных вычислительных устройств и способы их применения.

***Ключевые слова:** параллельные вычислительные системы, CUDA, GPU, транспортное средство, анализ направления движения.*

Актуальность темы

В наше время количество транспортных средств растет с большой скоростью. На глобальный уровень становятся проблемы составления маршрутов, анализ движения. Грамотно спланированный маршрут и автоматизация управления позволяют избежать лишних затрат топлива, а соответственно и уменьшить вред экологии, свести к минимуму аварии и катастрофы [2]. Все больше различных институтов по всему миру развивает это направление. Большой проблемой остается скорость, с которой будут выполняться расчеты движения и анализ выбора маршрута. Каждую секунду на транспортное средство действуют внешние факторы, изменяется обстановка. Поэтому необходимо постоянно пересчитывать и анализировать большое количество данных. Использование параллельных вычислительных систем позволяет увеличить скорость расчетов, не прибегая к увеличению вычислительных мощностей процессора.

Цель статьи

Провести анализ современных параллельных вычислительных устройств и систем для работы с ними. Определить способы взаимодействия CPU и GPU.

Оценить возможности применения параллельных вычислений для расчета параметров движения транспортного средства.

Обзор параллельных вычислительных систем

Сейчас практически любой гаджет имеет многоядерный процессор. Это дает возможность выполнять параллельные вычисления. Ядром называют часть процессора, которая выполняет один или несколько потоков команд. В простой многоядерной системе ядро выполняет один поток команд независимо от других ядер и имеет собственную кэш-память первого и второго уровней. В многоядерных системах с общей памятью каждое ядро имеет собственную кэш-память первого и общую память второго уровня, а ядро может выполнять несколько потоков команд.

Все большую популярность набирает вычисление на графических процессорах (GPU). Эти процессоры изначально создавались, как многоядерные, поскольку для обработки графики необходимо производить большое количество вычислений. Можно подумать, есть ли смысл использовать CPU, если графические процессоры производят вычисления гораздо быстрее. Но следует учитывать, что GPU являются SIMD системой (одиночный поток команд, множественный поток данных). То есть работают с большим количеством различных данных, выполняя при этом одинаковые команды над ними. Разница в структуре CPU и GPU (рис. 1) очевидна.

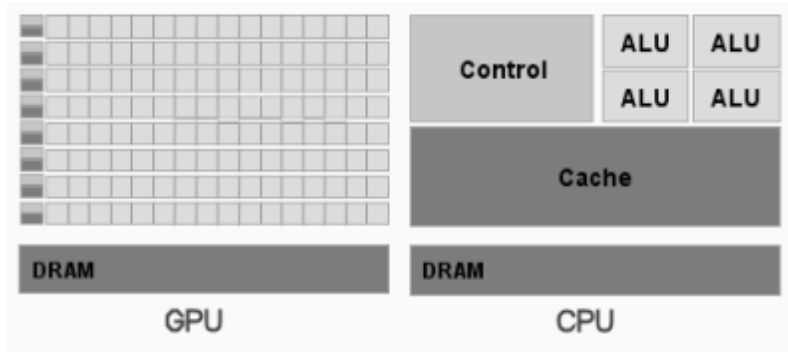


Рисунок 1 – Структура CPU и GPU

Простой способ понять разницу между GPU и CPU - сравнить то, как они выполняют задачи. CPU состоит из нескольких ядер, оптимизированных для последовательной обработки данных, в то время как GPU имеет тысячи более мелких ядер, созданных для выполнения нескольких задач одновременно [4].

Применение технологии CUDA

Для программирования GPU необходима определенная программно-аппаратная платформа. Компания NVIDIA предоставляет удобную платформу CUDA. Главным преимуществом которой, является ее бесплатность и простота использования (программирование производится на расширенном языке C). Технология CUDA позволяет задействовать любое АЛУ микросхемы [1].

Поскольку GPU ориентированы на обработку графики, то вычисление на данных процессорах необходимо выполнять, загружая данные в 128-битные структуры, которые будут выполняться на графическом конвейере. Технология CUDA скрывает работу графического конвейера и предоставляет более удобную работу программисту. Для выполнения на GPU нужно всего лишь перед участком кода на C указывать некоторую служебную информацию, которая указывает, что данный код необходимо выполнить на GPU. Обработка данных в приложения идет таким образом, что GPU обрабатывает части приложения, требующие большой вычислительной мощности, при этом остальная часть приложения выполняется на CPU. С точки зрения пользователя, приложение просто работает значительно быстрее [4].

Описание работы

Анализ направления движения транспортного средства (ТС) подразумевает наличие данных о местонахождении и окружающей обстановке (что впереди, сзади, по бокам). ТС имеет четко фиксированный набор вариантов продолжения движения: продолжить движение, уменьшить скорость, увеличить скорость, задний ход, повернуть вправо/влево.

Анализ можно разделить на два этапа:

1. Проверка, можно ли продолжить движение в данном направлении (нет ли впереди препятствия или неудовлетворяющей движению местности).
2. Выбор оптимального маршрута, если движение в том же направлении не возможно.

Как первый, так и второй этап могут дать лишь относительный ответ, поскольку в действительности можно выполнить просчет на ограниченное число шагов, так как на каждом шаге анализа количество вычислений возрастает по геометрической прогрессии (рис. 2), а время на ответ ограничено допустимым порогом 100 мс [3].

Необходимо так же учитывать тот факт, что ТС не может мгновенно изменить курс. Скорость «реакции» на изменение положения рулевого механизма будет зависеть от массы ТС, его сцепления с поверхностью. Если отбросить второй этап и определять только, можно ли продолжать движение в том же направлении, то время ответа будет гораздо меньше. На каждом шаге будет анализироваться лишь одно состояние (можно или нельзя перейти в эту

позицию). Таким образом, нагрузка на вычисления уменьшится в разы, что позволит производить вычисления для нескольких ТС одновременно.

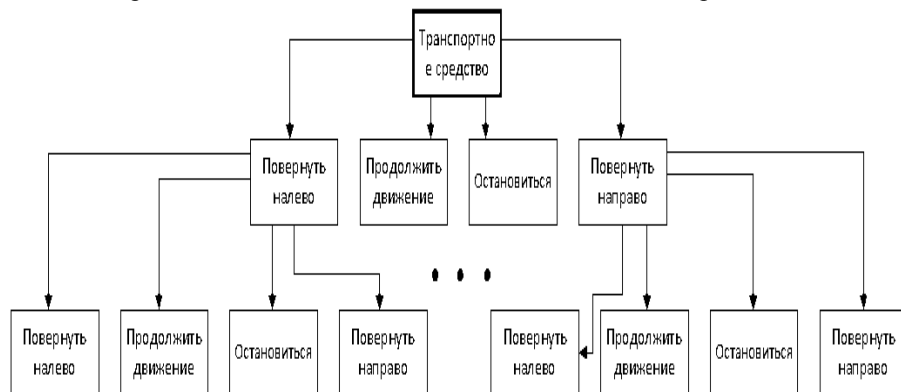


Рисунок 2 – Пример определения движения на два шага

Выводы

Определение направления дальнейшего движения транспортного средства требует значительных вычислительных мощностей. Использование параллельных вычислительных систем позволит выполнить расчет на большее количество шагов, что обеспечит более точные и правильный анализ дальнейшего движения.

Список литературы

1. Технология CUDA в примерах: введение в программирование графических процессоров; Пер. с англ. Слинкина А. А., научный редактор Боресков А. В. – М.: ДМК Пресс, 2013. – С. 17-21.
2. Кривошеев С.В. Исследование эффективности параллельных архитектур вычислительных систем для расчета параметров движения транспортного средства // Научные труды Донецкого национального технического университета. Выпуск № 1(10)-2(11). Серия «Проблемы моделирования и автоматизации проектирования». – Донецк, ДонНТУ, 2012. С. 207-214.
3. Eric Hansen, Terry Huntsberger and Les Elkins, Autonomous maritime navigation: developing autonomy skill sets for USVs, Proc. SPIE 6230, 62300U (2006).
4. Вычисления с ускорением на GPU [электронный ресурс] // NVIDIA Corporation: [сайт]. [2017]. URL: <http://www.nvidia.com.ua/object/gpu-computing-ru.html>

УДК 004.2

МЕТАЭВРИСТИЧЕСКИЙ АЛГОРИТМ ЛЕТУЧИХ МЫШЕЙ ДЛЯ ПОСТРОЕНИЯ МАРШРУТА ПУТИ РОБОТА

Иорданов Р. В., Завадская Т. В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
E-mail: roma-iordanov@mail.ru

Аннотация

Иорданов Р.В., Завадская Т.В. Метаэвристический алгоритм летучих мышей для построения маршрута пути робота. Описаны обязательные правила поведения агентов в алгоритме летучих мышей. Определены этапы реализации алгоритма.

Ключевые слова: робототехника, алгоритм летучих мышей, маршрут, путь, математический аппарат.

Введение

Робототехника является наиболее прогрессирующей областью. С каждым годом ученые и инженеры находят новые пути для развития данной отрасли в хозяйстве, военном деле, медицине. Отдельной ветвью развития данной области является групповая робототехника [1].

Данная работа освящает проблему построения маршрута с обходом препятствий группой роботов, поведение которых описывается алгоритмом летучих мышей.

Постановка задачи

Алгоритм летучих мышей (Bat Algorithm, BA) был разработан Янгом (X.-Sh. Yang) в 2010 году [2]. BA является метаэвристическим алгоритмом, что позволяет ему быть непривязанным к конкретной задаче, в отличие от эвристических [3]. Данный алгоритм основывается на поведении коллективных животных (волки, птицы, рыбы, пчёлы) и их способностям выживать в дикой природе.

Летучие мыши имеют совершенный эхолокационный аппарат, который позволяет им без труда находить добычу, препятствия и в полной темноте размещаться на насесте [4]. Они определяют расстояния до объекта, рассчитывая время между импульсом и его отражением.

В алгоритме летучих мышей обязательными являются следующие правила:

1. Летучие мыши используют эхолокацию, чтобы определять расстояние, находить добычу, препятствия.

2. Летучие мыши передвигаются случайным образом с некоторой скоростью v_i , в позиции S_i , с фиксированной частотой $[f_{max}, f_{min}]$. Регулируемыми параметрами являются: длина испускаемой волны λ , частота импульса r $[0;1]$ и громкость a .

3. Громкость меняется от максимального a_{max} к минимальному (постоянному) a_{min} [5].

Реализация

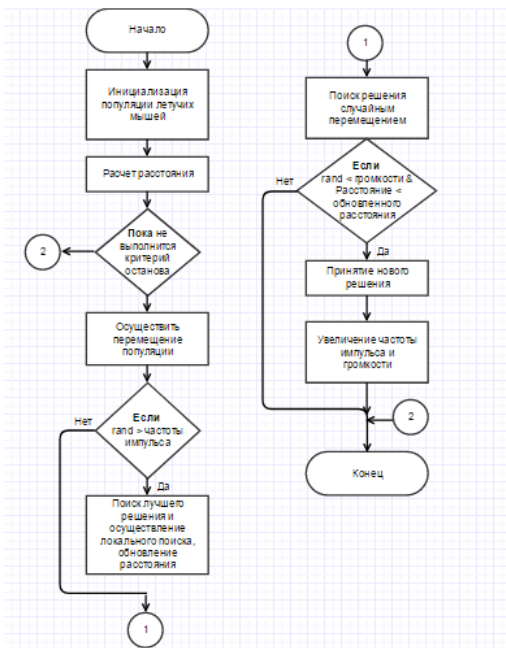


Рисунок 1 – Блок-схема алгоритма летучих мышей

Работу алгоритма можно разделить на следующие этапы:

Этап 1: Инициализировать популяцию летучих мышей в позиции S_i ($i = 1, 2 \dots n$) и скорость v_i .

Этап 2: Инициализировать частоту f_i в позиции S_i

Этап 3: Определить начальную частоту импульса r_i и громкость a_i .

Этап 4: Измерить расстояние между начальным и конечным значениями:

$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, где x_1, y_1 – координаты точки старта, x_2, y_2 – координаты точки финиша.

Этап 5: Вычислить частоту для новых решений через следующую формулу:

$f_i = f_{min} + (f_{max} - f_{min}) * U(0,1)$, где U – случайное число.

Этап 6: Обновить скорость и глобальное лучшее решение через следующие формулы:

$v_i^t = v_i^{t-1} + (x_i^t - best)f_i$, $S_i^t = S_i^{t-1} + v_i^t$, где best – лучшее текущее решение.

Этап 7: Локальное лучшее решение для каждой из летучих мышей вычисляется следующим образом:

$$S_{new} = S_{old} + \varepsilon * a^t$$

На этапе инициализации популяции летучих мышей начальные значения их громкости (a), частоты (f) и частоты импульса (r) представляется в виде значений на промежутках: $[a_{min}, a_{max}]$, $[f_{max}, f_{min}]$, $[r_{min}, r_{max}]$. Новое решение находится случайным полетом летучих мышей в границах предыдущего лучшего решения. После нахождения препятствия уменьшается громкость и увеличивается частота ультразвукового импульса [6].

Заключение

В работе был рассмотрен математический аппарат алгоритма летучих мышей для поиска маршрута пути робота. Этот алгоритм полностью описывает перемещение агентов в двумерном пространстве. Приведенное исследование в дальнейшем будет использовано для реализации алгоритма.

Список литературы

1. Групповая роботехника. Интернет ресурс. Режим доступа: <https://ru.wikipedia.org/wiki/%D0%93%D1%80%D1%83%D0%BF%D0%BF%D0%BE%D0%B2%D0%B0%D1%8F%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0>.
2. Частикова В.А., Новикова Е. Ф. Алгоритм летучих мышей для решения задачи глобальной оптимизации. – 2015. – С. 2.
3. Yogita Gigras, Kusum Gupta, Vandana, Kavita Choudhary. A Comparison between Bat Algorithm and Cuckoo Search for Path Planning. – 2015. – С. 1-2.
4. Карнено А. П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов. – 2012. – С. 20-24.
5. Yogita Gigras, Kusum Gupta, Vandana, Kavita Choudhary. A Comparison between Bat Algorithm and Cuckoo Search for Path Planning. – 2015. – С. 5-8.
6. Карнено А. П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов. – 2012. – С. 24-28.

УДК 681.3

РАЗРАБОТКА СИСТЕМЫ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ

Н.А. Горин, В.Н. Струнилин

Донецкий национальный технический университет, г. Донецк
кафедра компьютерная инженерия
E-mail: MegaNikolai@yandex.ua

Аннотация

Горин Н.А., Струнилин В.Н. Разработка системы дистанционного управления для мобильного робота. Выполнена разработка системы для дистанционного управления мобильным роботом, посредством которой будет производиться движение робота по требуемому маршруту.

Ключевые слова: микроконтроллер, модуль связи, дистанционное управление, FPV система.

Постановка проблемы. Современная робототехника всё больше стремится к автономности, но учесть все возможные ситуации на данный момент невозможно, так как ещё не создан искусственный интеллект. При возникновении непредвиденных преград либо сложных условий для перемещения мобильного робота, возникает необходимость вмешательства оператора и полный перехват управления некоторыми системами. В случае если условия, с которыми предстоит столкнуться мобильному роботу, непредсказуемы и для них нельзя составить алгоритмы для безопасного перемещения робота, дистанционная система управления и действия оператора являются единственным правильным решением.

Цель статьи – разработать систему для дистанционного управления мобильным роботом, посредством которой будет производиться направление движения робота по требуемому маршруту.

Выбор платформы. Разработка дистанционного управления производится для мобильного робота, описанного в статье [1] и разработанных для него систем ориентирования в пространстве [2, 3]. Мобильный робот построен на платформе Arduino Uno, питание платформы осуществляется от батареи [4]. Разработка пульта управления осуществляется на платформе Arduino Nano, так как она имеет более компактные размеры. Обе платформы программируются посредством программного обеспечения Arduino с использованием среды разработки Arduino 1.0.5.

Структурная схема аппаратной части состоит из двух блоков – структурная схема робота и структурная схема пульта управления (рис.1).

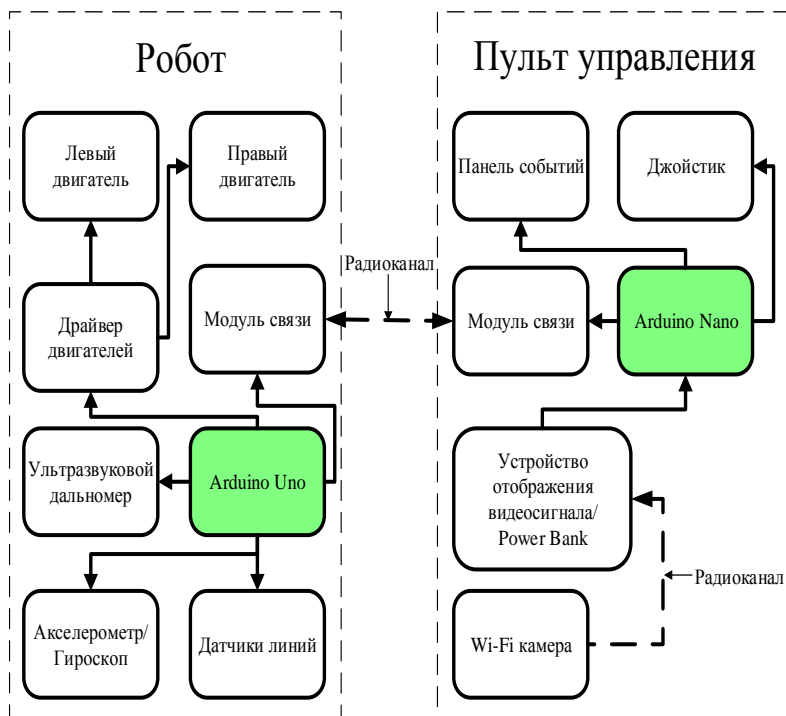


Рисунок 1 – Структурная схема аппаратной части проекта

Обоснование выбора модуля nRF24L01+. В качестве модулей связи используется nRF24L01+ (рис.2). Модуль nRF24L01+ включает в себя 2.4 ГГц RF трансивер, логику, которая поддерживает высокоскоростной SPI интерфейс для подключения и обмена данными. Преимуществом данного радио-трансивера является дешевизна и быстрдействие. Модуль поддерживает работу со скоростью до 2 Мбит/с и может работать на 126 независимых каналах передачи данных. Для передачи данных используется собственный проприетарный протокол. Недостатком данного модуля является то, что большинство окружающее нас радиооборудование, такое как Wi-Fi роутеры, Bluetooth гарнитуры и другие бытовые устройства работают на данной частоте и большая часть каналов может быть зашумлена и не пригодна для использования.

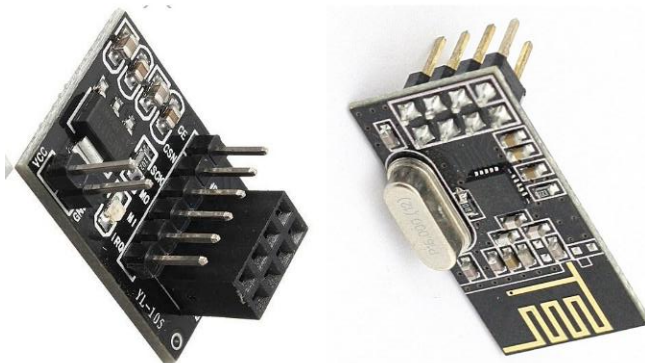


Рисунок 2 – Радио модуль nRF24L01+

Обоснование построения FPV системы. Система FPV (First Person View) построена на основании экшен-камеры EKEN H9R (рис.3) и планшета Lenovo TAB3 7 Essential. Камера имеет небольшой вес, а именно 65 грамм, что является её преимуществом. Передача видеосигнала осуществляется посредством Wi-Fi канала между камерой и планшетом. Посредством установленной программы Ez iCam на планшете, производится синхронизации камеры и планшета. Данная программа обеспечивает дистанционную работу с камерой, а именно просмотр получаемого изображения и настройка параметров камеры. В качестве приёмника видеосигнала можно использовать любой современный планшет или смартфон, имеющий Wi-Fi модуль на борту. Дальность связи такой FPV системы, ограничивается техническими характеристиками технологии Wi-Fi.



Рисунок 3 – Внешний вид экшен-камеры EKEN H9R

Выводы. Разработана система для дистанционного управления мобильным роботом, посредством которой производится направление движения робота по необходимому маршруту. Были изучены особенности передачи данных по радиоканалу и построения FPV систем. Аппаратная часть проекта основана на недорогих и распространённых модульных компонентах, что сказывается на точности полученных данных с систем ориентирования в пространстве, но отклонения не критичные и это не сказывается на работоспособности в целом. Достоинством является то, что в дальнейшем удобно проводить улучшение систем либо полностью перерабатывать их для большей универсальности и функциональности. Разработка систем ориентирования в пространстве и роботостроение имеют большие перспективы для развития, так как с каждым годом растёт уровень роботизации, окружающей нас техники. Системы ориентирования в пространстве всё больше проникают в нашу повседневную жизнь: от мобильных телефонов (датчики приближения, GPS, акселерометр и многое другое) до автомобилей (GPS, парктроник). Внедрение данных систем в аэрокосмическую отрасль является неотъемлемой частью прогресса в этой области. Следовательно, изучение данной тематики является перспективной, так как актуальность растёт с каждым днём.

Список литературы

1. Горин Н.А. Разработка системы ориентирования в пространстве с использованием ультразвукового датчика/ Н.А. Горин, В.Н. Струнилин // *Материалы Международной научно-технической конференции студентов, аспирантов и молодых ученых «Информационные управляющие системы и компьютерный мониторинг».* – Донецк, 2015. – С. 34-36.
2. Горин Н.А. Разработка системы ориентирования в пространстве с использованием модуля GY-531/ Н.А. Горин, В.Н. Струнилин // *Материалы Международной научно-технической конференции студентов, аспирантов и молодых ученых «Информатика, управляющие системы, математическое и компьютерное моделирование».* – Донецк, 2016. – С. 534-538.
3. Горин Н.А. Применение инфракрасного отражателя TCRT5000 в системе ориентирования в пространстве / Н.А. Горин, В.Н. Струнилин // *Материалы Международной научно-технической конференции студентов, аспирантов и молодых ученых «Информатика, управляющие системы, математическое и компьютерное моделирование».* – Донецк, 2016.
4. Платформа Arduino Uno [электронный ресурс] // *Официальный сайт Ардуино в России:* [сайт]. [2016]. URL: <http://arduino.ru/Hardware/ArduinoBoardUno>.

Секция № 9
Системный анализ
и управление



УДК 517.9

ON THE POSSIBLE SEMINORMS ON CERTAIN WEIGHTED HOLDER SPACES IN MATHEMATICALS MODELS OF THE THIN FILMS

Degtyarev S.P.

State Institute for applied mathematics and mechanics, Donetsk,
Donetsk national technical university, Faculty of Computer Science and Technology

E-mail: degtyar@i.ua

Annotation

Degtyarev S.P. On the possible seminorms on certain weighted Holder spaces in mathematical models of the thin films. The presentation is devoted to studying of some weighted Holder spaces. These spaces are designed in the way to serve as a framework for studying different statements for the thin film equations in weighted classes of smooth functions in the multidimensional setting. These spaces can serve also for considering of other equations with the degeneration on the boundary of the domain of definition.

Introduction

The present paper is devoted to studying of some weighted Holder spaces $C_{n,\omega\gamma}^{m+\gamma, \frac{m+\gamma}{m}}$. These spaces were introduced in [1] and they are designed in the way to serve as a framework for consideration of different statements for the thin film equations in weighted classes of smooth functions in the multidimensional setting. These spaces can serve also for considering of other equations with the degeneration on the boundary of the domain of definition, for example, in the spirit of [2]. The literature on the subject of the thin film equations is very numerous but almost all results with sufficient regularity are devoted to the case of one spatial variable. As a possible target for an application of the spaces $C_{n,\omega\gamma}^{m+\gamma, \frac{m+\gamma}{m}}$ we only mention the papers [3]-[5].

Statement of the problem

The spaces $C_{n,\omega\gamma}^{m+\gamma, \frac{m+\gamma}{m}}$ arise at the considering of linearised version of the thin film equations. Let us explain this on the example for the thin film equation in the case of partial wetting (see, for example, [3] for the accurate statement). Consider the thin film equation of fourth order for an unknown function $h(x,t)$

$$\frac{\partial h}{\partial t} + \nabla(h^n \nabla \Delta h - \beta \nabla h) = f(x,t), \text{ in } \Omega, \quad (1)$$

where $n > 0$ is fixed, Ω is a half space $\Omega = \{(x, t) : x = (x', x_N) \in R^N, x_N > 0, t > 0\}$. Consider also partial wetting conditions at

$$x_N = 0 \quad h(x', 0) = 0, \quad \frac{\partial h}{\partial x_N}(x', 0) = 1 \quad (2)$$

and an initial condition

$$h(x, 0) = w(x). \quad (3)$$

From (2) it follows that we must have for $w(x)$

$$w(x', 0) = 0, \quad \frac{\partial w}{\partial x_N}(x', 0) = 1. \quad (4)$$

Consequently, we have

$$w(x) \approx x_N, \quad x_N \rightarrow 0. \quad (5)$$

The linearization of equation (1) at the initial datum $w(x)$ means that we denote in (1) $h = w + u$ and extract linear with respect to u part (we also drop lower order terms). Formally, one can just replace h^n by w^n in (1) and replace h by u in other places of this equation. Taking into account (5) and replacing w by just x_N , we arrive at

$$\frac{\partial u}{\partial t} + \nabla(x_N^n \nabla \Delta u - \beta \nabla u) = f(x, t). \quad (6)$$

If we are going to consider equations (6) (and correspondingly (1)) in classes of Holder functions we have to consider $f(x, t)$ in (6) from some (may be weighted) Holder class. This leads to the consideration of $\nabla(x_N^n \nabla \Delta u)$ from the same

weighted Holder class. In our definition below this will be the class $C_{n, n\gamma/4}^{4+\gamma, \frac{4+\gamma}{4}}$. In the case of second order equations such classes were used in [2]. Note that we

consider the framework of classes $C_{n, \omega\gamma}^{m+\gamma, \frac{m+\gamma}{m}}$ as an alternative for considering the Carnot-Carathéodory metric for studying degenerate equations in classes of smooth. Therefore, in this paper we are going to present the equivalence of the Carnot-

Carathéodory metric in spaces $C_{n, \omega\gamma}^{m+\gamma, \frac{m+\gamma}{m}}$ to some another weighted metric in these spaces. Note that in the case of elliptic equations more simple weighted Holder classes with unweighted Holder constants can be used. The reason is that in the

elliptic case no agreement between smoothness in x -variables and t -variable is needed.

Let us turn now to exact definitions and to the main results. Denote $H = \{x = (x', x_N) \in R^N, x_N > 0\}$, $Q = \{(x, t) : x \in H, t > 0\}$. Let m be a positive integer and let n be a positive number, $n < m$. Denote

$$\omega = n/m < 1.$$

Let $C_{\omega\gamma}^\gamma(\bar{H})$, $\gamma \in (0, 1)$, be the weighted Hölder space of continuous functions $u(x)$ with the finite norm

$$\|u\|_{\omega\gamma, \bar{H}}^{(\gamma)} \equiv \|u\|_{C_{\omega\gamma}^\gamma(\bar{H})} \equiv |u|_{\bar{H}}^{(0)} + \langle u \rangle_{\omega\gamma, \bar{H}}^{(\gamma)}, \quad (7)$$

where

$$|u|_{\bar{H}}^{(0)} = \max_{x \in \bar{H}} |u(x)|, \quad \langle u \rangle_{\omega\gamma, \bar{H}}^{(\gamma)} = \sup_{x, \bar{x} \in \bar{H}} (x_N^*)^{\omega\gamma} \frac{|u(x) - u(\bar{x})|}{|x - \bar{x}|^\gamma},$$

$$x_N^* = \max\{x_N, \bar{x}_N\}. \quad (8)$$

Thus $\langle u \rangle_{\omega\gamma, \bar{H}}^{(\gamma)}$ represents a weighted Hölder constant of the function $u(x)$.

We suppose that $n < m$, if n is a noninteger

$$(1 - \omega)\gamma = \gamma \left(1 - \frac{n}{m}\right) < \min(\{n\}, 1 - \{n\}), \quad (9)$$

where for a real number a , $\{a\}$ is the fractional part of a , $[a]$ is the integer part of a . This assumption is technical and it allows us, for example, to consider the functions x_N^{n-j} as elements of $C_{\omega\gamma}^\gamma(\bar{H})$ for all integers $j < n$.

In the similar way we define the Hölder seminorms with respect to each variable separately

$$\langle u \rangle_{\omega\gamma, x_i, \bar{H}}^{(\gamma)} = \sup_{x, \bar{x} \in \bar{H}} (x_N^*)^{\omega\gamma} \frac{|u(x) - u(\bar{x})|}{h^\gamma}, \quad x_N^* = \max\{x_N, \bar{x}_N\},$$

$$i = \overline{1, N}, \quad (10)$$

where $x = (x_1, \dots, x_i, \dots, x_N)$, $\bar{x} = (x_1, \dots, x_i + h, \dots, x_N)$, $h > 0$.

In the standard way we denote by $\langle u \rangle_{x, \bar{H}}^{(\gamma)}$, $\langle u \rangle_{x', \bar{H}}^{(\gamma)}$, and $\langle u \rangle_{x, \bar{H}}^{(\gamma)}$ usual unweighted Hölder seminorms with respect to each variable separately, with respect to $x' = (x_1, \dots, x_{N-1})$ or with respect to all x -variables.

Note that in terms of the Carnot-Carathéodory metric seminorm (8) is equivalent to

$$\langle u \rangle_{\omega, \bar{H}}^{(\gamma)} = \sup_{x, \bar{x} \in \bar{H}} \frac{|u(x) - u(\bar{x})|}{s(x, \bar{x})^\gamma},$$

where the Carnot-Carathéodory distance is defined as

$$s(x, \bar{x}) = \frac{|x - \bar{x}|}{|x - \bar{x}|^\omega + x_N^\omega + \bar{x}_N^\omega}. \quad (11)$$

In the case of $m = 2, n \in (0, 1)$ this was proved in [2] and in the general case we have the following theorem which is the main result of the present paper. Denote

$$[u]_s^{(\gamma)} \equiv \sup_{x, \bar{x} \in \bar{H}} \frac{|u(x) - u(\bar{x})|}{s(x, \bar{x})^\gamma}, \quad (12)$$

where $s(x, \bar{x})$ is defined in (11).

Conclusion. We have the following assertion.

Theorem. *Seminorm (10) is equivalent to seminorm (12). This means that there are constants C_1 and C_2 with the property*

$$[f]_s^{(\gamma)} \leq C_1 \langle f \rangle_{\omega, \bar{H}}^{(\gamma)} \leq C_2 [f]_s^{(\gamma)} \quad (13)$$

for any continuous in \bar{H} function $f(x)$.

References

1. Degtyarev, S.P.: On some weighted Holder spaces as a possible functional framework for the thin film equation and other parabolic equations with a degeneration at the boundary of a domain, ArXiv.org, <http://arxiv.org/abs/1507.01106>.
2. Degtyarev, S.P.: Classical solvability of multidimensional two-phase Stefan problem for degenerate parabolic equations and Schauder's estimates for a degenerate parabolic problem with dynamic boundary conditions, *Nonlinear Differential Equations and Applications (NoDEA)*. **22** (2), 185-237 (2015).
3. Knüpfer, H.: Well-posedness for the Navier slip thin-film equation in the case of partial wetting. *Comm. Pure Appl. Math.* **64** (9), 1263-1296 (2011).
4. Giacomelli, L., Knüpfer H., Otto, F.: Smooth zero-contact-angle solutions to a thin-film equation around the steady state. *J. Differential Equations* **245** (6), 1454-1506 (2008).
5. Giacomelli, L., Knüpfer H.: A free boundary problem of fourth order: classical solutions in weighted Hölder spaces. *Commun. Partial Differ. Equations*, **35** (10-12), 2059-2091 (2010).

УДК 65.011.56

НАУЧНО-ТЕХНИЧЕСКИЙ РИСК В СИСТЕМЕ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ УГОЛЬНОЙ ПРОМЫШЛЕННОСТИ

Т. В. Ванжа, Ю. К. Орлов

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
vanzha_tv7@mail.ru

Аннотация

Ванжа Т.В., Орлов Ю.К. Научно-технический риск в системе управления предприятием угольной промышленности. В работе дана детальная классификация экономических рисков, рассмотрены понятие научно-технического риска, причины его возникновения и способы предупреждения в системе управления горным предприятием.

Ключевые слова: экономический риск, методы управления, угледобывающее предприятие.

Постановка проблемы. Наиболее крупные аварии на угольных шахтах все более стали характеризоваться взрывами метана и угольной пыли, перерастающими в обширные пожары в сети горных выработок и в выработанном пространстве. Это обстоятельство побудило выполнить анализ причин возникновения и выявить возможные пути их предотвращения.

Рассмотренные в процессе анализа примеры современных крупных аварий на угольных шахтах России приводят к пониманию, что основной причиной возникновения и формирования крупных аварий являются неудовлетворительные технические решения по обеспечению промышленной безопасности угольных шахт. Скорее всего, это следствие сложившейся и бытующей хозяйственной политики, используемой в системах государственного регулирования и хозяйственного управления производством, в том числе охраной труда и промышленной безопасностью [1-2].

Классификация экономических рисков. Изучению отдельных аспектов риска специфического горного производства посвящены работы видных отечественных ученых: Астахова А.С., Ястребинского М.А., Петросова А.А., Изыгзона Н.Б., Соколова В.П., Мангуш К.С., Масыкина Б.В., Евдокимова Ф.И., Кучер В.А., Галкина А.В. Обобщив опыт отечественной научной мысли в области рисков горного производства, предлагаем их детальную классификацию, которая представлена на рисунке 1. В основу положена общепринятая систематизация экономических рисков горного производства, предложенная профессором, доктором технических наук Петросовым А.А. и кандидатом экономических наук Мангуш К.С., а также классификация основных факторов инвестиционных рисков угледобывающего предприятия, разработанная профессором, доктором экономических наук Астаховым А.С. В зависимости от природы возникновения все риски горной экономики условно разделены на две группы: макроэкономические и микроэкономические [3].



Рисунок 1 – Классификация экономических рисков горного предприятия

Научно-технический риск, его сущность, способы предупреждения.

Инновационный риск связан с вероятностью неблагоприятного исхода в результате внедрения нововведения или выпуске нового продукта, который может быть не востребован покупателем. Инновационная деятельность часто вносит значительные изменения в производственные процессы, что увеличивает неопределенность деятельности компании. Часто нововведения способствуют удешевлению способа производства, что позволяет предприятию получить кратковременные сверхприбыли, которые в дальнейшем существенно сокращаются из-за появления аналогов такой технологии [4].

Кроме того, нужно учитывать, что не всегда новый продукт с качеством, устраивающим имеющихся покупателей, можно произвести на имеющемся старом оборудовании или с привлечением имеющихся работников. Последние могут не обладать нужной квалификацией. А внедрение нового оборудования при отсутствии спроса приведет к тому, что на нем ничего нельзя будет произвести и продать его также будет сложно. Инновационные проекты часто подвержены рискам недофинансирования. Инновационные риски плотно связаны с такими специфическими видами рисков как *научно-технические*, риски (в области правового обеспечения или коммерческого предложения).

Научно-технический вариант риска может возникать в связи с отрицательными результатами научно-исследовательской работы, недочетах в опытно-конструкторских работах, в ходе которых создается комплект документации для серийного производства, изменением сроков в этапах проектирования и др.

Проблемы с правовым обеспечением появляются тогда, когда у конкурентов патентная защита появляется раньше, когда уходят в открытый доступ отдельные разработанные технические решения, когда заканчивается

срок действия лицензий на определенные виды деятельности, когда патентная защита получена слишком поздно, она недостаточно «плотная» или действует не на всех территориальных рынках.

Следует отметить, что максимально эффективное применение современных автоматизированных и роботизированных технологических комплексов при разработке месторождений на больших глубинах возможно только при совершенствовании нормативно-законодательной базы безопасности горных работ в изменяющихся условиях при соответствующем изменении структуры горных выработок и технологических процессов, позволяющих полностью реализовать потенциальные возможности новых технологий и оборудования.

В последнее время термин «инновации» стал популярным. Зачастую его употребляют в самой широкой трактовке, подразумевая под инновацией изменение продукта или технологии, до пугающей инвесторов аббревиатуры – НИОКР (Научно-исследовательские и опытно-конструкторские работы). Между тем, инновация вполне определенное понятие, под которым понимается преимущественно «новый» продукт, технология или решение, при этом успешно внедренное и обеспечивающее существенный рост эффективности процесса или производства [5].

Основные отличительные признаки инноваций, в т.ч. в горном деле, заключаются в следующем:

- принципиальная новизна техники, технологий и производственных решений;
- обеспечение существенного роста эффективности действующего производства или открытие новых направлений;
- успешное внедрение различных систем или механизмов в производство или бизнес-процессы.

В идеале, для сохранения конкурентоспособности, горное производство, нацеленное на долгосрочное существование, должно сопровождаться непрерывным поиском и внедрением инновационных решений. Хотя, зачастую инновации сталкиваются с определенными сложностями внедрения, они являются источником/резервом для существенного роста эффективности горных производств и отдельных процессов. Это особенно необходимо в периоды кризиса промышленности, когда важное направление развития принимает оптимизация производственных процессов, снижение себестоимости, повышение эффективности продаж за счёт создания новых продуктов и т.д.

Выводы. Большие финансовые затраты на развитие техники и технологии безопасного ведения работ часто становятся бесполезными при низкой культуре организации производства. Повышение уровня организации производства и эффективности управления персоналом угольных предприятий является наиболее экономичным направлением повышения безопасности

производства, которое практически не требует дополнительных финансовых ресурсов и затрат.

Основная причина крупных аварий не только и не столько в устаревшей технике и технологии производства, сколько в недопустимо низкой организации производства, в неэффективном управлении персоналом. Об этом говорит и международный опыт промышленного производства.

Как показал анализ истории крупных аварий, в военные и послевоенные годы причиной неадекватного действия персонала угольных шахт все больше была низкая квалификация работников. В последующие годы причиной неадекватного действия персонала все больше стала низкая дисциплина труда работников, обусловленная стремлением сократить объем своей работы за счет неисполнения или ненадлежащего исполнения регламентированных работ по охране труда и промышленной безопасности.

Список литературы

1. Т.В. Ванжа, Ю. К. Орлов. Разработка моделей и методов управления горным предприятием в условиях экономических рисков. Компьютерная и программная инженерия. Сборник материалов международной научно-технической конференции студентов, аспирантов и молодых ученых 15-16 декабря 2015 года. – Донецк, ДонНТУ – 2015. – 314-317 с.

2. Т.В. Ванжа, Ю. К. Орлов. Разработка моделей и методов управления горным предприятием в условиях экономических рисков. Сборник статей студенческой научно-технической конференции, г.Севастополь, 04 – 08 апреля 2016 г/ М-во образования и науки РФ, Севаст. Гос. Университет.– г. Севастополь: СевГУ, 2016. – 173-177 с.

3. Петросов А.А., Мангуш К.С. Экономические риски горного производства: Учебное пособие. М.: Издательство Московского государственного горного университета, 2002. – 142 с.

4. Балабанов И.Т. Риск-менеджмент / Балабанов И.Т. – М.: Юнити, 2002. – 192 с.

5. Райзберг Б.А. Предпринимательство и риск / Райзберг Б.А. –М.: Дело,2001.–61с

УДК 517.9

ПРИБЛИЖЕННЫЙ АНАЛИЗ ПРОЦЕССА КРИСТАЛЛИЗАЦИИ МЕТАЛЛА

А.С. Миненко, Е.В. Радевич

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
radevich_katerina@mail.ru

Аннотация

Миненко А.С., Радевич Е.В. Приближенный анализ процесса кристаллизации металла. Исследуем одну задачу Стефана с учетом конвекции в жидкой фазе. Построено приближенное решение этой задачи с использованием малого параметра. Управление процессом осуществляется с применением нечеткой логики.

Ключевые слова: функционал, кристаллизация, тепловой поток, управление

Постановка проблемы: Распространение тепла в различных средах оказывает большое влияние на характер многих важных для практики процессов. Среди задач, связанных с распространением тепла, выделяется класс задач, в которых исследуемое вещество переходит из одной фазы в другую с выделением или поглощением тепла.

Цель статьи: моделирование процесса кристаллизации металла, изучение процесса завершения получения слитка в кристаллизаторе путем его вытягивания.

Рассматривается задача управления информационными процессами при автоматизации технологий тепловой обработки металла, на основе математического моделирования, анализа статистических данных и теплофизических экспериментальных измерений. В качестве источника информации исследуется математическая модель, основанная на пространственной задаче Стефана, с учетом конвективного движения и примесей в жидкой фазе.

Постановка задачи. Пусть $D = (-1 < x < 1, y < 0)$ полуполоса, заполненная твердым металлом. Обозначим через $u(x, y)$ температуру этого металла. Требуется определить температуру $u(x, y)$ по следующим условиям:

$$u_{xx} + u_{yy} + \omega u_y = 0, (x, y) \in D, \quad (1)$$

$$u_x \pm \omega_0 u = 0, x = \pm 1, -\omega < y < 0, \quad (2)$$

$$u(x, -\infty) = 0, \quad (3)$$

$$u_y(x, 0) = v(x), -1 \leq x \leq 1, \quad (4)$$

здесь ω и ω_0 – постоянные, соответственно, число Пекле и Нуссельта. Решение задачи (1)-(4) имеет вид

$$u(x, y) = \sum_{n=0}^{\omega} \frac{\cos \lambda_n x^{\mu_n Y}}{\mu(1 + \omega_0 \frac{\cos^2 \lambda_n}{\lambda_n^2})^{-1}} \int_{-1}^1 v(\zeta) \cos \lambda_n \zeta d\zeta, \quad (5)$$

где $\mu = -\frac{\omega}{2} + \sqrt{\frac{\omega^2}{4} + \lambda_n^2}$, $n = 1, 2, 3, \dots, \lambda_n$ – положительные корни уравнения

$$\lambda = \omega_0 \operatorname{ctg} \lambda.$$

Отождествим теперь температуру $u(x, y)$ с температурой твердого слитка находящегося в кристаллизаторе при электрошлаковом переплаве. Для вытягивания слитка из кристаллизатора поверхность слитка предварительно обогревается тремя электронными лучами W1, W2 и W3, причем мощность W3 одного из них равномерно распределена в центральной зоне $\{-1 \leq x \leq 1, y = 0\}$, а два других сконцентрированы по краям $x = \pm 1$ [1]. Независимо от того, в каком отношении находится температура поверхности слитка с критической температурой T^k , при которой поверхность слитка отделяется от стенок кристаллизатора, теплообмен слитка с кристаллизатором осуществляется по формуле (2). Для получения температуры слитка достаточно положить в формуле (5) $v(x) = (W_1, W_2, W_3)$.

Далее введем в рассмотрение функционал:

$$I(v) = \int_n^0 (u(1, y) - T^k)^2 dy \quad (6)$$

Рассматривается задача. Требуется определить поток $v(x)$ из допустимого множества U , доставляющий наименьшее значение функционалу $I(v)$. Минимизирующая последовательность v_n строится по формуле (7) параметр e_n выбирается из условия (8). В качестве области определения функции U берется множество кусочно-постоянных ступенчатых функций (9):

$$v_{n+1} = v_n + e_n (v_{n-1} - v_n), \quad (7)$$

$$\min I(v_n + e_n (v_{n-1} - v_n)), 0 \leq e_n \leq 1 \quad (8)$$

$$v = v_k, x_k \leq x \leq x_{k+1}, v_k = \text{const}, k = 0, 1, 2, \dots, m. \quad (9)$$

При этом формула (5) имеет вид:

$$u(x, y) = \sum_{n=0}^{\omega} \frac{\cos \lambda_n x e^{\mu_n y}}{\mu_n (1 + \omega_0 \frac{\cos^2 \lambda_n}{\lambda_n^2})} \sum_{k=0}^m v_k \frac{\sin \lambda_n x_{k+1} - \sin \lambda_n x_k}{\lambda_n}, \quad (10)$$

$$aI(v) = I(v_0, v_1, v_2, \dots, v_m)$$

При численной реализации задачи необходимо учесть ограничение, здесь $v(x)$ – мощность потока в единицах МВт/м², а также $\omega = 2,66$, $\omega = 3,05$.

Способы решения задачи. Нулевое приближение. Найдем минимум функционала (6), в случае когда (11), по формуле (13).

$$2500 \leq v(x) \leq 5000 \quad (11)$$

$$u_0(x, y) = f_0(x, y)v \quad (12)$$

$$f_0(x, y) = 2 \frac{\cos \lambda_0 x \sin \lambda_0}{\lambda_0 \mu_0 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})} e^{\mu_0 y} \quad (13)$$

Минимум функционала (6) находим из условия (14).

$$\frac{\partial I}{\partial v} = 0. \quad (14)$$

Непосредственные вычисления показывают, что

$$v_0 = 4T^* \frac{\mu_0 \lambda_0 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})}{\sin 2\lambda_0}, \quad (15)$$

$$I(v_0) = v_0^2 \frac{\sin 2\lambda_0 (1 - e^{2\mu_0 H})}{2\lambda_0^2 \mu_0^3 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})} - (T^*)^2 H - 2T^* v_0 \frac{\sin 2\lambda_0 (1 - e^{\mu_0 H})}{\lambda_0 \mu_0^2 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})}.$$

Первое приближение. Найдем теперь минимум функционала (6), в случае (16), получим формулу (17)

$$u_1(x, y) = (f_0(x, y))v, \quad (16)$$

$$f_1(x, y) = 2 \frac{\cos \lambda_1 x e^{\mu_1 y}}{\lambda_1 (1 + \omega_0 \frac{\cos^2 \lambda_1}{\lambda_1^2})} \frac{\sin \lambda_1}{\lambda_1}. \quad (17)$$

Поступая, аналогично тому, как это было сделано в случае нулевого приближения, получим:

$$v_1 = 2T^* \left[\frac{\sin 2\lambda_0 (1 - e^{\mu_0 H})}{\lambda_0 \mu_0^2 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})} + \frac{\sin 2\lambda_2 (1 - e^{\mu_2 H})}{\lambda_1 \mu_1^2 (1 + \omega_0 \frac{\cos^2 \lambda_1}{\lambda_1^2})} \right] A,$$

$$A = \frac{\sin^2 \lambda_0 (1 - e^{\mu_0 H})}{\lambda_0 \mu_1^2 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})^2} + \frac{\sin^2 2\lambda_1 (1 - e^{2\mu_1 H})}{2\mu_1^2 (1 + \omega_1 \frac{\cos^2 \lambda_1}{\lambda_1^2})^2} +$$

$$+ 2 \frac{\sin 2\lambda_0 \sin 2\lambda_1}{\mu_0 \lambda_0 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})} \frac{(1 - e^{\mu_0 H})(1 - e^{\mu_1 H})}{\mu_0 \lambda_1 (1 + \omega_0 \frac{\cos^2 \mu}{\lambda_1^2})}.$$

Далее, имеет место следующая формула:

$$I(v_1) = v_1^2 \frac{\sin^2 2\lambda_0 (1 - e^{2\mu_0 H})}{2\mu_0^3 \lambda_0^2 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})^2} + v_1^2 \frac{\sin^2 2\lambda_1 (1 - e^{2\mu_1 H})}{2\mu_1^3 \lambda_1^2 (1 + \omega_0 \frac{\cos^2 \lambda_1}{\lambda_1^2})^2} +$$

$$+ 2v_1^2 \frac{\sin 2\lambda_0 \sin 2\lambda_1 (1 - e^{\mu_0 H})(1 - e^{\mu_0 H})}{\mu_1 \lambda_1 (1 + \omega_0 \frac{\cos^2 \lambda_1}{\lambda_1^2}) \mu_0 \lambda_0 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})} -$$

$$- 2T^* v_1 \left[\frac{\sin 2\lambda_1 (1 - e^{\mu_1 H})}{\lambda_1 \mu_1^2 (1 + \omega_0 \frac{\cos^2 \lambda_1}{\lambda_1^2})} + \frac{\sin 2\lambda_0 (1 - e^{\mu_0 H})}{\lambda_0 \mu_0^2 (1 + \omega_0 \frac{\cos^2 \lambda_0}{\lambda_0^2})} \right] - H(T^*)^2.$$

Приближение любого порядка. Аналогичным образом можно исследовать минимум функционала $I(v_n)$, когда

$$u_n(x, y) = 2 \frac{v_n \sin \lambda_0 \cos \lambda_0 x (1 - e^{\lambda_0 H})}{\lambda_0 \mu_0 [1 + \omega_0 \frac{\cos^2 \lambda_k}{\lambda_k^2}]} e^{\mu_0 y} + 2 \omega_0 v_n \sum_{k=1}^n \frac{\cos \lambda_k x \cos \lambda_k (1 - e^{\mu_k H})}{\lambda_k^2 \mu_k [1 + \omega_0 \frac{\cos^2 \lambda_k}{\lambda_k^2}]} e^{\mu_k y}.$$

Оценить погрешность предлагаемого метода вычисления минимума функционала (6) можно, используя следующее утверждение.

При достаточно малых значениях ω и при $(x, y) \in \bar{D}$ справедлива оценка:

$$\left| \sum_{k=n+1}^{\infty} \frac{\cos \lambda_k x \cos \lambda_k (1 - e^{\omega_k H})}{\lambda_k^2 \mu_k [1 + \omega_0 \frac{\cos^2 \lambda_k}{\lambda_k^2}]} e^{\mu_k y} \right| \leq \sum_{k=n+1}^{\infty} \frac{1}{(k\pi)^2} e^{\mu_k y}.$$

Справедливо также утверждение. Пусть выполнены условия (18-19), где v_0 и v_1 – некоторые постоянные. Тогда решение краевой задачи (1)-(5) удовлетворяет условиям (20).

$$\omega_0 \geq \omega \sqrt{2} \operatorname{tg} \omega \sqrt{2}, \quad 0 < \omega \leq A, \quad 0 < \omega \leq \frac{\pi^2}{16}, \quad 0 < v_0 \leq v(x) < v_1, \quad (18)$$

$$x \in [-1, 1], \quad (19)$$

$$x, y \in \bar{D}: u_y(x, y) \leq C_1 \omega \exp(\mu_0 y) \leq C_1 \omega \exp(\omega y) \quad (20)$$

$$C_0 = \exp(\mu_0 y) \leq u(x, y) \leq C_1 \exp(\mu_0 y) C_1 \exp(\omega y), \quad (21)$$

Для утверждения необходимо сравнить с помощью принципа максимума функции $u_y(x, y)$ и $v_y(x, y)$, где $v(x, y)$ – решение задачи (1)-(5).

Функция $f(x, y)$ не может принимать наименьшее отрицательное значение внутри D в силу принципа максимума. На вертикальных частях границы $x = \pm 1$ функция $f_x(x, y)$ также не может принимать отрицательный минимум. В такой точке имели бы $f_x(x, y) < 0$, между тем $f_x(x, y) = -\omega_0 f(x, y) > 0$, $x = \pm 1$, так как $f(x, y) < 0$, по предложению. На бесконечности функция $f(x, y)$ исчезает, т.е. $f(x, -\infty) = 0$. Следовательно, всюду \bar{D} справедливо неравенство $u_y(x, y) \leq v_y(x, y)$ при $(x, y) \in \bar{D}$. Отсюда с помощью интегрирования по переменной y следует оценка для функции $u(x, y)$ сверху. Аналогичным образом, можно получить оценку на производную $u_x(x, y)$ сверху

при $(x,y) \in \overline{D}$. Полученные оценки позволяют оценить температуру $u(x,y)$ и тепловой поток внутри области D не прибегая к решению задачи (1)-(4) [3-4]. Прделанные численные результаты задачи представлены в таблице 1.

Таблица 1- Численные результаты при различных значениях параметров

ω_0	λ_0	λ_1	T	H	v_0	v_1	$I(v_0)$	$I(v_1)$
1,5	0,9882	3,5422	0,9	-3,0	2,783	4,367	1,011	21,997
1,5	0,9882	3,5422	0,9	-4,0	2,783	5,730	5,428	38,576
1,5	0,9882	3,5422	0,9	-6,0	2,783	6,588	-3,038	62,490
1,5	0,9882	3,5422	0,95	-3,0	2,937	4,610	1,126	24,509
1,5	0,9882	3,5422	0,95	-4,0	2,937	5,730	-0,559	42,982
1,5	0,9882	3,5422	0,95	-6,0	2,937	6,954	-3,385	69,627
1,5	0,9882	3,5422	0,9	-3,0	4,5	4,5	16,728	23,709
1,5	0,9882	3,5422	0,9	-4,0	4,5	4,5	15,256	24,355
1,5	0,9882	3,5422	0,9	-6,0	4,5	4,5	12,397	25,275
1,5	0,9882	3,5422	0,95	-3,0	4,5	4,5	15,537	23,062
1,5	0,9882	3,5422	0,95	-4,0	4,5	4,5	13,866	23,695
1,5	0,9882	3,5422	0,95	-6,0	4,5	4,5	10,718	24,696
1,5	0,9882	3,5422	0,9	-3,0	11,1	11,1	173,683	192,101
1,5	0,882	3,5422	0,9	-4,0	11,1	11,1	175,717	198,788
1,5	0,9882	3,5422	0,9	-6,0	11,1	11,1	173,351	202,940

Выводы: в заключении необходимо отметить, что в данной статье смоделирован процесс кристаллизации металла. Для решения этой задачи был использован метод нулевого приближения, получившиеся данные в результате решения определены в таблице 1

Список литературы

1. Патон Б.Е. Избранные труды. – Киев: Институт электросварки им. Е.О. Патона НАН Украины, 2008.-893 с.
2. Шевченко А.И., Миненко А.С. Методы исследования нелинейных моделей, -Киев: Наук. Думка, 2012.-132 с.
3. Миненко А.С. Вариационные задачи со свободной границей. – Киев: Наук. Думка, 2005.-341 с.
4. Шевченко А.И., Миненко А.С., Сьпко И.А. Моделирование одного класса сложных с нечеткими управлениями // Доп. НАН Украины.- 2013. - №8. – С. 52-54.

УДК 65.011

ИССЛЕДОВАНИЕ РОЛИ КИСЛОРОДНО-КОМПРЕССОРНОГО ЦЕХА В ДЕЯТЕЛЬНОСТИ МЕТАЛЛУРГИЧЕСКОГО ЗАВОДА

В.А. Гуляев, Ю.К. Орлов

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа

E-mail: gvard12@gmail.com

Аннотация

Гуляев В.А., Орлов Ю.К. Исследование влияния работы кислородно-компрессорного цеха на деятельность металлургического завода. Разработаны организационная диаграмма предприятия, на базе которого было проведено исследование, функциональная диаграмма работы рассматриваемого подразделения. Описаны основные процессы, выполняемые подразделением. Выделены сопутствующие риски, проведен анализ ущерба и способов уменьшения.

***Ключевые слова:** кислород, риск, компрессор, блок разделения воздуха, резерв*

Введение

Работа кислородно-компрессорного цеха представляет собой одну из приоритетных задач деятельности металлургического завода. Продукция этого подразделения является одной из компонент при производстве основной продукции предприятия этого типа – чугуна и стали. Все достаточно крупные металлургические заводы обязаны иметь выделенный цех по производству кислорода, иначе закупка у сторонних производителей повысит себестоимость выпускаемой продукции и снизит её конкурентоспособность, к тому же это несет соответствующие риски срыва поставок, что в условиях непрерывного производства крайне нежелательно. Помимо обеспечения собственных нужд, излишки произведенной продукции могут продаваться третьим лицам, повышая прибыльность предприятия. Очевидно, что поддержание работоспособности этого звена производства является перспективным направлением исследования.

Целью исследования является определение места рассматриваемого подразделения в контексте общей производственной структуры предприятия, описание основных процессов, которые на нём протекают и обнаружение потенциальных рисков, которые могут повлиять на деятельность предприятия.

Задача данной работы состоит в составлении организационной диаграммы предприятия, функциональной диаграммы данного подразделения,

вербальном описании основных процессов, выделение сопутствующих рисков и их оценка.

Актуальность данного исследования заключается в необходимости поддержания работоспособности важного звена производственной цепи в условиях повышенных рисков, связанных с внешними условиями.

Организационная диаграмма предприятия

Организационная диаграмма – это схема иерархии, которая обычно используется для отображения отношений между сотрудниками, должностями и группами, она позволяет оценить роль некоторого объекта в общей системе деятельности предприятия [1]. Поскольку нецелесообразно составлять полную организационную диаграмму, включающую в себя всех сотрудников (численность которых на момент 2017 года около 4500 человек), их должности и группы, к тому же исследование в первую очередь направлено на исследование той части деятельности предприятия, которая напрямую занимается выпуском какой-либо продукции, то целесообразно составить такую организационную диаграмму, которая бы включала только ключевые производственные подразделения предприятия. На рисунке 1 показана организационная диаграмма производственной части металлургического завода на примере ПрАО «Донецксталь – металлургический завод».

На рисунке 1 можно отметить то, что кислородно-компрессорный цех не является приоритетным подразделением на предприятии, подчиняется другому подразделению, отсюда можно сделать вывод что он не является таким уж и важным для производства, однако это не так, поскольку данная диаграмма не отражает производственные связи между подразделениями, а всего лишь их место в иерархии производства. Однако, она показывает место ККЦ в общей структуре производственной части иерархии подразделений предприятия.

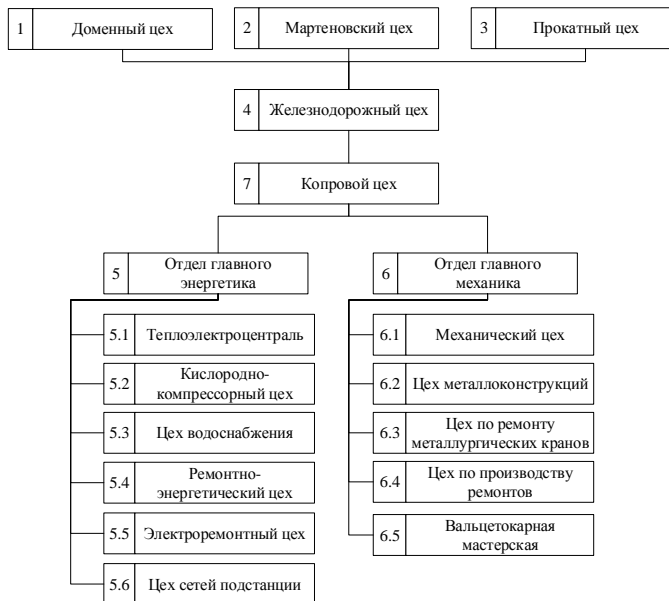


Рисунок 1 – Организационная диаграмма производственной части ПрАО «Донецксталь – металлургический завод»

Функциональная модель кислородно-компрессорного цеха

Функциональная модель предназначена для изучения особенностей работы (функционирования) системы и её назначения во взаимосвязи с внутренними и внешними элементами [1,2]. Её также часто называют IDEF0-диаграммой. На рисунке 2 показана функциональная модель рассматриваемого подразделения.

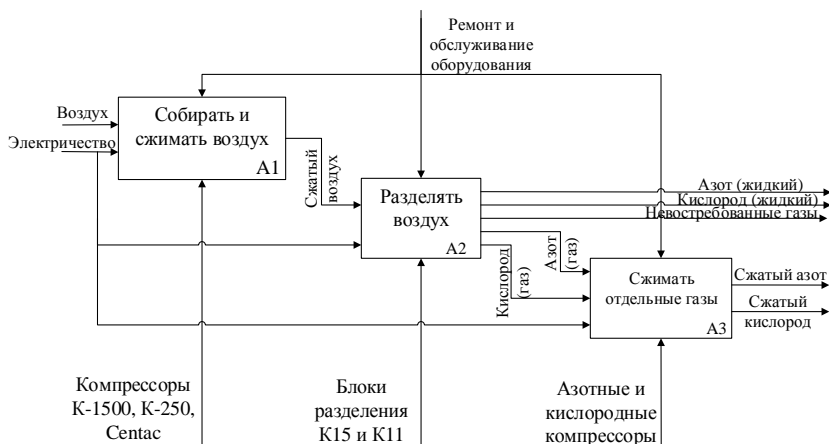


Рисунок 2 – Функциональная модель кислородно-компрессорного цеха на предприятии

На рисунке 2 видны ключевые аспекты деятельности кислородно-компрессорного цеха. Каждый из этапов производится на своём оборудовании, которое имеет свой перечень рисков. Исследование этих рисков, их описание и оценка являются приоритетными задачами данного исследования.

Вербальное описание процессов

Исследование проблем, которые могут возникнуть в процессе эксплуатации оборудования, невозможно без понимания что собственно происходит в процессе их работы, для этого необходимо показать в виде вербального описания что из себя представляет исследуемое оборудование.

Воздушный компрессор является установкой, действие которой основано на сжатии воздуха и подачи его под определенным давлением в пневматическое оборудование. Выбирая компрессорное оборудование для выполнения различных видов работ, необходимо учитывать устройство компрессора, его конструктивные особенности, а также технические и рабочие характеристики установки [4].

Конструктивные особенности, принцип действия и устройство воздушного компрессора зависят от типа установки. Современные компрессоры имеют несколько классификаций, главной из которых является различие компрессоров по принципу действия. Сегодня производители компрессорного и пневматического оборудования предлагают большое количество данных установок различного типа, наиболее распространенными среди которых являются турбинные и поршневые установки. Все виды компрессоров имеют, как общие элементы, так и различия в конструкции [4].

Установленные на предприятии воздушные компрессора К-1500 и К-250 представляют собой турбинные модели, разработанные в СССР в 50-ых годах XX века. Модель К-1500 является основной, на которую приходится большая часть нагрузки, в то же время К-250 является корректирующим, который компенсирует возможные нехватки или избытки продукции. В свою очередь турбинные компрессоры Centac были установлены и запущены в ходе модернизации производства вместо некоторой части К-1500 и представляют собой более энергоэффективное решение, чем К-1500 при сопоставимой мощности если сравнивать два компрессора Centac и один К-1500. Следует отметить, что на данный момент в работе могут участвовать один компрессор К-1500, четыре Centac и один К-250. На рисунке 3 показана модель компрессора К-250.

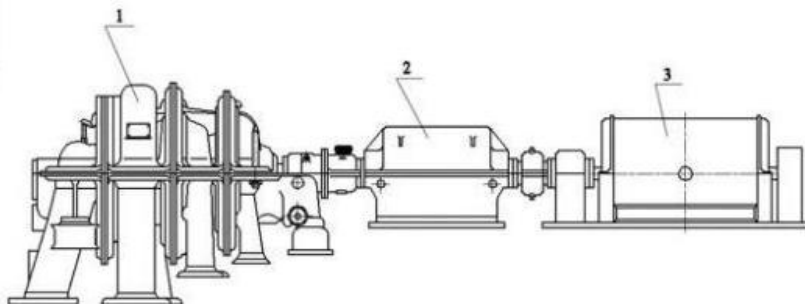


Рисунок 3 – Модель компрессора К-250

Обозначения на рисунке 3:

- 1 – компрессор;
- 2 – редуктор;
- 3 – электродвигатель.

В качестве основы для дальнейшего анализа принято считать, что другие компрессоры выглядят и работает примерно также, как и К-250.

Установка разделения воздуха – это высокопроизводительное универсальное оборудование, позволяющее для металлообрабатывающих, машиностроительных и химических производств иметь собственный жидкий азот и кислород. Установка производит так же газообразный кислород с уровнем очистки 99,7% и кислород повышенной очистки – 99,9%. Так же установка разделения воздуха может быть использована на других производственных циклах при операциях резки и сварки металлических поверхностей [4].

На рисунке 4 показана последовательность работы блока разделения воздуха.

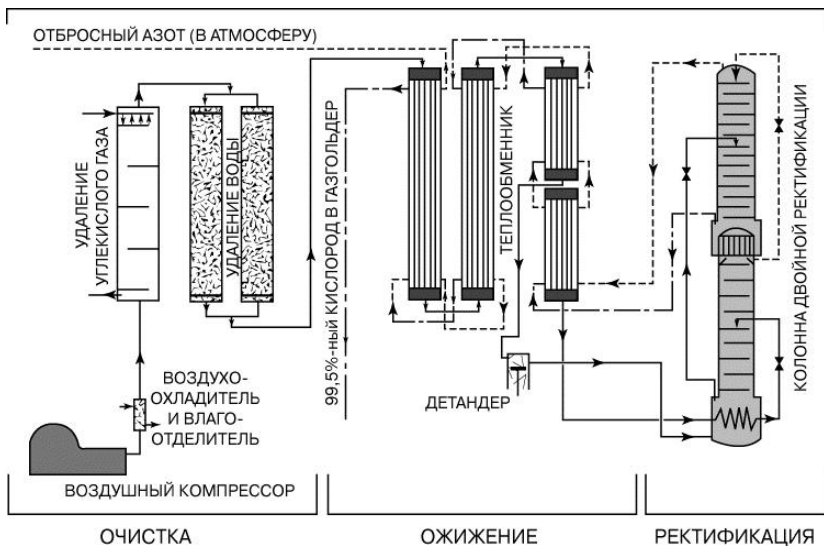


Рисунок 4 – Процесс разделения воздуха

В процессе сжатия азота и кислорода также используются компрессоры, причем для кислорода используется тот же механизм, что и для воздушных (турбинный КТК), а для азотного – поршневой ЗГП. В целом следует заметить, что основные проблемы, с которыми сталкиваются при работе обоих типов компрессоров совпадают.

Исследование сопутствующих рисков

Для начала следует отметить, что основные проблемы, которые могут возникнуть в процессе работы любого устройства обычно выносятся в приложение документации к этому оборудованию, однако практика показывает, что чаще всего некоторые описанные проблемы не встречаются в реальных условиях, а некоторые не включаются в этот список [2]. В процессе исследования была получена информация с кислородно-компрессорного цеха ПрАО «Донецксталь – металлургический завод» о наиболее часто встречаемых проблемах при работе оборудования.

При работе компрессоров были выделены следующие риски:

- а) выход температуры подшипников за пределы нормы (проблема редуктора);
- б) повышение вибрации оборудования (проблема двигателя или компрессора);
- в) нарушение противопомпажной защиты (проблема компрессора).

Ущерб, который может нанести реализация каждого из рисков, напрямую зависит от того какие действия будут предприняты для их устранения – при полной остановке оборудования простой будет составлять не менее 2 дней, в то время как при ремонте «на ходу» оценочное время ремонта составляет до полутора часов. Важно отметить, что предприятие имеет резервный компрессор на случай остановки какого-либо из рабочих, таким образом риск остановки всего производства минимизируется до статистически незначительного. Таким образом, основной риск остановки производства был резко снижен. На рисунке 5 показаны алгоритмы действий при обнаружении вышеперечисленных проблем [3].

Как можно заметить на рисунке 4, процесс разделения воздуха – очень сложная процедура, в данной работе не приводится полный анализ возможных проблем, однако существует один показатель, который характеризует правильность работы механизма – цикличность. В случае нарушения цикличности работы оборудование подлежит выявлению причин, приведших к этому, тогда ущерб в плане простоя оценивается в случае если ремонтная бригада успевает произвести ремонт за 10 часов до остановки – не более 10 часов, иначе более 6 суток, которые необходимы для перезапуска системы. Поскольку на предприятии установлены 2 блока, то в случае остановки одного из них второй не даст остановиться производству в принципе, сохраняя допустимый уровень газов для поддержания основного производства, это важно поскольку остановка печей может привести к необратимым последствиям в виде необходимости их ремонта и перезапуска.

Выводы

Таким образом, в ходе исследования было выявлено, что основной риск остановки производства был существенно снижен путём создания резервных мощностей, что, тем не менее, не затрагивает риски поломки оборудования. Были проанализированы возможные поломки оборудования и описаны те решения, которые используются на базовом предприятии, описаны варианты последствий во времени, которые могут они нести.

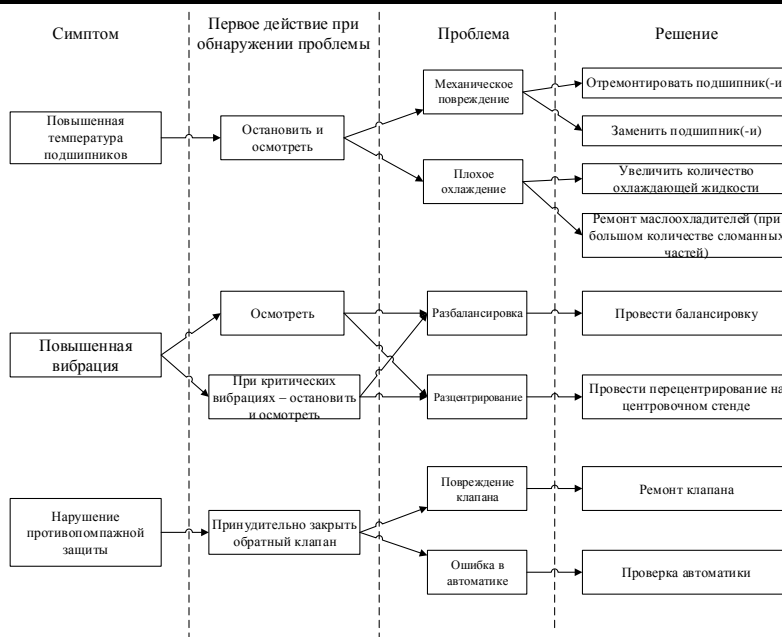


Рисунок 5 – Алгоритмы действий при обнаружении проблемных ситуаций на компрессорах

Список литературы

1. Юзов О.В. Экономика и организация производства / О.В. Юзов, Ф.И. Щетинов – М.: Металлургия, 1991. – 298 с.
2. Липсиц Н.А. Экономика, организация и планирование промышленного производства / Н.А. Липсиц – Минск. Высшая школа, 1990. – 542 с.
3. Блюмин С.Л. Модели и методы принятия решений в условиях неопределенности / С.Л. Блюмин, И.А. Шуйкова. – Липецк: ЛЭГИ, 2001. – 138 с.
4. Батицкий В.А. Автоматизация производственных процессов и АСУ ТП в горной промышленности / В.А. Батицкий, В.И. Куроедов, А.А. Рыжков. – М.: Недра, 1991. – 303 с.

УДК 004.023

РАЗРАБОТКА МОДЕЛИ ОЦЕНКИ И ПРОГНОЗИРОВАНИЯ ФАКТОРОВ ТРАВМАТИЗМА В ГОРНОМ ПРОИЗВОДСТВЕ

П.А. Гримало, Ю.К. Орлов

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
polina.insome@gmail.com

Аннотация

Гримало П.А., Орлов Ю.К. Разработка модели оценки и прогнозирования факторов травматизма в горном производстве. Приведено описание угледобывающих предприятий Донбасса, произведен анализ методов прогнозирования.

Ключевые слова: *прогнозирование, травматизм, горное производство.*

Постановка проблемы. Прогнозирование уровня травматизма является важной задачей любого промышленного предприятия и является особо актуальной задачей для горнодобывающих предприятий Донбасса. Для оценки и прогнозирования факторов травматизма необходимо выполнить следующее:

- дать характеристику угледобывающих предприятий Донбасса;
- провести анализ существующих методов прогнозирования;
- исходя из полученного анализа, выполнить прогнозирование.

Цель статьи: Провести анализ существующих методов прогнозирования и определить наиболее подходящий метод для решений поставленной проблемы.

Введение. Процесс прогнозирования на сегодняшний день все так же остается актуальным во всех сферах нашей жизни. Особо актуальной проблема прогнозирования остается в промышленности, а прогнозирование травматизма является важной задачей для любого промышленного предприятия. Прогнозирование травматизма на горнодобывающих предприятиях Донбасса и оценка его факторов позволит сохранить здоровье и жизнь многих рабочих.

Анализ угледобывающих предприятий Донбасса. Донбасс – это исторически сложившийся регион, который включает в себя Донецкую и

Луганскую области. В 1720 году был открыт Донецкий каменноугольный бассейн. Его промышленное освоение началось с конца XIX века.

Угольная промышленность – это отрасль топливной промышленности, которая включает добычу открытым способом или в шахтах, обогащение и переработку бурого и каменного угля. Донбасс – крупный центр угольной промышленности.

Наиболее крупной угледобывающей компанией Донбасса является государственное предприятие «Донецкая угольная энергетическая компания». Она создана в соответствии с приказом Министерства топлива и энергетики Украины от 18 августа 2004 года.

В состав ГП "ДУЭК" на правах обособленных подразделений входят шесть шахт и шесть подразделений, основным видом деятельности которых не является добыча угля:

- управление горнотехнических работ;
- управление информационного обеспечения и компьютерных технологий;
- управление по тушению, профилактике породных отвалов и рекультивации земель;
- управление материально-технического снабжения;
- бюро специализированных маркшейдерских работ;
- проектно-конструкторское бюро.

Все предприятия угольной промышленности Донецкой области подчинены управлению Министерства угля и энергетики, и их деятельность осуществляется по согласованию с Министром.

Основной задачей Министерства является реализация совместно с соответствующими ведомствами и органами местного самоуправления государственной политики развития отраслей топливной, нефтегазовой, угольной, энергетики; координация деятельности предприятий, учреждений и организаций государственной собственности подчиненных Министерству.

Министерство осуществляет исполнительные функции и полномочия по вопросам обеспечения деятельности топливно-энергетического, нефтегазового комплексов, энергосбережения, угольной сферы, ветроэнергетики и другие виды альтернативной энергетики. Министерство обобщает практику применения действующих нормативно-правовых актов, по вопросам, входящим в его компетенцию.

Угольная промышленность отличается особо тяжелыми и опасными условиями труда, это обуславливает высокий уровень травматизма.

Сложные горно-геологические условия разработки угольных пластов, определяют проявление различного рода природных опасных факторов при добыче угля.

Производственные системы в промышленности, и в частности в угольной отрасли, относятся к сложным техническим системам, поэтому выбор и взаимосвязанность методов управления ими представляют собой трудоемкую задачу. Для ее решения необходимо уметь количественно оценивать действие различных опасных и вредных производственных факторов.

Кроме предупреждения аварий, вызванных горно-геологическими факторами, на подземных работах особое внимание необходимо уделять средствам пожаротушения, индивидуальным и коллективным средствам защиты, следить за исправностью оборудования и измерительной техники.

Приоритетной задачей охраны труда в горнодобывающей промышленности является уменьшение уровня травматизма, внедрение и проведение профилактических мероприятий, направленных на искоренение причин ведущим к профессиональным заболеваниям.

Анализ современных методов прогнозирования. Важнейшим составным элементом методологии планирования и прогнозирования являются методологические принципы, под которыми понимаются исходные положения, основополагающие правила формирования и обоснования планов и прогнозов. Они обеспечивают целенаправленность, целостность, определенную структуру и логику разрабатываемых прогнозов [2].

На сегодняшний день существует более 200 различных методов прогнозирования. В общем методы можно разделить на две группы: интуитивные и формализованные (см. рис. 1).



Рисунок 1 – Общая классификация методов прогнозирования

Интуитивные методы прогнозирования позволяют получить оценку состояния объекта в будущем независимо от наличия информации о его

прошлом [3]. Различные методы прогнозирования, которые относят к классу интуитивных, основываются на подобию метода экспертных оценок.

Сущность метода экспертных оценок заключается в построении рациональной процедуры интуитивно-логического мышления человека в сочетании с количественными методами оценки и обработки полученных результатов. При этом обобщенное мнение экспертов принимается как решение проблемы [4]. Очевидно, что для прогнозирования травматизма на предприятиях горной промышленности основываться на мнении экспертов недостаточно, а необходимо подобрать метод, который будет учитывать уровни влияния различных факторов на показатели травматизма и основываться на показателях травматизма в прошлом.

Фактографические методы базируются на математической теории, которая обеспечивает повышение достоверности и точности прогнозов.

Эти методы моделирования можно разделить на два класса: модели предметной области и модели временных рядов. Схема классификации формализованных методов прогнозирования представлена на рисунке 2.



Рисунок 2 – Формализованные методы прогнозирования

К достоинствам регрессионных моделей прогнозирования относят простоту, гибкость, а также единообразие их анализа и проектирования.

При использовании линейных регрессионных моделей результат прогнозирования может быть получен быстрее, чем при использовании

остальных моделей. Кроме того, достоинством является прозрачность моделирования, т. е. доступность для анализа всех промежуточных вычислений.

Основным недостатком нелинейных регрессионных моделей является сложность определения вида функциональной зависимости, а также трудоемкость определения параметров модели.

Недостатками линейных регрессионных моделей являются низкая адаптивность и отсутствие способности моделирования нелинейных процессов.

Важными достоинствами авторегрессионных моделей являются их простота. Недостатками данного класса моделей являются: большое число параметров модели, идентификация которых неоднозначна и ресурсоемка; низкая адаптивность моделей, а также линейность и, как следствие, отсутствие способности моделирования нелинейных процессов, часто встречающихся на практике.

Достоинствами данного класса моделей экспоненциального сглаживания являются простота и единообразие их анализа и проектирования. Данный класс моделей чаще других используется для долгосрочного прогнозирования. Недостатком данного класса моделей прогнозирования является отсутствие гибкости.

Основным достоинством нейросетевых моделей является нелинейность, т.е. способность устанавливать нелинейные зависимости между будущими и фактическими значениями процессов.

Другими важными достоинствами являются: адаптивность, масштабируемость и единообразие их анализа и проектирования.

При этом недостатками являются отсутствие прозрачности моделирования; сложность выбора архитектуры, высокие требования к непротиворечивости обучающей выборки; сложность выбора алгоритма обучения и ресурсоемкость процесса их обучения.

Простота и единообразие анализа и проектирования на базе цепей Маркова являются достоинствами моделей на базе цепей Маркова. Недостатком данных моделей является отсутствие возможности моделирования процессов с длинной памятью.

Достоинствами моделей на базе классификационно-регрессионных деревьев являются: масштабируемость, за счет которой возможна быстрая обработка сверхбольших объемов данных; быстрота и однозначность процесса

обучения дерева, а также возможность использовать внешние факторы. Недостатками данных моделей являются неоднозначность алгоритма построения структуры дерева; сложность вопроса остановки алгоритма.

В каждом конкретном случае прогнозирования возможны различные методы и способы, каждый из которых характеризуется не одним показателем, а набором показателей, изменяющихся при изменении формулировки задачи и условий ее решения. При решении задач прогнозирования и принятии решений существенной проблемой является количество и качество требуемой информации.

На выбор метода прогнозирования влияют: сущность практической проблемы, подлежащей решению; динамические характеристики объекта прогнозирования; вид и характер имеющейся информации; комбинации факторов травматизма; требование к результатам прогнозирования.

Обоснованность выбора метода прогнозирования определяется условиями его применения и соответствия решаемым задачам

Для улучшения качества прогноза необходимо улучшить качество информации, необходимой при его разработке. Эта информация, в первую очередь, должна обладать такими свойствами, как достоверность, полнота, своевременность.

Выводы. Был произведен анализ угледобывающих предприятий Донбасса. Дана характеристика основных методов прогнозирования, описаны их достоинства и недостатки. Результаты показали, травматизм на горном предприятии сложный показатель, на значение которого влияет множество факторов и методы прогнозирования должны быть многофакторные. Модели прогноза актуальны, так как разрабатываются для принятия решений.

Список использованных источников

1. Голик А.С. Охрана труда на предприятиях угольной промышленности / А.С. Голик, В.А. Зубарева, В.А. Орурецкий – К.: Мир, 2009. – 625 с.
2. Садовникова Н.А. Анализ временных рядов и прогнозирование: учебник / Н.А. Садовникова, Р.А. Шмойлова – М.: Прогресс, 2010. – 259 с.
3. Дуброва Т.А. Статистические методы прогнозирования / Т.А. Дуброва – М.: Юнити-Дата, 2003. – 208 с.
4. Бестужев-Лада И.В. Рабочая книга по прогнозированию / И.В. Бестужев-Лада – М.: Мысль, 1997. – 130 с.

УДК 681.5.017

РАЗРАБОТКА МОДЕЛЕЙ СИСТЕМНОГО УПРАВЛЕНИЯ ПРОЦЕССА СУШКИ МАКАРОННЫХ ИЗДЕЛИЙ

Л.А. Рославцева, Ю.К. Орлов

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
lilka.roslavceva@mail.ru

Аннотация

Рославцева Л.А., Орлов Ю.К. Разработка моделей системного управления процесса сушки макаронных изделий. Выполнен анализ и оптимизация процесса сушки макаронных изделий.

Ключевые слова: макаронные изделия, сушильные установки, обезвоживание, этапы сушки изделий.

Постановка проблемы. Проблемным вопросом для макаронной промышленности остается использование традиционного, но морально устаревшего оборудования, которое на сегодняшний день экономически неэффективно и отличается повышенным энергопотреблением. Высокий уровень энергоемкости производства является актуальной проблемой, существенно ограничивающей конкурентоспособность отечественной экономики.

Цель статьи – целью настоящей работы является научное обоснование метода сушки макаронных изделий, позволяющего снизить энергопотребление по сравнению с традиционными методами (конвективным, инфракрасным) за счет применения моделей системного управления.

Анализ моделей процесса сушки и методов их получения. Тепловая обработка составляет основу многих технологических процессов. Одними из основных процессов обработки сырья в пищевой, макаронной промышленности являются тепловые процессы, связанные с обезвоживанием (сушкой), которые по своей структуре очень энергоемки и сложны. Выбранный температурный режим и способ подвода тепловой энергии оказывают непосредственное влияние на весь технологический процесс: продолжительность тепловой обработки, объемы готовой продукции и сроки ее хранения.

На сегодняшний день в макаронной промышленности в основном применяется конвективная сушка. Разработаны разновидности сушильных установок – от замкнутых камер до современных сушильных, туннельных, непрерывно действующих агрегатов, снабженных системами автоматического регулирования параметров режима сушки. Однако даже при высокой степени механизации и автоматизации этих установок процесс сушки изделий остается длительным. Известно немало исследований, посвященных проблеме интенсификации этого процесса путем повышения сушильной способности воздуха; применения новых способов сушки; терморadiационный, радиационно-конвективный, сублимационный и др.

Технология сушки макаронных изделий в первый период предусматривает возможность достаточно интенсивного процесса. Во второй период могут образоваться микротрещины, так как внутренние пласты не имеют усадки, а внешние – высыхают и усаживаются. На этой стадии нужно выбрать мягкий режим сушки. Оптимальным считается режим, при котором скорость подведения влаги из внутренних слоев равна скорости удаления влаги с поверхности изделия. При не соблюдении оптимальных режимов сушки, интенсивном удалении влаги может произойти слипание или деформация продукта, растрескивание, закисание изделий.

На макаронных предприятиях применяют конвективный способ сушки макаронной продукции, т.е. обдувание высушиваемого продукта сушильным агентом, в качестве которого чаще всего используют нагретый воздух.

Е.С. Поляков предложил конструкцию шкафной сушилки закрытого типа специально для сушки длинных макаронных изделий [1]. Сушилка оборудована двумя отсеками закрытого типа, куда закатываются бастуны с подвешенными на них изделиями. Недостатками такого способа является длительный процесс сушки и неравномерное удаление влаги с поверхности макаронных изделий, что приводит к сильному искривлению изделий во время сушки.

Сушка в жестких режимах приводит к растрескиванию. Исходя из этого, при современных режимах сушки макарон выявлены две технологические стадии, правильное проведение которых позволяет получать готовые изделия высокого качества. Первая – предварительная сушка, которая характеризуется большой интенсивностью и практически постоянной скоростью сушки. Второй период – период окончательной сушки, здесь предъявляются особые требования к технологическим параметрам режимов сушки, на выходе процесса должны быть строго обеспечены параметры продукта, предусмотренного нормативными документами, в частности финальная влажность продукта должна находиться в диапазоне 12 – 12,5 процентов.

В настоящее время используются различные варианты высокотемпературных режимов сушки. Однако главная предпосылка их разработки заключается в том, что повышение температуры сушки

макаронных изделий должно сопровождаться повышением относительной влажности воздуха. Высокотемпературная сушка позволяет снизить расход энергии за счет уменьшения продолжительности процесса и сократить производственные площади на единицу вырабатываемой продукции, значительно улучшить её микробиологическое состояние и при правильно подобранных параметрах сушки улучшить цвет и варочные свойства изделий.

Говоря о преимуществах высокотемпературной сушки перед низкотемпературной необходимо иметь ввиду, что чем ниже температура

сушки, тем в меньшей степени возникает опасность возникновения в изделиях чрезмерных напряжений сдвига в результате тех или иных нарушений в заданном режиме. Поэтому применение ВТ режимов возможно только на оборудовании, оснащённом автоматизированными системами контроля и регулирования заданного режима.

Методология оптимизации параметров процесса сушки и сушильного оборудования. Патентованный способ сушки макаронных изделий содержит 4 этапа. Согласно данному способу для оптимизации и дальнейшей автоматизации технологического процесса сушки была составлена обобщенная схема управления процессом с поэтапной зависимостью выходных параметров (целевых функций) от управляющих воздействий (рисунок 1).

На всех этапах в качестве контролируемых параметров целесообразно использовать экспертную оценку органолептических свойств макаронных изделий, комплексный показатель качества и удельные затраты энергии на проведение этапа. Причем для каждого этапа необходимо получить низкие удельные затраты и высокую экспертную оценку органолептических свойств.

Комплексный показатель качества должен увеличиваться от этапа к этапу таким образом, чтобы после завершения последнего достигнуть максимального значения. С целью создания оптимальных условий для проведения последующих этапов контролируется также исходное относительное удлинение изделий при растяжении.

Самым значимым с точки зрения повышения прочности макаронных изделий в процессе сушки, и как следствие обеспечения возможности сушки макаронных изделий в жестких режимах, является первый этап. В соответствии с вышесказанным на данном этапе осуществляется пропаривание и разогрев макаронных изделий.

Этап сопровождается разогревом макаронных изделий, вызывающим возникновением, так называемого, градиента влажности, который направляет влагу от более нагретых слоев к менее нагретым.

Цель этапа – как можно более быстрое разогревание изделий с параллельным упрочнением их клейковинного и крахмального каркаса за счет пропаривания всех слоев.

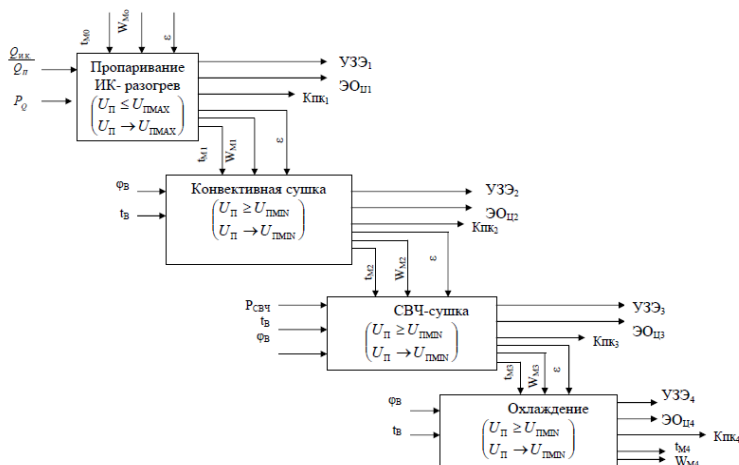


Рисунок 1 – Обобщенная схема управления процессом сушки

В качестве управляемого параметра целесообразно использовать изменение относительного удлинения при растяжении в процессе, об окончании этапа можно судить по достижению относительного удлинения при растяжении максимального значения. Отношение количества энергии, вносимой с инфракрасным излучением к количеству энергии, вносимой с паром и интенсивность подвода энергии следует принять за управляющие воздействия.

Не менее значимым для производства макаронных изделий, является этап в процессе осуществления, которого полуфабрикат можно сушить в жестких режимах, а как следствие, с наименьшими энергозатратами. Данный этап целесообразно проводить с использованием конвективного способа сушки. Следует отметить, что жесткие режимы сушки можно применять до потери полуфабрикатом пластических свойств, а именно до достижения высушиваемыми макаронными изделиями упруго-вязкого состояния.

Цель этапа – максимально быстрое снижение влажности с одновременным упрочнением их структуры. В качестве управляющих воздействий целесообразно использовать температуру (t_B) и относительную влажность сушильного воздуха (φ), а в качестве управляемого параметра целесообразно использовать изменение относительного удлинения при растяжении.

Об окончании этапа можно судить по началу отклонения уменьшения относительного удлинения при растяжении. Интенсивность сушки

определяется разницей между средневзвешенной влажностью изделий и влажностью поверхностных слоев.

На третьем этапе – этапе сверхвысококачественной сушки наблюдается высушивание макаронных изделий до превращения их в упругий материал.

Цель этапа – быстрое удаление влаги из изделий с одновременным предупреждением возникновения больших касательных напряжений, вызванных смещением слоев. В качестве управляемого параметра также можно использовать относительное удлинение при растяжении.

В качестве управляющих воздействий – интенсивность сверхвысококачественного воздействия и изменение влажности поверхности изделий (U_n), определяющейся параметрами теплоносителя. Об окончании этапа можно судить по прекращению изменения относительного удлинения.

Интенсивность сушки определяется сверхвысококачественной мощностью – воздействия, также как при конвективной сушке, относительной влажностью и температурой сушильного воздуха.

На четвертом этапе – этапе охлаждения происходит доведение макаронных изделий до стандартной влажности. Цель этапа – как можно более быстрое доведение продукта до кондиционной влажности с одновременным охлаждением.

Об окончании этапа следует судить по достижению изделиями требуемой влажности и температуры. В качестве управляемых параметров целесообразно использовать изменение температуры и влажности макаронных изделий.

Температура изделий на этапе охлаждения практически равна температуре сушильного воздуха, влажность изделий определяется изменением средневзвешенной влажности. В качестве управляющих воздействий целесообразно использовать температуру (t_b) и относительную влажность охлаждающего воздуха (ϕ).

Таким образом, управление процессом сушки макаронных изделий носит дискретный характер и осуществляется без учета времени воздействия на продукт, временные изменения улавливаются за счет того, что процесс разделен на этапы [2].

На каждом этапе производится проверка на соответствие оптимальной области значений управляющих параметров. Задачу управления процессом сушки макаронных изделий можно рассматривать как задачу поддержания управляющих параметров в оптимальной области. На основании обобщенной схемы управления процессом сушки может быть предложена следующая схема (рисунок 2).

Основными элементами схемы являются: исполнительный механизм 1-го управляющего параметра, различного для каждого этапа сушки (ИМ1), исполнительный механизм 2-го управляющего параметра (ИМ2), регулирующие органы 1-го и 2-го параметров (РО1, РО2), датчики управляющих воздействий для 1-го и 2-го параметров (Д1, Д2), блок

адаптации к оптимальным значениям параметров, являющийся системой настройки. Блок адаптации предполагает перевод аналоговых сигналов в цифровой код, вычисление оптимальных значений управляющих воздействий для каждого этапа сушки и перевод их в аналоговый сигнал.

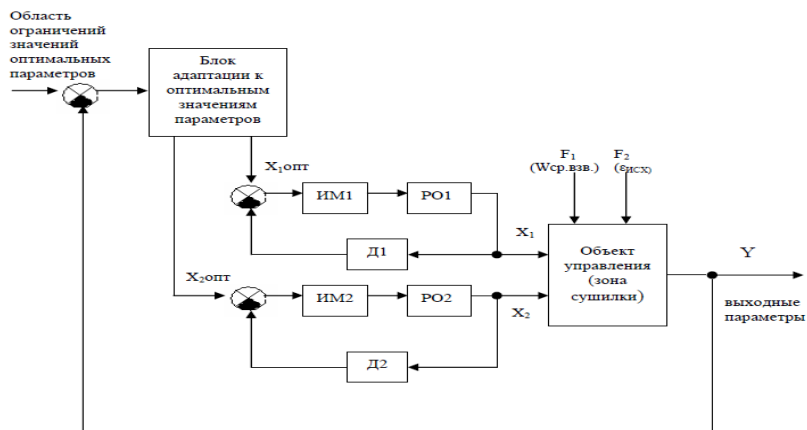


Рисунок 2 – Структурная схема управления процессом сушки макаронных изделий

Выводы: произведен анализ математической модели процесса сушки макаронных изделий, на основании которой процесс полностью автоматизируется. Основные функции, выполняемые подобной системой, включают в себя контроль и управление, обмен данными, обработку накопление и хранение информации, формирование сигналов отклонения от нормального функционирования. Экономические преимущества использования автоматических систем в производстве вытекают из их технических преимуществ, а именно более экономичное использование труда, материалов и энергии.

Список литературы

1. Буров, Л.А. Технологическое оборудование макаронных предприятий / Л.А. Буров, Г.М. Медведев. – М.: Пищевая промышленность, 1980. – 246 с.;
2. Грачев, Ю.П. Математические методы планирования эксперимента / Ю.П. Грачев, Ю.М. Плаксин. – М.: ДеЛи принт, 2005. – 296 с.

УДК 61.001.12/.18

СОВРЕМЕННЫЕ МЕДИЦИНСКИЕ ИНФОРМАЦИОННО-СПРАВОЧНЫЕ СИСТЕМЫ

А.С. Бровкина, Ю.К. Орлов

Донецкий национальный технический университет
кафедра искусственный интеллект и системный анализ
annbrunya@gmail.com

Аннотация

Бровкина А.С., Орлов Ю.К. Современные медицинские информационно-справочные системы. *Данные системы широко используются и разрабатываются во всем мире. Главной особенностью систем является высокая степень адаптации под современные технологии, смартфоны, а также возможность связи баз данных между лечебными заведениями, использующими подобные информационные системы.*

Ключевые слова: *медицинские информационные системы, база данных, медицинские девайсы, автономные роботы, двухмерный штрихкод.*

Постановка проблемы. Здравоохранение выходит за рамки больниц и смещается в сторону самопознания пациента и расширения его прав и возможностей. Интернет привел к тому (плохо это или хорошо), что много людей сами исследуют свои симптомы, ставят диагнозы и лечат себя. Хотя следует избегать крайностей, ведь загнать джина обратно в бутылку с точки зрения самообучения пациентов уже невозможно. Медицинские услуги на дому и инновационные технологии обеспечивают связь между врачами и пациентами, что ранее было невозможным, сохраняя как их жизни, так и деньги.

Цель статьи – провести анализ современных разработок для медицинских информационных систем и изучение проблем, связанных с их внедрением.

Автономные роботы. Мониторинг пациента до, во время и после процедуры может теперь привлекать автономных роботов, таких как iRobot's RP-VITA. В 2012 году вышел медицинский робот телеприсутствия компании RP-VITA, которым можно управлять с помощью iPad.. RP-VITA (уже установлен в шести больницах в США) способен автономно путешествовать в определенную палату или даже следовать за пациентом, прокладывая маршрут в зависимости от препятствий. Врачи могут общаться с пациентами лицом к

лицу, устанавливая диагноз с помощью веб-камеры, и даже использовать специальное оборудование. Облегчение предоставления медицинской информации поставщикам медицинских услуг и отдельным лицам по широкому спектру устройств от компьютеров до планшетов и до мобильных телефонов.

Технологии позволяют нам измерять основные параметры здоровья удобным и недорогим способом. Электронная одежда в сочетании с датчиками будет использоваться для сбора таких данных (см рис. 1).

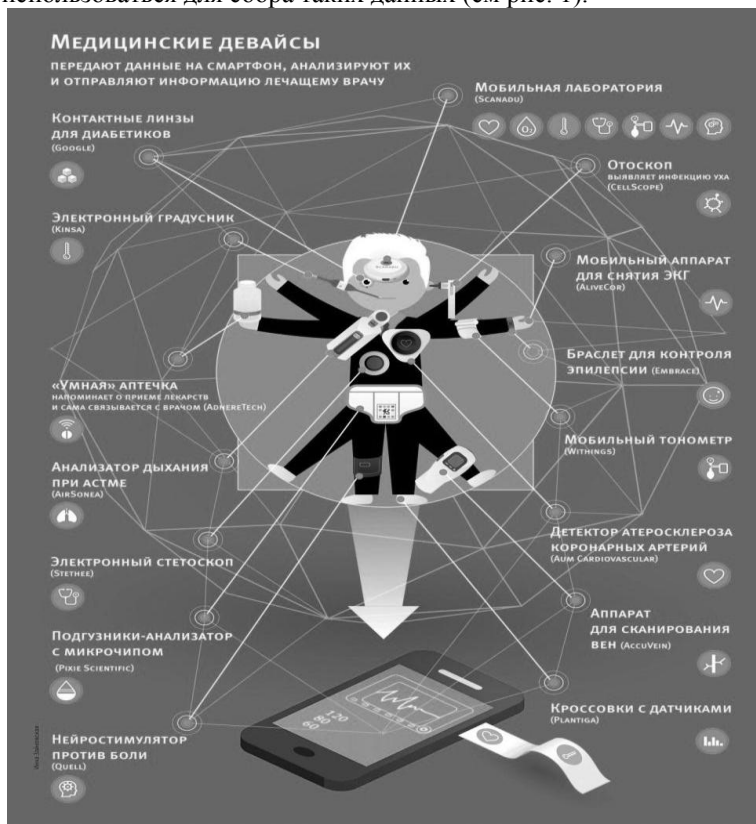


Рисунок 1 – Медицинские девайсы

Сообщества биолобораторий появляются по всему миру, объединяя изобретателей, любителей и всех, кому интересно экспериментировать с оборудованием и знаниями.

Проблема внедрения и использования медицинских информационных систем (МИС) актуальна и находится в стадии исследования и развития.

Появление разного рода медицинских инноваций приводит и к зарождению новых способов оказания медицинских услуг. К ним, например, относится и так называемая интегрированная медицина. Она подразумевает удаленное лечение пациентов, что избавляет последних от необходимости лично идти в поликлинику. Вместо этого человек сам будет следить за своим самочувствием с помощью различных гаджетов и отправлять их показания вместе с описанием своих жалоб врачу. Специалисту останется только изучить их и прописать соответствующее лечение. Никаких затрат по времени, утомительных ожиданий в очереди, все довольны и счастливы, по крайней мере, в теории.

Обмен электронными медицинскими записями МЕЖДУ лечебными заведениями. Эксперты прогнозируют, что в 2017 году возможность связи между различными базами медицинских данных будет осуществлена. Так, планируется запуск инструмента, обеспечивающего совместимость между различными ресурсами в сфере здравоохранения, – Fast Healthcare Interoperability Resources, FHIR. FHIR основывается на предыдущих стандартах формата данных от HL7, таких как HL7 version 2.x и HL7 версии 3.x. Но его проще реализовать, так как в нем содержится современный веб-пакет API-технологий, в том числе HTTP-протокол RESTful, HTML и каскадные таблицы стилей для интеграции пользовательского интерфейса, выбор JSON или XML для предоставления данных и Atom для результатов [1].



Медицинская информационная система города Донецка

Рисунок 2 – Медицинская информационная система города Донецка

FHIR предоставляет альтернативу подходам, ориентированным на документы, путем непосредственного предоставления отдельных элементов данных в виде сервисов. Например, базовые элементы здравоохранения, такие как пациенты, госпитализация, диагностические отчеты и лекарства, могут извлекаться и обрабатываться через собственные URL-адреса ресурсов. FHIR была поддержана на собрании Американской ассоциации медицинской информатики такими компаниями, как Cerner, которые ценят ее открытый и расширяемый характер [2]. В июле 2016 года на круглом столе FHIR Applications было продемонстрировано множество приложений [3]. Профиль Sync for Science (S4S) основывается на FHIR, чтобы помочь медицинским исследованиям спросить (и, если одобрено пациентом, получить) пациента – уровень электронных медицинских записей [4]. Соглашение о сотрудничестве с Консорциумом платформ здравоохранения было объявлено 2017 году.

Использование QR-кодов в медицине. QR-код – это двухмерный штрихкод (бар-код), предоставляющий информацию для быстрого ее распознавания с помощью камеры на мобильном телефоне. С появлением достаточно мощных мобильных телефонов, оснащенных встроенными камерами, QR-коды начали своё широкое распространение по всему миру (см. рис. 3).

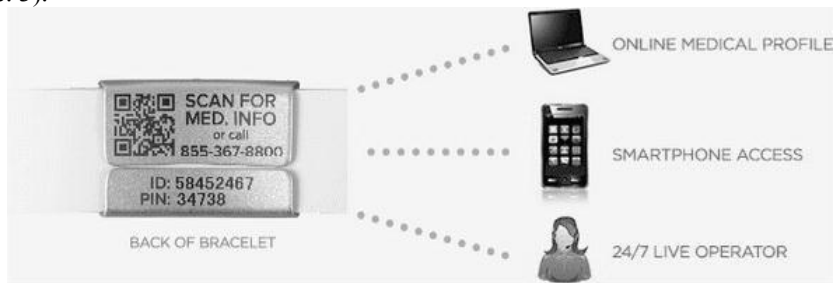


Рисунок 3 – Браслет с использованием QR-кода

С их помощью можно закодировать любую информацию, например: текст, номер телефона, ссылку на сайт или визитную карточку. QR-коды используются во всех сферах жизнедеятельности человека, помогая ему получать быстрый доступ к интересующей его информации, для удобства совершения покупок и оплаты банковских квитанций. В последнее время QR-коды стали часто использоваться и в медицинских целях. Некоторые клиники используют их для хранения и дальнейшего использования медицинской информации в цифровом виде. Эта услуга поможет избежать утери важных медицинских данных о пациенте и позволит получить быстрый, своевременный доступ к жизненно необходимой информации о больном. QR-код также может служить в качестве современного инструмента для предоставления счетов за медицинское обслуживание в клинике.

Такой двухмерный штрих код может помочь в экстренной ситуации спасти жизнь человека. В нем может быть зашифрованная информация о группе крови человека, его заболеваниях и аллергии на определенные медикаменты, а также другие медицинские особенности владельца современного аксессуара.

Внедрение МИС. Компания Philips попыталась ответить на вопрос о том, как обстоит дело на практике, проведя масштабное исследование «Индекс здоровья будущего». Точнее говоря, специалисты провели опрос среди медиков и пациентов по всему миру, чтобы понять их отношение и готовность к новым способам лечения. В исследовании приняло участие 2600 врачей разного профиля и 13 тысяч пациентов из 13 стран мира. Результаты оказались довольно пессимистичными. Так, лишь четверть опрошенных пациентов согласна отсылать данные своих медицинских гаджетов врачам, при этом более половины из них (57%) пользуется подобными приборами постоянно. Не остались в долгу и медики. 46% опрошенных врачей заявило, что не верит в способность своих пациентов самостоятельно следить за своим здоровьем. Кроме того, 74% опрошенных пациентов выказало недовольство тем фактом, что при посещении разных врачей им приходится заново повторять одну и ту же информацию о состоянии своего здоровья (и это при том, что в США и Европе, например, налажена довольно оперативная система сбора и обработки персональных медицинских данных) [5].

Выводы исследования Philips оказались совершенно неожиданными. Авторы заключили, что развивающиеся страны гораздо активнее и эффективнее внедряют новые медицинские технологии, чем развитые. Впрочем, объясняется все просто. У развивающихся государств есть преимущество – опыт их более развитых соседей. Это позволяет им избежать многих ошибок.

Компании «Яндекс» и «Новая медицина» запустили совместный проект – сервис для онлайн-консультаций с врачом. Его технической поддержкой, что логично, займется «Яндекс», а «Новая медицина», которая, кстати, является оператором ДОС+ (приложение для вызова врача на дом), возьмется за предоставление самой услуги. Сервис является частью приложения «Яндекс.Здоровье» и уже доступен для использования. На данный момент он позволяет связаться только с педиатром и терапевтом. Возможность получить удаленную консультацию от врачей других специализаций появится позже, обещают разработчики. Стоимость одной консультации составит 499 рублей. Правда, оплатить ее можно только банковской картой при условии, что она выпущена в России. Также для оплаты подойдут украинские, белорусские и казахстанские кредитки [6].

Выводы. Диагностические процедуры смещаются в сторону портативных устройств и будут выполняться из дома. Медицинские мобильные приложения начнут устанавливаться в зависимости от потребностей. Дешевые технологии

и дух изобретателей формируют новое поколение ученых и инженеров, которые не видят ограничений в исследованиях. Уже в этом году могут появиться небольшие устройства (размером с небольшой мобильный телефон), которые смогут собирать информацию о состоянии вашего здоровья. В частности, замерять артериальное давление, уровень глюкозы (без прокола) и множества других параметров, которые могут указать на начало того или иного заболевания. Внедрение медицинских информационных систем существенно облегчит работу лечебных заведений и сделает медицинскую информацию и помощь более доступной, больницы современными [7].

Список литературы:

1 HL7 FHIR: Fast Healthcare Interoperability Resources [электронный ресурс] Хабрахабр URL: <https://habrahabr.ru/post/260145>

2 Дэвид Ратс «Смесь приложений на демонстрации, чтобы продемонстрировать потенциал FHIR» [электронный ресурс] // Блоги по вопросам здравоохранения URL: www.healthcare-informatics.com

3 Разработка динамических REST-сервисов на документо-ориентированной БД Bagri [электронный ресурс] Хабрахабр URL: <https://habrahabr.ru/post/315392/>

4 «HL7 Teams with Healthcare Services Platform Consortium on FHIR Development | Healthcare Informatics Magazine | Health IT | Information Technology» [электронный ресурс]. [2017]. URL: www.healthcare-informatics.com

5 Опрос: насколько мир готов к внедрению интегрированной медицины? [электронный ресурс] // [2017] URL: <https://mhealthcongress.ru/ru/news/opros-naskolko-mir-gotov-k-vnedreniyu-integrirovannoy-meditsini-66995>

6 В «Яндексе» теперь тоже можно проконсультироваться с врачом [электронный ресурс] [2017] URL: <https://mhealthcongress.ru/ru/news/v-yandekse-teper-toge-mogno-prokonsultirovatsya-s-vrachom-66970>

7. Бровкина А.С. «Синтез информационно-справочной системы диагностирования и лечения болезней для медицинских заведений» / А. С. Бровкина, Ю. К. Орлов – ДонНТУ, Информационные управляющие системы и компьютерный мониторинг (ИУСКМ–15), Сборник материалов VI Международной научно-технической конференции студентов, аспирантов и молодых ученых, Донецк, 2015, - 230 – 235 с.

УДК 681.518.5

МОДЕРНИЗАЦИЯ СИСТЕМЫ АВТОМАТИЧЕСКОГО РОЗЛИВА НА ПРЕДПРИЯТИЯХ ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ

А.В. Дробитько, И.А. Тарасова

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
alexandra.drobitko@gmail.com

Аннотация

Дробитько А.В., Тарасова И.А. Модернизация системы автоматического розлива на предприятиях пищевой промышленности. Произведено устранение ряда конструктивных недостатков системы автоматического управления розливом за счет подбора микропроцессорного узла, а также проектирования мест расположения и выбора датчиков.

***Ключевые слова:** система автоматического управления технологическим процессом, структурная схема, модернизация, микропроцессорный узел.*

Постановка проблемы. Пищевая промышленность является социально значимой отраслью, производящей продукты питания. Состояние пищевой промышленности ДНР характеризуется наличием нерешенных проблем, узких мест, препятствующих развитию её конкурентного потенциала. Остается высокой доля старых пищевых предприятий с низким технико-технологическим уровнем производства, не требующих высокой квалификации персонала. Рост цен на сырьё, а также низкая платежеспособность населения Донецкой Народной Республики поставили на грань выживания многие пищевые производства. Как следствие, конкурентный потенциал формируется крайне медленно не только между отдельными подотраслями, но и внутри них. Лишь отдельные крупные предприятия обладают конкурентным потенциалом, достаточным для устойчивого развития [1].

Ключевым фактором успеха для компании, занимающейся производством жидких продуктов, будь то питьевая и минеральная вода, алкоголь, газированные напитки, подсолнечное масло и технические жидкости, является обеспечение максимально высокого уровня качества данной продукции. Не последнюю роль здесь играет уровень используемых при производстве технологий, а также автоматизация процессов.

Современная линия розлива должна не только обеспечивать бесперебойную работу, но также предлагать конкурентные преимущества в

области снижения расходов на производство, а также обеспечения высокого уровня качества производимой продукции [3].

С целью достижения успеха в конкурентной борьбе производители должны постоянно улучшать качество продукции и, по возможности, снижать цену. Эту задачу можно решать разными способами, например – совершенствованием технологии производства и оборудования. Одним из эффективных способов повышения качества продукции, доступных для системных интеграторов, а также собственных служб автоматизированных систем управления технологическим процессом предприятий, является более полная автоматизация всего производственного процесса. При этом использование современного оборудования существенно повысит надежность работы и сократит время простоев по причине отказа оборудования [2].

Цель статьи – устранение ряда конструктивных недостатков системы автоматического управления розливом на предприятиях пищевой промышленности за счет модернизации линии.

Для достижения поставленной цели необходимо осуществить анализ структуры автоматической линии розлива, а также её системы управления для выявления конструктивных недостатков. Устранение выявленных недостатков возможно при подборе микропроцессорного управляющего узла, а также проектировании мест расположения и выбора датчиков на линии автоматического розлива. Для работы была выбрана линия «РОЗМА 3000», так как она наиболее распространена на предприятиях с большим объемом выпуска продукции.

Анализ автоматической линии розлива. В ходе анализа была разработана структурная схема рассматриваемой автоматической системы розлива «РОЗМА 3000», представленная на рисунке 1.

Блок выдува А-3000 №24 предназначается для производства ПЭТ – бутылок из разогретых преформ методом вытяжки и двухстадийного выдува в трехместной пресс-форме. На оборудовании можно изготовить бутылки емкостью от 0,25 до 2,0 литров, производительностью 4000 бутылок в час.

Готовый блок проталкивает по столу вновь образованным блоком на транспортер термоусадочного модуля, в камере которого под действием горячего воздуха происходит усадка пленки.

После прохождения камеры блок обдувается вентилятором, пленка остывает и обтягивает блок, создавая упаковку, готовую к дальнейшей транспортировке.

Так же используется рольганг-накопитель, исключая повреждения упаковки при переносе неостывшего блока вручную.

Автомат состоит из силовой установки, печи для разогрева преформ, автозагрузчика преформ, контейнера с терминалом управления.

Существенным недостатком этого блока является высокая трудоемкость обслуживания и ремонта, рентабельность при выпуске не более 4000 бутылок в час.

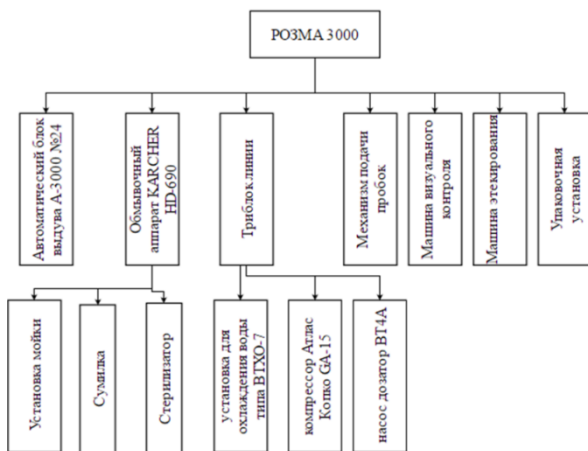


Рисунок 1 – Структурная схема автоматической системы розлива

Мощный аппарат высокого давления HD-690 с трехфазным двигателем без подогрева воды идеально подходит для коммерческого производства. Его недостатками являются только громоздкость и зависимость от оператора.

Установки охлаждения жидкости серии ВТХР специально разработаны и применяются для охлаждения воды (или других пищевых жидкостей), используемой при производстве продуктов питания, в случае, если ее расход через установку изменяется в циклическом режиме – от максимального до нуля. Но получение воды с нулевой температурой в пластинчатых или кожухотрубных теплообменниках сопряжено с риском ее замерзания и, соответственно, выходом из строя теплообменного оборудования.

Серия компрессоров GA 15 VSD является приемлемым решением для производства с переменной потребностью в сжатом воздухе. Посредством контроля давления на выходе техническое решение на основе регулируемой скорости привода (VSD) непрерывно изменяет подачу воздуха в соответствии с потребностью. Экономия энергии более 35 процентов воплощается в реальность благодаря динамическому диапазону регулирования и новому энергосберегающему циклу вентилятора [9]. Однако у данной серии есть значительный недостаток, у неё отсутствует система сбора диоксида углерода из канала обратного газа.

Насос-дозатор ВТ4А – это насос, который предназначается для точной дозировки заданного объема продукта.

Выбор микропроцессорного узла. Для устранения недостатков необходимо выбрать такой регулятор, конструктивно-техническое оформление которого удовлетворяло бы требованиям надёжности, работоспособности применительно к конкретным условиям производства

Гидравлический регулятор не подходит для данной системы, так как его недостатками является: небольшой радиус действия, огнеопасность, зависимость рабочих характеристик от температуры, что крайне недопустимо.

Пневматические регуляторы так же не могут быть применены в связи с неудовлетворением требования пожаровзрывобезопасности и необходимости наличия сжатого воздуха соответствующего качества, а также сравнительно небольшой протяженности импульсных и командных линий.

Электрические регуляторы, наиболее широко распространенные в пожаровзрывоопасных помещениях, удовлетворяют требованиям рассматриваемого объекта. Положительной стороной этих регуляторов является то, что велика протяжённость командных и импульсных линий (более 300 м).

Выбор типа регулятора будет основываться на выборе наиболее дешевого и простого в эксплуатации регулятора, обеспечивающего при различных возмущениях в заданных пределах динамическую ошибку, время регулирования и статическую ошибку.

В промышленности наиболее часто применяют регуляторы непрерывного действия (И-, П-, ПИ- и ПИД-регуляторы).

И-регуляторы применяют на объектах, имеющих значительное самовыравнивание и малое время запаздывания. Для регулирования технологических величин на нейтральных объектах И-регуляторы не применяют, т.к. такие системы неустойчивы при любых значениях настроечных параметров.

П-регуляторы обладают быстрым действием и работоспособны на инерционных объектах, но могут быть использованы лишь тогда, когда при изменении нагрузки объекта допустимо остаточное отклонение параметра от заданного значения.

ПИ-регуляторы имеют достаточное быстрое действие, кроме того, способны выводить параметр на заданное значение, поэтому их применяют чаще других. Если объекты характеризуются большим запаздыванием и претерпевают значительные изменения нагрузки, П-и ПИ-регуляторы могут не справиться с обеспечением требуемого качества регулирования; в таких случаях используют ПИД-регуляторы [5].

Этапы модификации системы розлива. В работе выделены четыре основных этапа процесса модернизации.

Первый этап. Для данной системы, внедрить в линию ЭВМ с ПИД-контроллером ТРА-1200, данный контроллер наиболее подходит линии «РОЗМА 3000», так как при относительно низкой стоимости обладает

широкими функциональными возможностями и высокой производительностью [4]. В следствии этого электро- и термклапаны системы станут наиболее устойчивые, а это в свою очередь позволит регулировать температуру автоматического блока выдува А-3000, что приведет к снижению брака.

Второй этап. Модернизация подачи газа CO₂ на обмывочный аппарат KARCHERHD-690. Изначально, на данном аппарате подача газа регулировалась оператором вручную с помощью редуктора. После модернизации регулирование будет происходить с помощью клапана. Данный клапан будет работать под управлением контроллера ТРА-1200.

Третий этап. Внедрение системы сбора диоксида углерода из канала обратного газа компрессора Атлас GA-15. Датчики и клапаны данной системы также работают от контроллера ТРА-1200. Данный этап значительно снизит расход продукции и повысит ее качество.

Четвертый этап. Для точного нанесения этикетки в этикирующую машину могут быть встроены система донной ориентации бутылки, новый копир и комплект столиков с донной ориентацией, а также установлены 3 аппликатора AveryDennison с синхронизацией скорости. Но для этого необходимо провести полное обновление электрооборудования этикирующей машины и оснащение её всеми необходимыми датчиками и блокаторами, которые обеспечивают беспроблемную автоматическую работу, в частности, это датчики наличия/отсутствия бутылки и этикетки, датчики обрыва этикеточной ленты и т.д.

Выводы. Таким образом, в работе рассмотрены наиболее рациональные и современные методы модернизации автоматической системы розлива и из них выбраны наиболее подходящие для линии «РОЗМА 3000». Результаты показали, что модернизация позволила упростить работу оператора, снизить расходы и увеличить качество продукции, что в свою очередь обеспечит бесперебойную работу и конкурентное преимущество.

Список литературы

1. Балашов В.Е. Техника и технология производства пива и безалкогольных напитков. / В.Е. Балашов, В.В Рудольф – М.: Легкая и пищевая промышленность, 1981. – 248 с.
2. Ермолаева Г.А. Технология и оборудование производства пива и безалкогольных напитков: Учеб. для нач. проф. образования. / Г.А. Ермолаева, Р.А. Колчева – М.: ИРПО; Изд. центр «Академия», 2000. – 416 с.
3. Перегудов Ф.И. Основы системного анализа. 3-е изд., исп. и доп. / Ф.И. Перегудов, Ф.П Тарасенко Томск: НТЛ, 2000. – 352 с.
4. Программируемые контроллеры [электронный ресурс] // Stela: [сайт]. [1992-2015]. URL:http://www.ste.ru/siemens/pdf/rus/S7-1200_2014.pdf
5. Федоров Ю.Н. Справочник инженера по АСУТП. Проектирование и разработка. / Ю.Н. Федоров – М.: Инфра-Инженерия, 2008. – 928с.

УДК 004.896

**СОЗДАНИЕ ПРОГРАММЫ ДЛЯ ФРЕЗЕРНОГО РОБОТА
С ЧИСЛЕННЫМ ПРОГРАММНЫМ УПРАВЛЕНИЕМ
И РЕАЛИЗАЦИЯ ЕГО ФУНКЦИОНИРОВАНИЯ**

**Тарасов Н.Ю., Удовика Ю.Д., Бутенко Д.А., Колесников В.Ю.,
Лёвкин А.В. Научный руководитель ст. пр. Елисеев В.И.,
Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
sonic1837@rambler.ru**

Аннотация

Тарасов Н.Ю., Удовика Ю.Д., Бутенко Д.А., Лёвкин А.В., Колесников В.Ю., Елисеев В.И. Создание программы для робота и реализация его функционирования. Выполнен анализ форматов 3D моделей для робота. Определён формат для считывания координат точек 3D модели. Определён способ считывания координат точек 3D модели. Проведён анализ и определён способ минимизации количества точек. Определено, в каких программных средствах запрограммирован робот, а также рассмотрены все его функции.

***Ключевые слова:** 3D модель, координаты точек, программирование, формат.*

Постановка проблемы. Для реализации обработки материала фрезерным роботом с численным программным управлением необходимо иметь рабочую программу и загруженные в неё координаты точек. В связи с этим существует ряд проблем, которые требуют решения. Прежде всего, необходимо провести анализ форматов 3D моделей и сделать выбор формата 3D модели для считывания координат точек с неё. После этого необходимо определить способ считывания координат точек 3D модели. Далее необходимо минимизировать количество точек под фрезерного робота, потому что возможности робота ограничены конкретным их количеством. В завершении необходимо определиться со средой программирования робота и запрограммировать его действия.

Цель статьи – провести анализ существующих проблем, препятствующих реализации обработки материала фрезерным роботом с числовым программным управлением, и определить способы их решения.

Выбор формата 3D модели. Для дальнейшего считывания координат точек 3D модели требуется определиться с форматом 3D модели. Исследование показало, что всего существует очень большое количество форматов для 3D моделей. Самыми распространёнными являются модели формата DWG, DXF, DNG, STL, OBJ, IGS и MA. В ходе исследования оказалось, что для считывания координат точек подойдёт 3D модель формата

STL, потому что был найден способ с помощью программы Matlab и специальным расширением для неё считать координаты точки 3D модели. Если же необходимо считать координаты с 3D модели, например, DWG формата, то необходимо конвертировать 3D модель DWG формата в формат STL, с помощью специальной программы-конвертера. Например, существует программа 3D object converter, которая работает с большинством существующих форматов.

Считывание координат точек 3D модели. Считывание координат точек 3D модели необходимо для дальнейшего использования в программе, где координаты будет задавать движение моторам робота. Как показала практика, графические редакторы позволяют смотреть и редактировать 3D модель, но не позволяют вывести координаты каждой точки в какой-либо список. Как возможное решение был рассмотрен Matlab. Matlab имеет возможность из массива координат точек показывать 3D или 2D объект. Сами координаты хранятся в определённом списке, который потребуется в дальнейшей работе. Программе Matlab необходимо прочитать 3D модель STL формата и вывести массив координат в свой список, благодаря установленному расширению и командам, которые позволяет считывать все координаты точек 3D модели STL формата.

Минимизация количества точек. Как показывает практика точек в 3D модели много и обычно насчитывает более 100 тысяч. Данный робот предусматривает точность 2,5 - 10 тысяч точек, поэтому необходимо минимизировать количество точек. Так как точек свыше 100 тысяч, то необходимо автоматизировать процесс минимизации точек, создав программу, которая будет осуществлять этот процесс. Программирование будет осуществляться на языке C++, т.к. данный язык программирования имеет все необходимые возможности для осуществления данной задачи. Сам процесс минимизации будет осуществлять пару задач. Первой задачей является то, что необходимо убрать все отрицательные числа с координат, а именно найти минимальный элемент во всех трёх осях и в каждой оси прибавить минимальное число по модулю. Эта задача необходима для следующей задачи разбиения координат по шагам. Вторая задача разбиения координат по шагам заключается в том, что необходимо найти максимальный элемент на оси, разделить его на количество шагов на каждой оси, найти высоту на положении двух осей максимально схожими по шагам из общего списка координат.

Программирование движения робота. Основным программным обеспечением для программирования робота Lego Mindstorm NXT является NXT programming. Как показало исследование, для программирования робота так же существуют программы RobotC и BrickCC. Был выбран BrickCC потому что он более удобен для программирования, пишется на языке NXC, который очень похож на C++ с возможностью осуществлять те же функции что и в NXT programming, например, задать движение мотора. Существует проблема,

которая препятствует передаче координат роботу и заключается она в том, что программы, взаимодействующие с контроллером Lego, не могут из текстового файла считать массив данных для дальнейшей работы. В связи с этим была разработана программа, которая пишет код в текстовый файл, в котором записаны все координаты в программу робота. После работы этой программы необходимо копировать код из текстового файла и вставить его в код программы робота. На вход поступают минимизированные координаты. Фрезерный робот с численным программным управлением произведёт движение в одну сторону, и в зависимости от высоты точки в 3D модели в данной позиции будет регулировать уровень высоты фрезы. Когда закончится считывание координат по одной стороне, то двигатель возвращается на исходную позицию и другой двигатель делает шаг в сторону относительно другой оси, после чего первый двигатель продолжает движение. Фрезерный робот с численным программным управлением будет работать до тех пор, пока не будут пройдены все заданные координаты.

Выводы. Произведён анализ форматов 3D моделей и определён формат для считывания координат, для дальнейшего взаимодействия с фрезерным станком с численным программным управлением. Найден способ считать координаты точек 3D модели с помощью программного обеспечения Matlab. Обоснован выбор языка программирования, с помощью которого. Определены задачи, которые выполняет программа для процесса минимизации точек. Обоснован выбор программного обеспечения. Описан способ передачи координат роботу и принцип работы фрезерного робота с численным программным управлением.

Список литературы

1. Matlab [электронный ресурс] // Википедия – свободная энциклопедия: [сайт][2017] URL: <https://ru.wikipedia.org/wiki/MATLAB>
2. Методические указания к лабораторным занятиям «Программирование LegoMindstormNXT» // Томский государственный университет систем управления и радиотехники: [Томск]. [2012].

УДК 004.65

ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОГО УЧЕТА ДВИЖЕНИЯ ДВИГАТЕЛЕЙ ПО УЧАСТКАМ ЦЕХА РЕМОНТА МАШИН

А.В. Горчаков, А.И. Ольшевский

Донецкий национальный технический университет, г. Донецк

кафедра искусственного интеллекта и системного анализа

a_olshevskiy@mail.ru

Аннотация

Горчаков А.В., Ольшевский А.И. Проектирование автоматизированного учета движения двигателей по участкам цеха ремонта машин. В работе представлены концептуальная, логическая и физическая модели проектирования баз данных для разработки пакета программ автоматизированного учета движения двигателей по участкам цеха ремонта машин.

Ключевые слова: структура предприятия, концептуальное проектирование, двигатель, ремонт, модели баз данных, программа.

Общая постановка проблемы. Компьютеры проникли во все сферы деятельности человека. Благодаря разнообразию программного и аппаратного обеспечения сегодня возможно использование всех потенциальных возможностей компьютерных технологий. Это позволяет хранить огромное количество информации, занимая при этом минимальное место. Также компьютерные технологии позволяют быстро эту информацию обрабатывать и держать ее в защищенном виде.

Большие объемы информации практически невозможно проработать без специальных средств машинной обработки. В последнее время широкое распространение получили автоматизированные информационные системы: информационно-справочные, информационно-поисковые. Все они предназначены для регистрации, хранения и обработки данных с целью поиска и выдачи ответов на запросы пользователей. В большинстве случаев автоматизированные информационные системы разрабатывают как базы данных [1].

Цель нашей работы – разработать концептуальную, логическую и физическую модели проектирования баз данных для создания пакета программ автоматизированного учета движения двигателей по участкам цеха ремонта машин предприятия.

ООО «Механик» является признанным лидером в области производства, диагностики и ремонта электрического и механического оборудования среди

промышленных предприятий. Заводы компании оснащены современным металлообрабатывающим оборудованием, в частности, токарно-винторезной (в том числе крупногабаритными токарно-винторезными станками ДИП-500, ДИП-600 способные обрабатывать детали до 6-ти метров длинной и весом до 20-ти тонн), расточной, фрезерной, карусельной, шлифовальной, наплавочной группой станков, прессовым оборудованием, а также термическим производством, аттестованной испытательной базой и компьютерным оборудованием для конструкторских, технологических и управленческих работ. В состав компании ООО «Механик» входят три завода: «Энергозавод», «Энергомеханический завод», «Литейно-механический завод». Место расположения всех предприятий – г. Макеевка Донецкой области.

Словесное описание предметной области. Оператор вносит данные о двигателе и тип двигателя. Программа автоматически находит по типу двигателя все комплектующие этого двигателя (ротор, статор и т.д.) и заносит в таблицу. Оператор выбирает для каждого комплектующего его статус, то есть: годен, ремонт, брак или отсутствует. Затем из списка выбирает те работы, которые необходимо произвести. И последним этапом является составление графика выполняемых работ [2].

Цели автоматизированной информационной системы следующие:

- сокращение трудозатрат при выполнении традиционных информационных процессов и операций;
 - устранение рутинных операций;
 - ускорение процессов обработки и преобразования информации;
 - расширение возможностей осуществления статистического анализа и повышении точности учетно-отчетной информации.
- Задачи автоматизированной информационной системы таковы:
- добавление, изменение и удаление данных;
 - поиск данных по всем критериям;
 - сортировка данных;
 - архивирование и резервное копирование данных;
 - генерация отчетов с возможностью вывода на печать.

Исследования. Процесс проектирования базы данных был разбит на три основных этапа: концептуальное, логическое и физическое.

Концептуальное проектирование – сбор, анализ и редактирование требований к данным. Для этого осуществлялись следующие мероприятия: обследование предметной области, изучение ее информационной структуры; выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами; моделирование и интеграция всех представлений (рис.1).

Логическое проектирование баз данных – процесс конструирования общей информационной модели предприятия на основе отдельных моделей данных пользователей, которая является независимой от особенностей реально используемой СУБД и других физических условий.

При проектировании логической структуры реляционной базы данных определяется оптимальный состав таблиц для хранения исходной информации. Для каждой таблицы указывается ее название, перечень полей и первичный ключ. Идентифицируются связи между таблицами. В рамках логического проектирования БД могут формулироваться ограничения целостности, приниматься решения о создании индексов и т. д. В основе реляционной модели используется понятие «отношения», которое используется для представления набора экземпляров объекта (сущность) и отношений (связей) между объектами.

Отношение представляется как определенным образом организованная таблица.

Программная реализация. Физическое проектирование - определение особенностей хранения данных, методов доступа и т.д. Самым первым заданием на этапе физического проектирования баз данных является преобразование отношений, созданных на основе глобальной логической модели данных, в такую форму, которая может быть реализована в среде целевой СУБД.

При создании программ использовалась среды разработки C++ Builder [3].

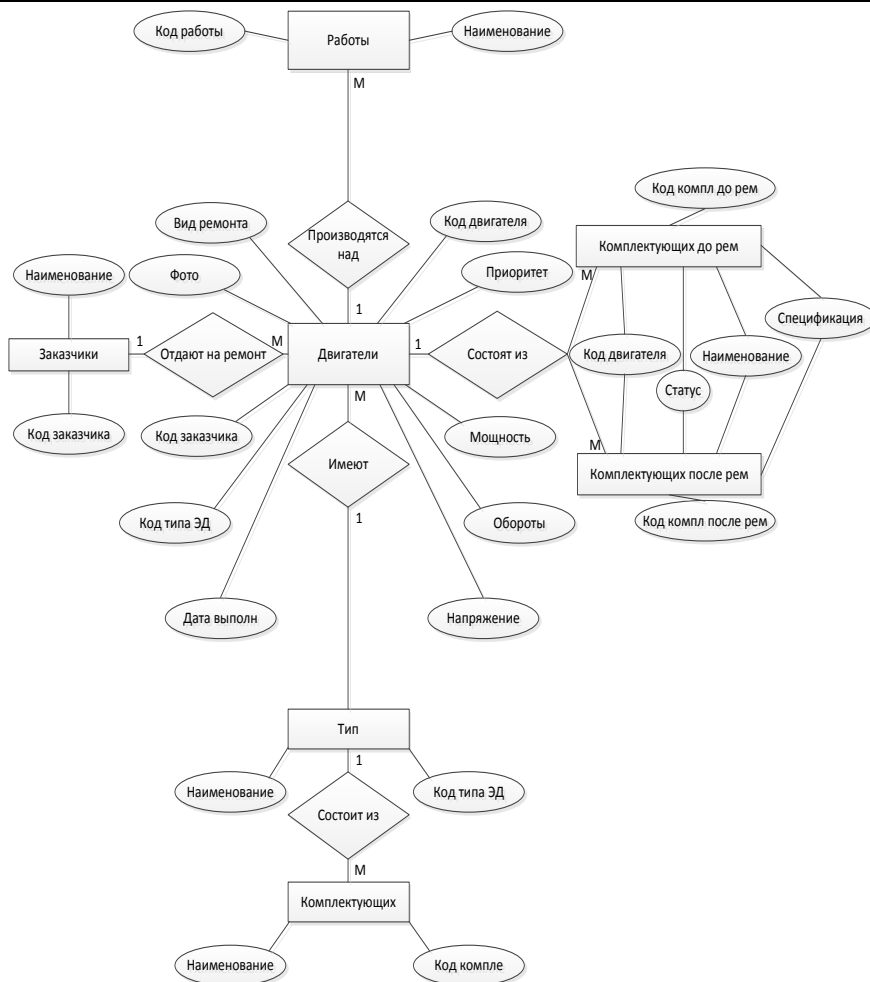


Рисунок 1 – Концептуальная модель

Физическая модель представлена на рисунке 2.

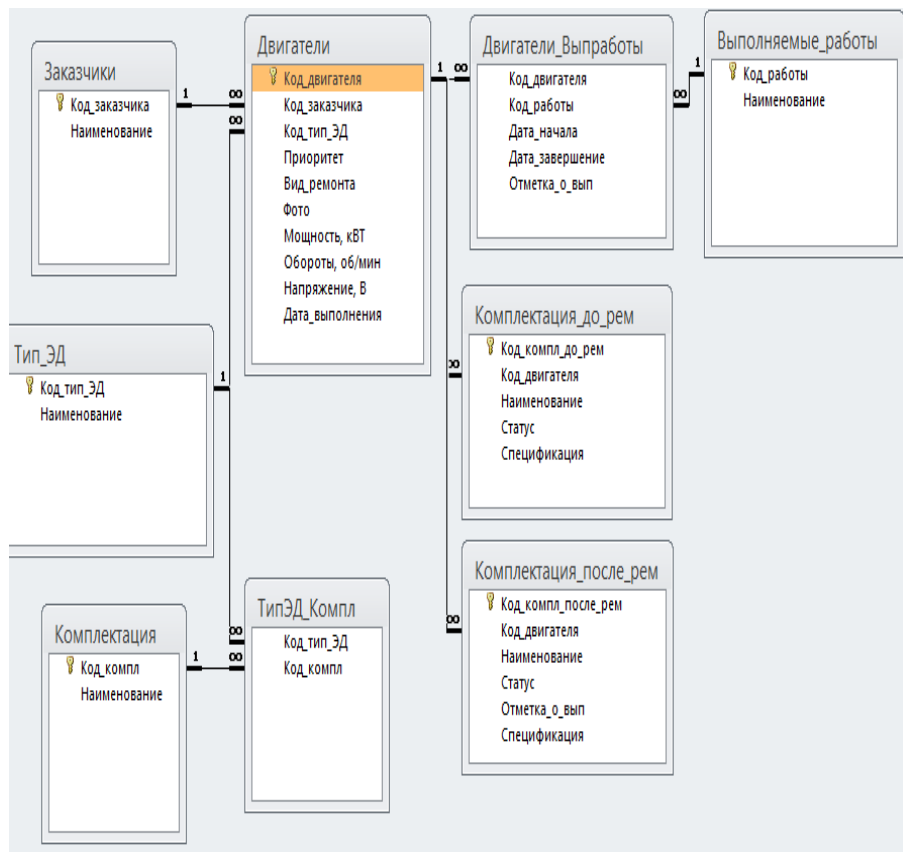


Рисунок 2 – Физическая модель

В результате была разработана программа, предназначенная для использования отделом ИТР. Основные реализованные процедуры программы: добавление двигателей, планировка выполняемых работ и при входе в программу автоматический вывод работ, которые необходимо выполнить на сегодняшний день, то есть наряды.

Выводы

В результате анализа предметной области, изучения внутренних и внешних информационных потоков и связей была спроектирована концептуальная модель данных.

На основе отдельных моделей данных пользователей разработана логическая модель. С учетом особенностей хранения данных, методов доступа

и среды разработки программно создана физическая модель и реализованы основные процедуры обработки.

Изучив досконально технологию производства, можно усовершенствовать систему на более высоком уровне. Для этого в дальнейшем необходимо добавить новые процедуры обработки базы данных:

- генерацию отчетов;
- спецификацию (то есть возможность хранения чертежей для каждого комплектующего);
- привязать комплектующие к работам и др.

Список литературы

1. Кузнецов, С. Д. Основы баз данных [Текст] / С. Д. Кузнецов. — 2-е изд. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с.
2. Бараонов В.В. Информационные технологии и управление предприятием / Бараонов В.В. – М. : Дело, 2006. – 201 с.
3. Архангельский, А. Я. Программирование в С++ Builder [Текст] / А. Я. Архангельский — Москва: Бином-Пресс, 2010 .- 1304 с.

УДК 004.91:004.738.52

ПОСТРОЕНИЕ СИСТЕМ УПРАВЛЕНИЯ ДОКУМЕНТАМИ В ЭЛЕКТРОННЫХ БИБЛИОТЕКАХ

Д.С. Мартынова, О.А. Гудаев

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа

dorothy4@yandex.ru

Аннотация

Мартынова Д.С., Гудаев О.А. Построение систем управления документами в электронных библиотеках. В работе рассматриваются новые подходы в построении электронных библиотек, выделяются проблемы, связанные с переходом документов на электронный вариант, выделяются востребованные характеристики современного проектирования систем управления документами и дается архитектура их построения, применительно к электронным библиотекам.

Ключевые слова: электронная библиотека, электронный документ, вычислительная система, клиент-серверная технология, глубинное обучение.

Постановка проблемы. С развитием вычислительной техники, в особенности глобальных вычислительных систем, перед человечеством стала задача отказа от традиционных бумажных способов хранения информации и перехода к электронным документам.

Целью любой библиотеки является формирование фонда произведений печати в соответствии с профилем комплектования, и информационное обслуживание читателей и абонентов. Круг читателей любой библиотеки обычно ограничен размерами города и близлежащих районов. Поиск любой информации осуществляется вручную, с использованием алфавитных и тематических каталогов, что может занять от пары минут до нескольких часов времени.

Создание электронных библиотек позволило устранить проблемы классических библиотек. В частности, время поиска электронного документа и время реакции на запрос стало гораздо меньше, чем при работе с бумажными документами. Увеличилась возможность доступа к редким литературным источникам, доступ к которым часто разрешен только в архивах. Повысилась надежность хранения информации, тогда как бумага подвержена многим внешним факторам разрушения: старение, нагрева или охлаждения. Кроме того, очень трудно сделать копию всего архива бумажных документов на случай непредвиденных обстоятельств. В случае с электронными носителями все происходит наоборот. Компактность, быстрота и дешевизна копирования позволяет делать и хранить столько копий информации, сколько необходимо для обеспечения надежности библиотечной системы.

Основные принципы построения системы управления документами (е-СУД). Выделяют следующие принципы построения е-СУД:

1. Масштабируемость системы – это означает, что система должна работать как с одним пользователем, так и с 10000, как с 10 документами, так и с 10 миллионами. При увеличении нагрузки на систему нужно сменить сервер, на котором работает система. Поэтому при построении решения нужно руководствоваться принципом поддержки максимально возможного количества операционных систем, если это не удастся, необходима поддержка, по крайней мере, операционных систем семейства Windows и Android. Чтобы обеспечить переносимость данных, желательна поддержка многоплатформенных серверов баз данных, таких как MySQL, MongoDB, SQLite, Oracle Database, Microsoft SQL Server, DB2 [1, 2];

2. Распределенность. Основные проблемы при работе с информацией возникают при удаленном обращении к ней. Это значит, что архитектура систем документооборота должна поддерживать взаимодействие распределенных площадок. Причем распределенные площадки могут объединяться самыми разнообразными по скорости и качеству каналами связи;

3. Открытость. Система должна аккуратно вписываться в уже существующие или новые приложения;

4. Модульность и технологичность. Не всегда необходимо внедрять весь комплекс работы с документами в организации сразу. Это может быть вызвано разными причинами: от нехватки средств, до неспособности организации резко перестроить свою работу. Поэтому система должна состоять из модулей, каждый из которых позволяет решить ту или иную задачу, причем эти модули могут быть без особого труда добавлены в работающую систему в произвольной последовательности. И что самое главное они по возможности должны быть независимы друг от друга, при сохранении глубокой интеграции между ними.

Основные задачи современных систем управления документами. К основным задачам систем управления документами относятся:

– регистрация поступающей информации (заполнение необходимых атрибутов документа);

– организация хранения документов (обеспечение хранения произвольного количества документов на разнообразных носителях);

– автоматизация операций с документами (просмотр, создание, копирование, уничтожение);

– организация индексирования и быстрого поиска документов (поддержка индексов различных типов);

– обеспечение безопасности документов (контроль доступа к информации);

– организация коллективной работы с информацией (сокращение времени ожидания запрошенных данных);

– организация распределенных хранилищ документов.

При создании систем управления документами, часто возникает вопрос о поддержке распределенного в пространстве хранилища данных. Под требованием распределённости понимается следующее:

1) доступ удаленных пользователей к хранилищу информации:

а) через локальную сеть;

б) посредством Интернет соединения с сервером;

в) через службу доставки сообщений (электронная почта).

2) взаимодействие нескольких хранилищ и одновременный доступ пользователей к информации, расположенной в разных архивах. Такого рода взаимодействие может быть построено на двух основных принципах:

а) взаимное тиражирование хранилищ – технология, которая позволяет каждому пользователю на любом рабочем месте иметь доступ к информации со всех других рабочих мест. Однако, это приемлемо только при небольших объемах информации, и практически не реализуемо для хранилищ документов серьезных размеров (порядка сотни Гигабайт и выше);

б) технология распределенного доступа – система регистрирует несколько хранилищ и как бы создает одно глобальное информационное пространство.

Модель работы системы. Пользователь делает один запрос к глобальному облачному хранилищу, и система сама делит запросы по реальным хранилищам, собирает с них ответы и выдает консолидированный результат пользователю.

В современных условиях представления информации в глобальных сетях обойтись без использования Web-технологий практически невозможно. Использование таких технологий позволяет работать с данными через обычные Web-браузеры, а они могут быть размещены на самых разнообразных клиентских платформах. Тем самым оказывается отчасти решенной проблема работы в гетерогенной сетевой среде.

Для работы в Web-среде необходимо наличие сети поддерживающей TCP/IP-протоколы. Наличие TCP/IP позволяет легко интегрировать систему с другими информационными сервисами, например, электронной почтой. При этом автоматически решается проблема масштабируемости, так как для технологий TCP/IP нет никакой разницы в том, где расположены ресурсы и интерфейс пользователя: локально, в рамках корпоративной сети, или распределены по глобальной сети. При использовании Web-технологии у е-СУД появляются серверные компоненты, отвечающие за доступ к документам через обычный Web-браузер [1, 2, 3].

Система должна обладать следующими функциями:

- формирование и сохранение информации с применением алгоритмов сжатия данных;
- организация хранилища данных на сервере;
- обеспечение доступа (просмотр, редактирование) к документам;
- добавление информации в систему, удаление из системы;
- автоматическое выделение ключевых слов из текста документа;
- быстрый поиск данных по ключевым словам;
- автоматическая классификация данных для организации каталога.

Основные задачи при проектировании е-СУД. Из рассмотрения функций, которыми должна обладать система, можно сделать вывод, что основными задачами, которые необходимо решить при проектировании е-СУД являются следующие: представление информации в системе, организация хранилища информации, организация каталога и поиск данных в соответствии с информационными потребностями пользователя.

Для хранения внутренней информации системы, такой как индексы, структура каталога с данными о вхождении в него документов и другой дополнительной информации, целесообразно использовать систему управления базами данных (СУБД). Выбор объясняется наличием такой характеристики реляционных баз данных, как небольшое время выборки конкретной записи из миллионов других, что очень важно в условиях большого количества информации в библиотеках и высоких требований к быстродействию системы. Малое время выборки из базы данных достигается путем создания индекса к таблице по одному или нескольким из ее полей. Индексы создаются посредством СУБД и обычно реализуются с применением алгоритма сбалансированного двоичного дерева.

Необходимо решение также проблемы хранения самой информации. На сегодняшний день применяется два подхода к организации хранения электронных документов. Первый состоит в том, что собственно тело документов хранится в файловой системе, второй предусматривает хранение документов в реляционной или специализированной базе данных. Второй подход хотя и обладает большей степенью защиты документов, но несет в себе ряд следующих ключевых недостатков:

- трудности с поддержкой носителей информации, отличных от жестких дисков (немного СУБД поддерживает магнитооптические и другие накопители) и практическая невозможность построения гетерогенных систем хранения;
- при работе с приложениями, в которых создаются и изменяются электронные документы, тела документов в любом случае проходят через файловую систему, а так как приложение не умеет работать напрямую с базами данных, это означает удвоение числа операций записи и считывания с

жесткого диска. При больших размерах тел документов это серьезно влияет на скорость работы.

Архитектура е-СУД. Таким образом, хранение информации необходимо осуществлять в файловой системе с новым представлением документов.

Для обеспечения эффективного поиска документов в системе управления документами необходимо наличие информационно-поисковой подсистемы [4].

Для ее реализации можно использовать методы, применяемые в классических информационно-поисковых системах. Для быстрого поиска, по ключевым словам, можно использовать методы информационно-поисковых систем словарного типа. Вся информация должна быть проиндексирована, по ключевым словам, причем выделение ключевых слов должно производиться автоматически. Поиск данных осуществляется по индексу.

Логически документы должны быть представлены в виде тематического каталога (распределены по рубрикам). Классификацию документов по рубрикам целесообразно производить автоматически с использованием возможностей Deep Learning искусственных нейронных сетей.

Схематично система управления документами может быть отображена в виде, представленном на рисунке 1.

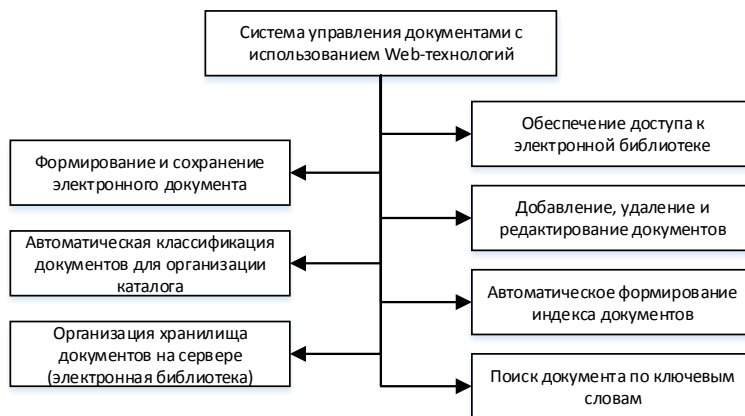


Рисунок 1 – Схема управления электронными документами

При создании электронной библиотеки следует выбрать: сервер хранения баз данных, тип хранения данных, протокол для передачи данных между сервером и клиентами, и приложения для сервера и клиента.

Для организации современных корпоративных сетей и систем с клиент-серверной архитектурой используется JSON, XML в качестве протокола передачи данных и язык Java при построении приложений для сервера и клиентов.

Приложения на Java являются эффективными и надежными для корпоративной системы.

При построении электронной библиотеки нужно использовать компонент JDBC в качестве базы данных для хранения документов, для системы запросов к базе данных, а также ответов с сервера баз данных использовать протокола SOAP в RESTful WEB, и сервер для управления базами данных – Apache 2 [5].

Структура электронной библиотеки представлена на рисунке 2.

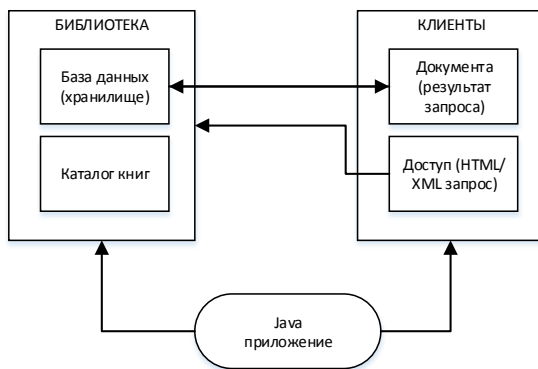


Рисунок 2 – Архитектура системы электронной библиотеки

Выводы. Рассмотрены современные подходы построения электронных библиотек. Построена архитектура работы системы и выделены востребованные функциональные характеристики системы при проектировании e-СУД. Так же была разработана структура электронной библиотеки с новыми возможностями для дальнейшего воплощения.

Список литературы

1. Петриченко Г.С. Методика выбора автоматизированных библиотечно-информационных систем для электронной библиотеки / Г.С. Петриченко, Н.К. Григорян, А.Г. Гейн. – Естественные и технические науки. – 2010. –№ 4 (49). – С. 281-287.
2. Ахметов Д.Ю. Сервис-ориентированная информационная система научного журнала «Электронные библиотеки» / Д.Ю. Ахметов, А.М. Елизаров, Е.К. Липачёв. – Электронные библиотеки. – 2016. – Т. 19. № 1. – С. 2-39.
3. Кормен Т. Алгоритмы, построение и анализ / Т. Кормен, Р. Ривест – М.: МНЦМО, 2002. – 389с.
4. Гринберг И. Разработка новых технологий информационного поиска / И. Гринберг, Л. Гербер – М.: МНЦМО, 2004. – 187с.
5. Бишоп Дж. Эффективная работа: Java 2 / Бишоп Дж. – СПб.: Питер, 2002. – 457с.

УДК 004.89

РАЗРАБОТКА СИСТЕМЫ КОНТРОЛЯ И МОНИТОРИНГА СОСТОЯНИЯ ПРОМЫШЛЕННОЙ БЕЗОПАСНОСТИ НА УГЛЕДОБЫВАЮЩИХ ПРЕДПРИЯТИЯХ

Г.А. Резникова, И.А. Тарасова

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
alinakohtan@inbox.ru

Аннотация

Резникова Г.А., Тарасова И.А. Разработка системы контроля и мониторинга состояния промышленной безопасности на угледобывающих предприятиях. Выполнен анализ угледобывающих предприятий, на основании которого разработана система контроля и мониторинга состояния промышленной безопасности на шахтах государственного предприятия «Донецкая угольная энергетическая компания».

Ключевые слова: угледобывающие предприятия, система контроля и мониторинга, промышленная безопасность, травматизм, база данных.

Постановка задачи. Уголь – самый распространенный в мире энергетический ресурс. Тема угольной промышленности в данный момент остается актуальной. Во-первых, потому, что в топливном балансе теплоэлектростанций уголь занимает более 90 процентов. Во-вторых, металлургическое производство невозможно без коксующегося угля. Это говорит об огромной значимости отрасли для экономики нашего региона.

Шахты отличаются сложными горно-геологическими условиями, которые продолжают ухудшаться. 90 процентов ныне действующих угледобывающих предприятий работают без реконструкции и существенной модернизации свыше 30 лет. Поэтому охрана труда и промышленная безопасность являются приоритетными вопросами в работе на угледобывающих предприятиях.

Цель статьи – снижение производственного травматизма и обеспечение безопасности производственных процессов, оборудования, зданий и сооружений, за счет разработки системы контроля и мониторинга состояния промышленной безопасности на угледобывающих предприятиях, позволяющей минимизировать время на принятие управленческих решений.

Анализ угледобывающих предприятий. Одной из важнейших проблем на предприятии является обеспечение безопасности и безвредности

производства, профилактика травматизма и заболеваний, связанных с неблагоприятными условиями труда.

Для решения указанной проблемы необходимо соблюдение соответствующих правовых и технических норм в области безопасности и гигиены труда, правильная организация трудового процесса, постоянный контроль и отчетность перед вышестоящими административными и хозяйственными органами.

Анализ основан на работе наиболее крупного угледобывающего государственного предприятия «Донецкая угольная энергетическая компания» (далее ГП «ДУЭК»), в состав которого на правах обособленных подразделений входят шесть шахт.

Структура подразделений ГП «ДУЭК» представлена на рисунке 1.

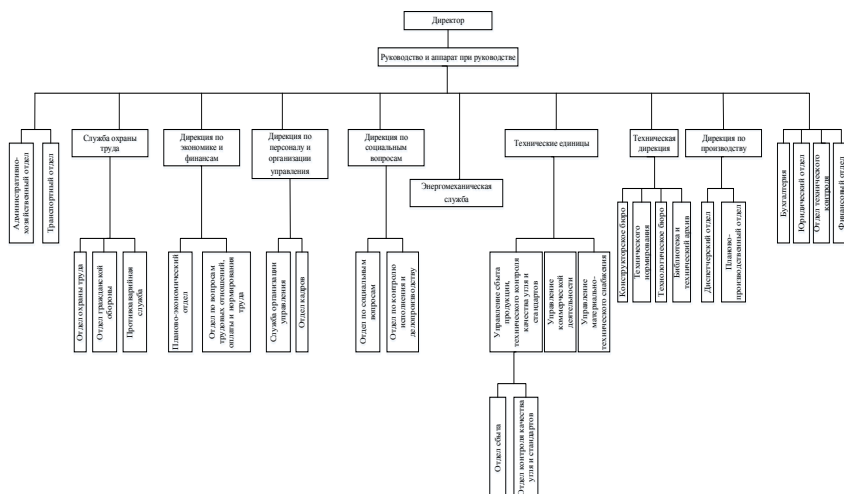


Рисунок 1 – Структура подразделений ГП «ДУЭК»

Директоры шахт для контроля за состоянием и соблюдением требований нормативных и других документов по охране труда создают службу охраны труда, целью которой является обеспечение высокопроизводительного трудового процесса, предотвращение аварийности, травматизма и профзаболеваемости при выполнении производственных заданий на предприятии, в его структурных подразделениях и на рабочих местах [1].

В свою очередь отдел охраны труда, гражданской обороны и чрезвычайных ситуаций ГП «ДУЭК» является аппаратом управления, главная функция которого обеспечение и контроль на шахтах и других обособленных

подразделениях выполнения законодательных актов и нормативных документов по охране труда.

Уровень производственного травматизма, степень профессиональных заболеваний, количество несчастных случаев и аварий можно снизить за счет своевременного принятия мер. Для того, чтобы выработать решение, отделу необходимо обработать и систематизировать большое количество информации, поэтому для автоматизации процесса обработки данных и минимизации времени на принятие решения необходимо разработать систему контроля и мониторинга состояния охраны труда на обособленных подразделениях предприятия.

Разработка системы контроля и мониторинга состояния промышленной безопасности. Внешней средой для системы являются лица ответственные за сбор, хранение и обработку поступающей информации, а также руководители, контролирующие работу, а именно:

- генеральный директор отдела охраны труда;
- специалисты отдела охраны труда;
- директор шахты;
- главный инженер;
- заместитель директора по производству;
- заместитель директора по охране труда;
- заведующий горными работами;
- заместитель главного инженера;
- главный механик (энергетик);
- главный маркшейдер (геолог);
- участковый маркшейдер (геолог);
- начальник смены;
- специалист службы охраны труда;
- горный диспетчер.

Входными данными системы являются:

- x_1 – общешахтная книга нарядов (распоряжений);
- x_2 – журнал горного диспетчера;
- x_3 – наряд-путевка горного мастера;
- x_4 – книга нарядов участка;
- x_5 – книга нарядов участка ВТБ;
- x_6 – наряд-путевка горного мастера участка ВТБ;
- x_7 – оперативный журнал горного диспетчера;
- x_8 – книга нарядов службы прогноза ГДЯ;
- x_9 – наряд-путевка горного мастера службы прогноза ГДЯ;
- x_{10} – книга нарядов участка ПРТБ;
- x_{11} – наряд-путевка горного мастера участка ПРТБ;

Выходными данными системы являются:

- y_1 – предписания;
 y_2 – графики проверок состояния охраны труда на участках;
 y_3 – план работы ПДК по охране труда;
 y_4 – график обучения горнорабочих и специалистов безопасным методам работы;
 y_5 – план оздоровительных мероприятий;
 y_6 – отчет экспресс-анализа травматизма, профзаболеваний и аварийности;
 y_7 – комплексная программа (статистика исправленных нарушений, количество проведенных мероприятий по данным направлениям).

Модель системы типа «черный ящик» представлена на рисунке 2.

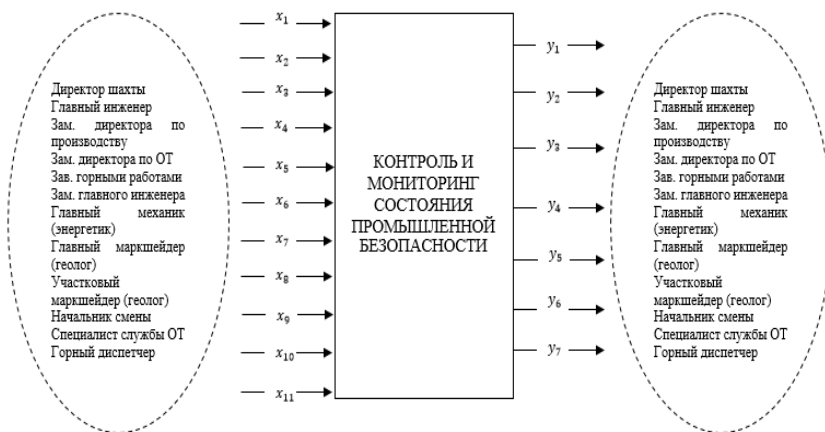


Рисунок 2 – Модель «черный ящик» системы «Контроль и мониторинг состояния промышленной безопасности»

Оперативный учет и экспресс-анализ травматизма на шахте осуществляется ежемесячно, ежеквартально составляется краткий отчет (динамика) по установленной Министерством угля и энергетики форме, который передается в службу охраны труда ГП «ДУЭК» для установления причин и тенденций травматизма, разработки отраслевых мероприятий для его предотвращения.

Наиболее полный и постоянный мониторинг опасных и вредных производственных воздействий, а также вызываемых ими случаев травматизма и заболеваемости позволит разработать комплекс защитных мероприятий, обеспечивающих безопасность и безвредность условий трудовой

деятельности. При этом система постоянно пополняется данными из предписаний, которые предоставляются руководителями и специалистами шахты для дальнейшей обработки и выработки управленческого решения.

Каждый год разрабатывается и утверждается график проведения целевых проверок отделом охраны труда, гражданской обороны и чрезвычайных ситуаций ГП «ДУЭК» на всех шахтах входящих в состав компании. На основании статистики выявленных нарушений, составляется новый график проверок, а именно очередность и название, вид проверки может меняться. Замечания, занесенные в предписания, являются основой для базы данных.

Диаграмма потоков данных системы контроля и мониторинга состояния промышленной безопасности на угледобывающих предприятиях представлена на рисунке 3.

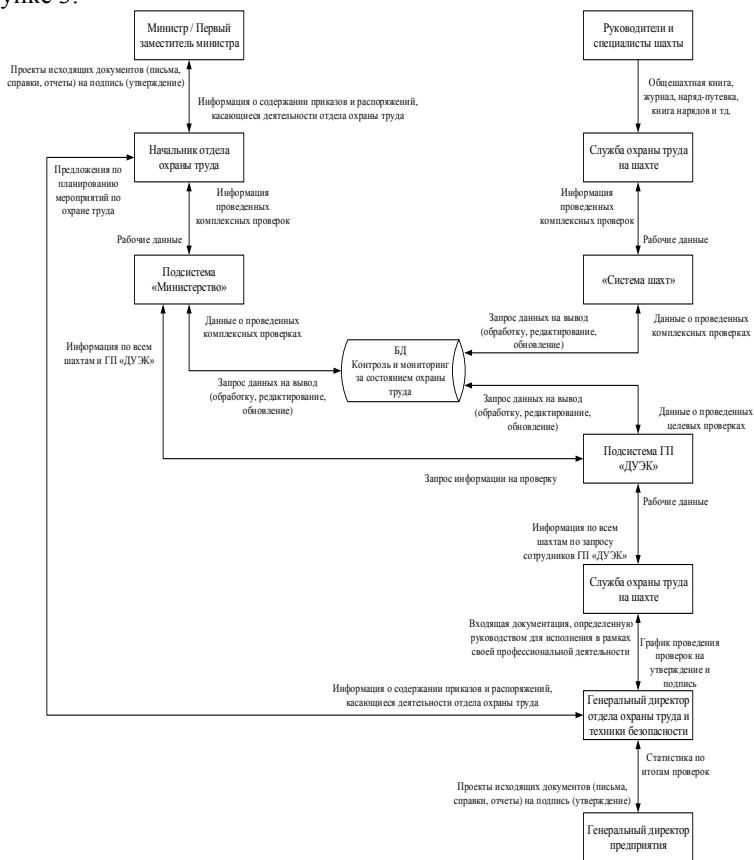


Рисунок 3 – Диаграмма потоков данных

Приведенная на рисунке 3 диаграмма отображает всех участников (пользователей) системы, а также данные, которыми обмениваются все подсистемы.

База данных имеет три уровня доступа.

Первый – это сами шахты, которые вносят данные о проверках, проведенных руководящими лицами и специалистами шахты.

Второй – это ГП «ДУЭК», который имеет доступ к информации по всем шахтам. Статистическая информация предоставляется генеральным директором отдела охраны труда генеральному директору компании.

И, третий – это Министерство угля и энергетики, которое взаимодействует с подсистемой компании для доступа к информации для дальнейших селекторных совещаний и принятия управленческих решений на основе обработанной информации.

Выводы. Анализ угледобывающих предприятий показал, что обеспечение безопасности и безвредности производства, профилактика травматизма и заболеваний, связанных с неблагоприятными условиями труда одна из ведущих проблем на предприятиях такого рода.

Разработанная система контроля и мониторинга состояния промышленной безопасности позволяет минимизировать время на принятие управленческих решений, что приводит к снижению уровня производственного травматизма.

Список литературы

1. Ефремова, О.С. Охрана труда в организации в схемах и таблицах / О.С. Ефремова. – М.: Альфа-Пресс, 2012. – 108 с.
2. Ефремова, О.С. Охрана труда от А до Я: Практическое пособие / О.С. Ефремова. – М.: Альфа-Пресс, 2013. – 672 с.
3. Беляков, Г.И. Безопасность жизнедеятельности. Охрана труда: Учебник для бакалавров / Г.И. Беляков. – М.: Юрайт, 2013. – 572 с.
4. Тимченко, Т.Н. Системный анализ в управлении: Учебное пособие / Т.Н. Тимченко. – М.: ИД РИОР, 2013. – 161 с.
5. Дейт К.Дж. Введение в системы баз данных / К.Дж Дейт, Ю.Г. Гордиенко, А.В. Слепцов; [пер. с англ.] – М.: Вильямс, 2001. – 1072 с.

УДК 004.896

СОЗДАНИЕ ПРОГРАММНОГО РОБОТА «ХУДОЖНИК» С ИСПОЛЬЗОВАНИЕМ LEGO MINDSTORMS NXT

Удовика Ю.Д., Бутенко Д.А., Тарасов Н.Ю., Колесников В.Ю.,
Лёвкин А.В. Научный руководитель ст. пр. Елисеев В.И.
Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
youlia.u@mail.ru

Аннотация

Удовика Ю.Д., Бутенко Д.А., Тарасов Н.Ю., Колесников В.Ю., Лёвкин А.В., Елисеев В.И. Создание программного робота «Художника» с использованием Lego Mindstorms. В результате проделанной работы был спроектирован, собран и запрограммирован робот-художник, работающий с использованием таких программ как: МАТЛАБ, Visual Studio 2015 и NXT Program 2.0.

Ключевые слова: робот, программное обеспечение, программа.

Постановка проблемы. Требуется разработать роботизированную руку, способную рисовать заданное ей графическое изображение при помощи набора Lego Mindstorms NXT. Для реализации этого следует выполнить следующие пункты:

- спроектировать виртуальную модель робота
- выбрать программы для дальнейшей работы
- решить основную задачу движения робота

Цель статьи – привести подробное описание проделанной работы, ознакомить пользователя с проблемами программирования и сборки робота «Художника».

Проектирование виртуальной модели робота. Для проектирования виртуальной 3D модели была использована программа LEGO Digital Designer. В программе предусмотрено создание моделей версии Mindstorms NXT. На рисунке 1 представлена полученная модель робота – художника.

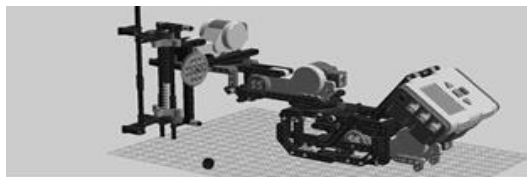
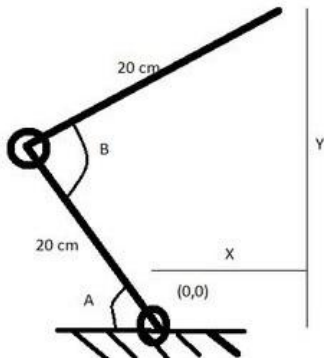


Рисунок 1 – 3D модель робота – художника

Выбор программ для дальнейшей работы. Под дальнейшей работой подразумевается создание программного обеспечения робота. Основной программной средой был выбран продукт MathWorks – МАТЛАБ R2009b, но для создания данного ПО подойдут и более ранние и более поздние версии данного продукта. Для проектирования роботизированных рук была создана программа SCARA, являющаяся производной составляющей Simulink МАТЛАБ, позволяющая спроектировать не только самого робота, но и траекторию движения его конкретных деталей, что позволяет воспроизвести движение шариковой ручки по бумаге и запрограммировать его. Для работы используются *.m файлы с указанной траекторией, математически рассчитанной в МАТЛАБ. Далее был использован уже готовый продукт, в результате работы которого были получены файлы формата srr, написанные на языке программирования С. Данный файл требуется скомпилировать в Visual Studio 2015. К сожалению, для данных программ не предусмотрена возможность экспорта или импорта ПО в Lego Mindstorms NXT или конвертации исходных файлов в формат, понимающийся роботом, поэтому полученный файл после компиляции требуется передать через подключенный процесс, выбрав NXT Program 2.0. Среда NXT Program 2.0. является основным официальным продуктом для работы с конструктором Lego, он предусматривает возможность создания, изменения и передачи готового файла контроллеру. Все программы находятся в открытом доступе и наиболее часто используются для создания подобного ПО.

Основная задача движения робота. Основной задачей является расчет углов между заданной координатой точек и частями роботизированной руки. Это позволит перемещать ручку, закрепленную в «кисти» робота, на требуемое расстояние. После расчета маршрута, программа SCARA передаст полученную матрицу значений конкретному двигателю, который в свою очередь повернет руку на нужный угол. Начальное положение «кисти» робота определяется датчиком движения, программа может прекратить свое действие при активации датчика касания. Наглядное представление математических расчетов показано на рисунке 2.



$$X = \cos(B-A) \cdot 20 - \cos(A) \cdot 20$$

$$Y = \sin(A) \cdot 20 + \sin(B-A) \cdot 20$$

From the above equations we get
 $\cos(B) = (800 - X^2 - Y^2) / 800$
 $A = 20 \cdot \tan(x/y) + (B/2)$

Рисунок 2 – Расчет углов поворота роботизированной руки

Внешний вид программы SCARA для данного робота представлен на рисунке 3.

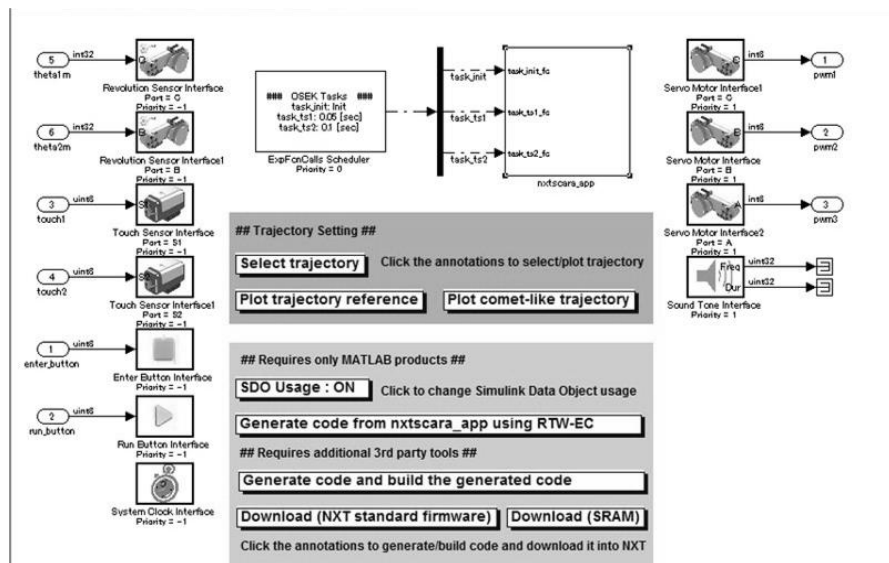


Рисунок 3 – Модель NXT контроллера

Выводы. Произведено подробное описание разработки ПО робота-художника, представлено описание использованных программ. Разработан метод передачи значений координат точек и маршрута движения роботу, с использованием углов, на которые способны переместить конструкцию

двигатели. Несмотря на неудобство, связанное с использованием большого количества программ и получения возможных ошибок, как следствие нескольких конвертаций, получен требуемый программный продукт.

Список литературы

1. Программирование на LEGO Mindstorms [электронный ресурс] // Alexey Kletsel Design: [сайт]. [1994-2013]. URL: <https://habrahabr.ru/post/274519/>
2. Методические указания к лабораторным занятиям «Программирование Lego Mindstorm NXT» // Томский государственный университет систем управления и радиотехники: [Томск]. [2012].
3. Работа в NXT Programming [электронный ресурс] University of Zurich, Department of Informatics, Artificial Intelligence Laboratory 3442 VU Introduction to Artificial Intelligence 06/07, Rolf Pfeifer, Prof., Dr. sc. techn. URL: <https://docviewer.yandex.ua/view/>
4. SCARA [электронный ресурс] // Википедия – свободная энциклопедия: [сайт] [2017] URL: <https://en.wikipedia.org/wiki/SCARA>

УДК 004.738.52:004.822

ОНТОЛОГИЧЕСКИЕ МОДЕЛИ ОПИСАНИЯ ОРГАНИЗАЦИИ ДЕЯТЕЛЬНОСТИ В ФОРМАТЕ JSON-LD

С.Н. Цыганков, О.А. Гудаев

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
fruts1997@gmail.com

Аннотация

Цыганков С.Н., Гудаев О.А. Онтологические модели описания организации деятельности в формате JSON-LD. Выполнен анализ онтологических моделей организации деятельности и описание онтологий форматом JSON-LD. Разработана практическая модель применения формата JSON-LD для описания онтологической модели бизнес-процессов.

Ключевые слова: связанные данные, онтологическая модель, организация деятельности, узел семантической сети.

Постановка проблемы. Для обмена информацией об организации функционирования цифровых проектов используется онтологические модели бизнес-процессов. Несогласованность терминологии, из-за новизны изобретения цифровых продуктов, не позволяет распространять опыт организации деятельности на аналогичные проекты. А в общем случае передавать опыт в экспертные хранилища для автоматического извлечения знаний. Конкуренция в области стандартизации онтологических приемов работы цифровых сервисов разворачивается во многих странах.

Цель статьи – расширить применение формата JSON-LD для описания онтологии организации деятельности.

Анализ формата JSON-LD и моделей организации деятельности.

JSON-LD — это способ передачи связанных данных (Linked Data, LD) с помощью текстового формата JSON (JavaScript Object Notation). Формат JSON-LD разработал Консорциум Всемирной паутины (W3C). Использование текстового формата JSON позволяет людям легко читать и писать документы, размеченные с помощью JSON-LD [1]. Страницы с разметкой JSON-LD облегчают структурирование данных машинами и распознавание понятий, что для владельцев сайтов важно в контексте поискового продвижения [2].

На сайте schema.org находим иерархический словарь всех классов. На любом онлайн ресурсе, предназначенном для работы с файлами JSON-LD формата, укажем ссылку на вышеуказанную библиотеку.

Например, возьмем из библиотеки класс «Vehicle», который является наследником класса «Product», который, в свою очередь, является наследником основного класса «Thing». Данный класс содержит 4 наследника: «BusOrCoach», «Car», «Motorcycle», «MotorizedBicycle». Добавим еще одного с названием «My vehicle», укажем ему двух наследников: «BMW», «Audi». Проверим все это в гугл-валидаторе, вид которого показан на рисунке 1.

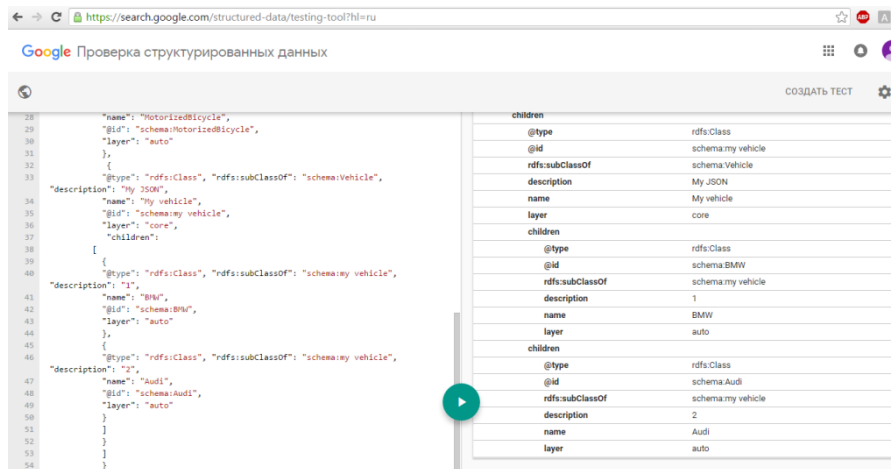


Рисунок 1 – Интерфейс гугл-валидатора

На практике использование формата JSON-LD улучшает представленность сайта в поисковой выдаче [3]. Вы получаете расширенные фрагменты исходного текста, которые привлекают внимание пользователей и повышают посещаемость ссылок. В частности, с помощью JSON-LD можно размечать данные для графа знаний, отображать в SERP поиск по сайту и делать разметку событий.

На рисунке 2 изображен пример синтаксиса формата JSON-LD.

```
{
  "@context": "http://schema.org",
  "@type": "Zoo",
  "alternateName": "Московский зоопарк",
  "address": "123242 Москва, Б. Грузинская, 1",
  "openingHours": "9.00-17.00",
  "telephone": "+74992522951",
  "url": "http://moscowzoo.ru"
}
```

Рисунок 2 – Пример синтаксиса формата JSON-LD

Модель организации деятельности – это система, исследование которой служит средством для получения информации об организации деятельности, представления устройства или концепции определенного плана работы [4].

В теории организации выделяют 4 типа моделей организации отражающих эволюцию теоретических концепций, определяющих сущность самих организаций, роль доминирующих функций менеджмента, а также критерии, применяемые для оценки эффективности их деятельности: механическая модель организации, органическая модель организации, патерналистская модель и конвенциональная модель. В каждой из этих моделей представлен различный тип управления различными процессами, основное отличие в данных подходах, это своеобразное видение представления организации деятельности.

Практическая реализация нотации бизнес-модели в формате JSON-LD. В данный момент времени формат JSON-LD существует лишь для облегчения жизни поисковых систем, но спектр возможностей данного формата намного шире, чем нам кажется. С помощью формата JSON-LD можно описать все что угодно, и увековечить в понятном как для компьютера, так и для человека формате данных. Что бы продемонстрировать данную гипотезу была реализована программа, которая позволяет создавать файл формата JSON, и помещать данные в него, которые описывают различные организации или процессы, в данном приложении можно описать любой зоопарк.

На рисунке 3 представлено проверка онтологической модели на соответствие формату связанных данных JSON-LD.

На рисунке 4 изображен интерфейс данного приложения. В будущем этот формат станет намного полезнее, чем сейчас и поможет человечеству ускорять исполнение одних и тех же процессов, путем стандартизации этих самых процессов.

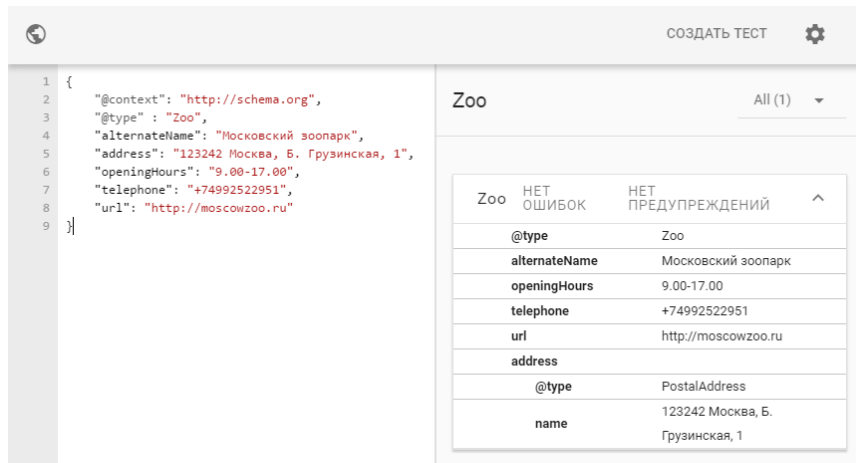


Рисунок 3 – Валидация онтологической модели JSON-LD

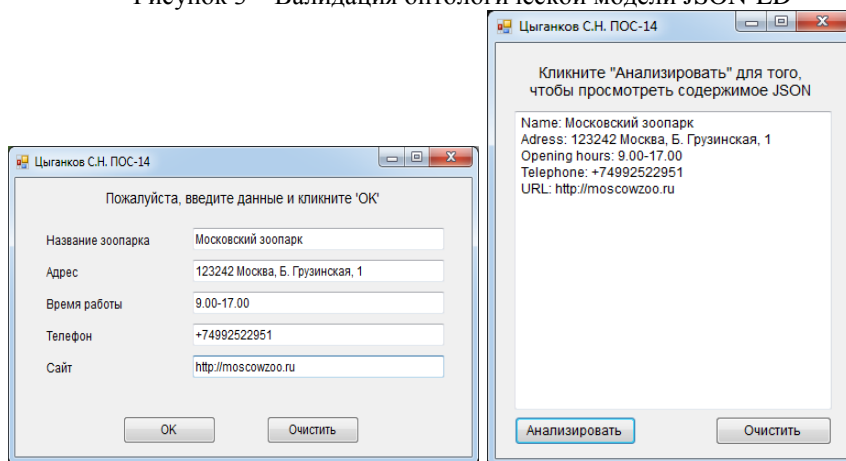


Рисунок 4 – Интерфейс ввода и анализа JSON-LD знаний

Выводы. Описан формат связанных данных JSON-LD, набирающий популярность в описании семантических знаний прикладных предметных областей. Предложена возможность применения формата в описании моделей организации деятельности. Разработано тестовое приложение для проверки гипотезы о доступности программной реализации обмена моделями онтологий в Интернете и понятности при чтении человеком открытых моделей.

Список литературы

1. Шульга Т.Э. Разработка требований к редактору веб-онтологий / Т.Э. Шульга, Н.Э. Нурмамедов // Проблемы управления в социально-экономических и технических системах : сборник научных статей по материалам XI Международной научно-практической конференции, 09-10 апреля 2015 г., г. Саратов. – Саратов : СГТУ, 2015. – С. 3-5.
2. Авдеенко Т.В. Гибридная модель представления знаний для реализации вывода во фреймовой онтологии / Т.В. Авдеенко, М.А. Бакаев. – Научный вестник Новосибирского государственного технического университета. – 2013. – № 3 (52). – С. 84-90.
3. Нестеренко С.А. Разработка модели онтологии диагностики сервис-ориентированных сетевых структур на основе многосортного языка прикладной логики / С.А. Нестеренко, П.М. Тишин, А.С. Маковецкий. – Электротехнические и компьютерные системы. – 2012. – № 7. – С. 102-108.
4. Боуш Г.Д. Новый взгляд на онтологию кластеров предприятий / Г.Д. Боуш // Издательство: Региональная общественная организация содействия развитию институтов Отделения экономики РАН (Москва). – Экономическая наука современной России – 2011. – № 1. С. 49-59.

УДК 005+65.011.56

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ФОРМИРОВАНИЯ АЛЬТЕРНАТИВНЫХ РЕШЕНИЙ В ЗАДАЧАХ ПРИНЯТИЯ РЕШЕНИЯ

А.С. Милая, Е.О. Савкова

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
milaya.don@gmail.com

Аннотация

Милая А.С., Савкова Е.О. Применение генетического алгоритма для формирования альтернативных решений в задачах принятия решения. Исследована возможность применения генетического алгоритма для формирования альтернативных решений в задачах принятия решений в условиях неопределённости. Алгоритм применён к практической задаче, определены его преимущества и удобство использования.

Ключевые слова: генетический алгоритм, альтернативные решения, задачи принятия решений, условия неопределённости.

Постановка проблемы. Для принятия решения руководителю необходимо иметь список альтернативных решений, на формирование и продумывание которых он часто не может выделить время. Для решения этой проблемы можно автоматизировать этот процесс, используя генетический алгоритм. В этом случае руководителю достаточно будет задать количество альтернатив, которые он хочет получить, количество итераций алгоритма и размер исходной популяции.

Цель, задачи и актуальность работы. Целью данной статьи является применение генетического алгоритма для формирования альтернативных решений, и таким образом упрощение принятия решения руководителем.

Основными задачами работы можно выделить применение генетического алгоритма к практической задаче, а также обоснование рациональности и оптимальности его использования.

Актуальность статьи обусловлена необходимостью принятия решения руководителем в короткие сроки и необходимостью использования для этого наименьшего количества ресурсов.

Краткое описание генетического алгоритма.

Генетический алгоритм (ГА) представляет собой алгоритм поиска, построенный на принципах сходных естественному отбору. Он работает с популяцией особей, в хромосомах каждой из которых закодировано возможное решение задачи. В начале работы алгоритма популяция формируется случайным образом. Для оценки качества закодированных решений используют функцию приспособленности. По результатам

оценивания особей наиболее приспособленные из них выбираются для скрещивания. В результате скрещивания выбранных особей посредством применения генетического оператора кроссовера создается потомство. Созданные потомки формируют новую популяцию, причем часть потомков мутирует (используется генетический оператор мутации). Этап, включающий последовательность «Оценивание популяции» – «Селекция» – «Скрещивание» – «Мутация», называется поколением. Эволюция популяции состоит из последовательности таких поколений. [3]

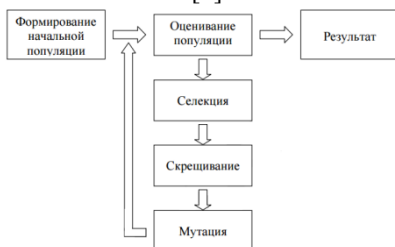


Рисунок 10 – Схема работы генетического алгоритма

Применение генетического алгоритма к практической задаче.

Применим генетический алгоритм для формирования альтернативных решений к задаче, рассмотренной в статье «Формализация задачи принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости». [2]

Применим алгоритм формирования альтернативных решений к задаче.

1. Задаём исходные данные: из 20 возможных альтернатив необходимо получить 4 самые лучшие за 3 итерации алгоритма.

2. Вынесем в таблицу данные по времени обслуживания для удобства расчётов.

3. Длина каждой хромосомы X_i равна 6, так как по условию имеем 6 организаций. Ген x_{ij} – время обслуживания одной организации. Таким образом хромосома имеет вид: $X_i = \{x_{i1}; x_{i2}; x_{i3}; x_{i4}; x_{i5}; x_{i6}\}$.

4. Посчитаем необходимый минимум для всех организаций, это будет первая хромосома: $10 + 10 + 10 + 10 + 10 + 30 = 80 \text{ часов}$

Первая хромосома имеет вид: $X_1 = \{10; 10; 10; 10; 10; 30\}$.

5. Формируем ещё 19 хромосом следующим образом.

1) Генерируется случайное число k в диапазоне от 1 до 6 (общее количество организаций). К этому количеству организаций добавим оставшееся время обслуживания $t_0 = 140 - 80 = 60$ часов, выделяя из него часть для каждой из k организаций. Не обязательно все 60 часов должны быть распределены между организациями.

Таблица 1 – Время обслуживания организаций

Организация	Время обслуживания (часов)	
	минимальное	максимальное
Управление физкультуры и спорта	10	30
Министерство здравоохранения	10	30
Министерство образования	10	30
Енакиевский коксохимический завод	10	20
Ясиновский коксохимический завод	10	20
Донецкий металлургический завод	30	70

2) Далее генерируем множество $\{K_i, i = \overline{1, k}\}$ не повторяющихся случайных чисел в диапазоне от 1 до 6. Эти числа обозначают номера организаций, для которых выделяется какая-то часть из оставшегося времени обслуживания.

3) Для каждой из k организаций генерируем число t_i в диапазоне от минимального до максимального времени её обслуживания. При этом сумма всех t_i не должна превышать t_0 , то есть 60 часов: $\sum_{i=1}^k t_i \leq 60$

Если сумма превышает 60 часов, все t_i для текущей хромосомы генерируются заново.

4) Прибавляем каждое t_i ко времени обслуживания соответствующей организации K_i .

Таким образом получаем следующее множество хромосом:

$$\begin{aligned}
 X_1 &= \{10; 10; 10; 10; 30\} & X_{11} &= \{10; 10; 10; 13; 10; 30\} \\
 X_2 &= \{10; 10; 10; 10; 19; 49\} & X_{12} &= \{23; 20; 28; 18; 11; 44\} \\
 X_3 &= \{10; 10; 10; 10; 10; 61\} & X_{13} &= \{10; 26; 21; 10; 10; 30\} \\
 X_4 &= \{22; 13; 10; 16; 11; 30\} & X_{14} &= \{19; 14; 28; 10; 10; 30\} \\
 X_5 &= \{18; 15; 26; 19; 16; 34\} & X_{15} &= \{21; 15; 19; 19; 14; 30\} \\
 X_6 &= \{10; 10; 22; 20; 20; 30\} & X_{16} &= \{21; 29; 10; 10; 10; 30\} \\
 X_7 &= \{10; 29; 16; 18; 10; 38\} & X_{17} &= \{10; 10; 10; 10; 15; 61\} \\
 X_8 &= \{12; 10; 10; 16; 15; 30\} & X_{18} &= \{15; 10; 10; 20; 14; 63\} \\
 X_9 &= \{14; 29; 21; 10; 20; 41\} & X_{19} &= \{10; 10; 10; 20; 10; 30\} \\
 X_{10} &= \{27; 10; 10; 10; 19; 66\} & X_{20} &= \{14; 20; 15; 11; 19; 59\}
 \end{aligned}$$

Затем выполняем отбор лучших десяти хромосом методом сигма отсечения, масштабирующего значение целевой функции. Вероятность выбора особи тем больше, чем оптимальнее значение масштабируемой целевой функции:

$$p_i = \frac{F_i}{\sum_{i=1}^N F_i}, \quad \text{где } F_i = 1 + \frac{f_i + f_{avg}}{2\sigma},$$

f_i – значение целевой функции для i -той хромосомы,

f_{avg} – среднее значение целевой функции для всей популяции,

σ – среднеквадратичное отклонение значения целевой функции.

В качестве целевой функции возьмём 25% от суммы стоимостей обслуживания (c_{ij}) каждой организации в хромосоме, так как именно эта сумма будет получена вне зависимости от произошедшего неопределённого события.

$$f_i = 0.25 * \sum_{j=1}^6 (x_{ij} * c_{ij})$$

Получим следующие 10 хромосом:

$$X_5 = \{18; 15; 26; 19; 16; 34\}$$

$$X_{15} = \{21; 15; 19; 19; 14; 30\}$$

$$X_6 = \{10; 10; 22; 20; 20; 30\}$$

$$X_{17} = \{10; 10; 10; 10; 15; 61\}$$

$$X_7 = \{10; 29; 16; 18; 10; 38\}$$

$$X_{18} = \{15; 10; 10; 20; 14; 63\}$$

$$X_9 = \{14; 29; 21; 10; 20; 41\}$$

$$X_{20} = \{14; 20; 15; 11; 19; 59\}$$

$$X_{10} = \{27; 10; 10; 10; 19; 66\}$$

$$X_{12} = \{23; 20; 28; 18; 11; 44\}$$

6. Из выбранных десяти хромосом случайным образом выбираем 5 пар хромосом-родителей для оператора скрещивания, каждая из которых даст двух потомков: $X_{18} - X_9$; $X_6 - X_7$; $X_9 - X_6$; $X_5 - X_{12}$; $X_{10} - X_6$.

7. Для выбранных пар выполняем одноточечное скрещивание следующим образом.

1) Генерируем случайное число $s = \overline{2, 6}$.

2) Хромосомы-родители обмениваются генами с s -того по 6-й.

3) Получаем две новые хромосомы-потомка, которые добавляются к старой популяции. Если суммарное время обслуживания какой-то из хромосом больше возможного (140 часов), эта хромосома не добавляется к популяции.

В результате получаем 10 хромосом потомков и добавляем их к популяции хромосом-родителей.

8. Получив новую популяцию из хромосом-родителей и хромосом-потомков, выполним оператор мутации, которая может произойти с каждой хромосомой с вероятностью 0.001%. При мутации выполняются следующие действия:

1) Генерируется случайное число $s = \overline{1, 6}$.

2) Генерируется число t в диапазоне $[-5; 5]$.

3) Число t добавляется к s -тому гену.

4) Если время обслуживания s -того гена-организации превысило максимальное или оказалось меньше минимального, пункты 1-3 выполняются заново для исходной хромосомы.

После применения оператора мутации получим новую исходную популяцию для следующей итерации алгоритма:

$$X_1 = \{18; 15; 26; 19; 16; 34\}$$

$$X_5 = \{27; 10; 10; 10; 19; 66\}$$

$$X_2 = \{10; 10; 22; 20; 20; 30\}$$

$$X_6 = \{23; 20; 28; 18; 11; 44\}$$

$$X_3 = \{10; 29; 16; 18; 10; 38\}$$

$$X_7 = \{21; 15; 19; 19; 14; 30\}$$

$$X_4 = \{14; 29; 21; 10; 20; 41\}$$

$$X_8 = \{10; 10; 10; 10; 15; 61\}$$

$$X_9 = \{15; 10; 10; 20; 14; 63\}$$

$$X_{10} = \{14; 20; 15; 11; 19; 59\}$$

$$X_{11} = \{15; 10; 10; 20; 14; 41\}$$

$$X_{12} = \{14; 29; 21; 10; 20; 63\}$$

$$X_{13} = \{10; 10; 22; 18; 10; 38\}$$

$$X_{14} = \{10; 29; 16; 20; 20; 30\}$$

$$X_{15} = \{14; 29; 22; 23; 20; 30\}$$

$$X_{16} = \{10; 10; 21; 10; 20; 41\}$$

$$X_{17} = \{18; 15; 26; 19; 16; 44\}$$

$$X_{18} = \{23; 20; 28; 18; 11; 34\}$$

$$X_{19} = \{27; 10; 22; 20; 20; 30\}$$

$$X_{20} = \{10; 10; 10; 10; 19; 66\}$$

Выполнив ещё две итерации генетического алгоритма, получим следующую популяцию хромосом-альтернатив и отсортируем её по убыванию их оптимальности:

$$X_1 = \{14; 29; 21; 10; 20; 63\}$$

$$X_2 = \{23; 20; 28; 18; 11; 44\}$$

$$X_3 = \{27; 10; 22; 20; 20; 41\}$$

$$X_4 = \{14; 29; 22; 23; 20; 30\}$$

$$X_5 = \{14; 29; 22; 23; 20; 30\}$$

$$X_6 = \{18; 15; 26; 19; 16; 44\}$$

$$X_7 = \{27; 10; 10; 10; 19; 66\}$$

$$X_8 = \{14; 29; 22; 20; 20; 30\}$$

$$X_9 = \{23; 20; 28; 18; 11; 34\}$$

$$X_{10} = \{14; 20; 15; 11; 19; 59\}$$

$$X_{11} = \{14; 20; 28; 18; 11; 34\}$$

$$X_{12} = \{27; 10; 10; 20; 20; 30\}$$

$$X_{13} = \{14; 20; 15; 11; 11; 44\}$$

$$X_{14} = \{14; 20; 15; 11; 11; 34\}$$

Таким образом, выбрав первые 4 хромосомы, получим четыре альтернативных решения, которые являются самыми оптимальными для заданных исходных параметров.

$$X_1 = \{14; 29; 21; 10; 20; 63\}$$

$$X_2 = \{23; 20; 28; 18; 11; 44\}$$

$$X_3 = \{27; 10; 22; 20; 20; 41\}$$

$$X_4 = \{14; 29; 22; 23; 20; 30\}$$

Выводы. В ходе работы была исследована возможность применения генетического алгоритма для формирования альтернативных решений, в результате чего упрощается задача принятия решения для руководителя. В случае применения алгоритма на практике руководителю не придётся тратить время и усилия на продумывание и подбор альтернативных решений, и он сможет сосредоточиться непосредственно на выборе лучшего решения из предложенных. Преимущество алгоритма заключается в том, что он может охватить достаточно большой объём данных и учесть такие альтернативные решения, которые без использования алгоритма сформировать будет трудно.

Список литературы

1. Кочетов Ю. Курс лекций по теории принятия решений. – Режим доступа: <http://www.math.nsc.ru/LBRT/k5/>.

2. Милая, А.С. Формализация задачи принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости /А.С. Милая, Е.О. Савкова // Научный журнал «Электронные информационные системы» №1 (12) 2017. – Москва, 2017. – С. 82–93.

3. Панченко Т.В. Генетические алгоритмы: уч.-метод. пособие /под ред. Ю.Ю.Тарасевича. –Астрахань: Издат.дом «Астраханский университет», 2007. – 87 [3] с.

УДК 004.932.2

ИССЛЕДОВАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ ТЕСТА ПОРСОЛТА (ПРИНУДИТЕЛЬНОЕ ПЛАВАНИЕ) ДЛЯ УСТАНОВЛЕНИЯ УРОВНЯ ДЕПРЕССИВНОСТИ ЛАБОРАТОРНЫХ ЖИВОТНЫХ

С.Л. Кацель, Т.Н. Кравец

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа

Аннотация

Кацель С.Л., Кравец Т.Н. Исследование программной реализации теста Порсолта (или принудительное плавание) для установления уровня депрессивности лабораторных грызунов. Приведена схема конфигурации оборудования для проведения теста Порсолта и его дальнейшей компьютерной обработки. Рассмотрены некоторые способы обработки видеоряда и вычисления уровня депрессивности подопытных животных.

***Ключевые слова:** тест Порсолта, принудительное плавание, уровень депрессивности, обработка изображений, вектор значений.*

Постановка проблемы. Для оценки эффективности препаратов различного механизма действия (психостимуляторов, ноотропов, анксиолитиков, антипсихотиков и т.д.) используется ряд методик, основанных на фиксации поведенческих реакций животных. Одним из таких поведенческих тестов является тест Порсолта (Porsolt) или отчаяния [1-2], получивший широкое распространение в психонейрофармакологии.

Методика поведенческого отчаяния (беспомощности) по методу Порсолта является базисной моделью оценки антидепрессантов. Суть метода состоит в том, что животные, попадая в воду, начинают проявлять бурную двигательную активность, направленную на поиск выхода из авersiveй (неприятной, неизбежной) ситуации. Затем животные оставляют попытки выбраться из сосуда и зависают в воде в характерной позе (рисунок 1), оставаясь полностью неподвижными или совершая незначительные движения, необходимые для поддержания морды над поверхностью воды.

Показателем выраженности депрессивного состояния в условиях данного теста является длительность неподвижности животных (иммобилизации). В тесте поведенческого отчаяния после активных попыток избавления из авersiveй ситуации у животных наступает стадия иммобилизации, которая отражает депрессивное состояние. Состояние иммобилизации оценивают визуально с определением ее длительности в течение 5 или 6 (чаще) мин.

наблюдения. Вещества с антидепрессантной активностью облегчают это состояние, уменьшая длительность неподвижности животных.

Общий вид опытной установки предоставлен на рисунке 2. Она представляет собой прозрачную ёмкость с водой, в которую погружается подопытное животное. Для фиксации процесса устанавливается видеокамера, снимающая опыт и передающая информацию на компьютер для обработки. Животное погружается в сосуд, начинается видеозапись.



Рисунок 1. Позиция покоя

Обработка видеоряда и демонстрация результатов происходит одновременно с проведением теста, либо первоначально делается запись опыта, которая затем обрабатывается.

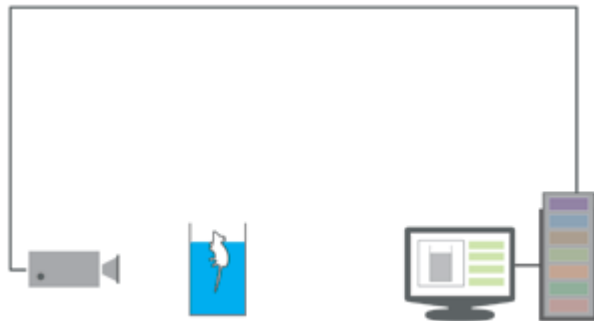


Рисунок 2. Устройство опытной установки

Цель статьи – провести анализ предметной области, разработать наиболее оптимальный способ оценки уровня депрессивности лабораторных животных в тесте Порсолта путём обработки видеозаписи опыта.

Возможные методы решения задачи. Предполагается, что работа с видео будет сведена к обработке последовательности кадров, каждый из которых будет преобразован в полутоновое («grayscale») изображение, где каждый его элемент представляет собой значение «яркости» соответствующего элемента (суммы его уровней по цветным каналам, взятым с определёнными коэффициентами).

Основной задачей в тесте Порсолта является определение количества движений лабораторного животного за квант времени, по уровню количества движения каждый квант относится к одной из трёх фаз (фаза покоя, фаза поддерживающих движений, фаза активных движений). После завершения теста определяется суммарная длительность каждой фазы, и по взаимному соотношению общей продолжительности трёх фаз и делается вывод о текущем уровне депрессивности испытуемого животного.

Таким образом, задачей программного решения становится определение количества движений за единицу времени (может иметь переменный размер). Рассмотрим некоторые возможные подходы к реализации.

Построение скелетной модели. Данный процесс строится на захвате движений подопытного животного и их дальнейшей обработке. Захват движений может осуществляться с помощью маркерной или безмаркерной технологий. При применении маркерной технологии требуется дополнительное оборудование либо манипуляции с экспериментальным животным:

1) на определённые места животного крепятся датчики, положение которых с высокой точностью отслеживается и передаётся в компьютер, где на основании анализа определяется количество движений в измеряемый момент

времени. Данный метод позволяет довольно точно определять местоположение характерных частей тела животного и строить достоверную скелетную модель, однако датчики имеют довольно заметный вес и размеры по сравнению с телом животного и могут причинять ему неудобства либо ограничивать в подвижности, что может исказить наблюдаемую картину. Кроме того, такое оборудование имеет довольно высокую стоимость.

2) Вместо радиодатчиков на поверхность тела животного наносятся маркеры из яркой краски, не совпадающей с окраской (например, пикриновой кислотой, получившей широкое применение в экспериментальной физиологии для маркирования лабораторных грызунов), либо выполненные из материала, хорошо отражающего определённый вид оптического излучения, например, инфракрасного. Тогда фиксация положения тела и его конечностей может осуществляться обычной видеокамерой, дополненной при необходимости светофильтрами, отсеивающими лишнее оптическое излучение. Данный метод не сковывает движений, однако налагает ряд ограничений на краску или отражатель (водостойкость, контрастность по сравнению с окрасом животного). Кроме того, в процессе проведения теста отдельные маркеры могут сливаться в проекции с другими либо закрываться телом, что потребует дополнительных проверок при создании скелетной модели с учётом биомеханических характеристик испытуемого животного.

Безмаркерная технология, основанная на технологиях компьютерного зрения и распознавания образов, не требует специальных датчиков и оборудования. Это существенно ускоряет подготовку к записи движений и позволяет снимать сложные движения без риска повреждения датчиков или маркеров.

Процесс безмаркерного захвата движений требует больших вычислительных ресурсов, поэтому захват чаще всего происходит не в режиме реального времени. Сначала производится запись видео, а затем – обработка и вычисление интересующих параметров.

Ключевым преимуществом разработки системы на основе безмаркерной технологии служит ее невысокая стоимость – фактически, достаточно наличие только камеры и компьютера.

Накопление изменений за отрезок времени. Другой метод определения количества движений состоит в суммировании изменений яркости каждого элемента изображения за период измерения. В данном подходе видеоряд рассматривается как таблично заданная функция, каждый элемент которой представляет собой матрицу значений яркости пикселей изображения, а последовательность значений яркости каждого отдельного пикселя образует вектор, в котором количество движения может быть представлено как производная, выраженная в конечных разностях (рисунк 3).

Анализ частотного распределения. Данный метод также использует представление последовательности яркости каждого пикселя как вектора значений (рисунок 3), но для определения количества движений используется разложение отрезка вектора в ряд Фурье с последующим выделением и анализом амплитуд характерных частот, отвечающих за фазы покоя, поддерживающих движений либо активных действий.

Выводы. Применение вычислительной техники в экспериментальной медицине и физиологических исследованиях является на сегодня одним из важных направлений развития рынка программного обеспечения. При этом построение биологически достоверной и достаточно подробной математической модели человеческого организма на сегодняшний день представляется малореальной задачей, потому для оценки влияния новых фармакологических средств широко используются лабораторные животные.

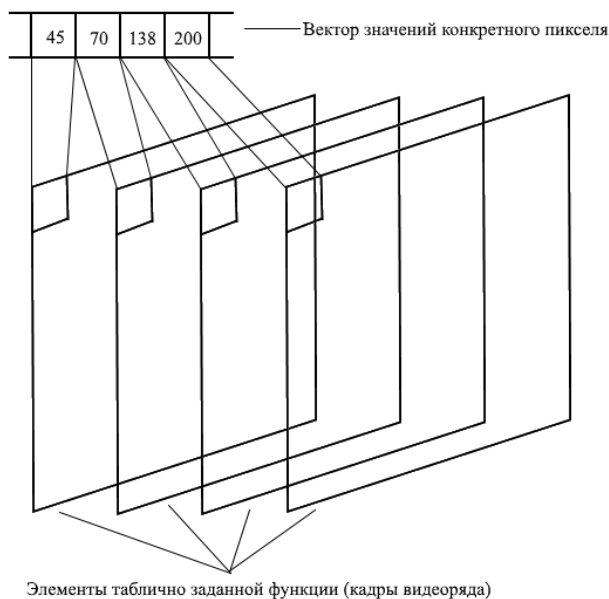


Рисунок 3. Представление видеоряда как таблично заданной функции

Для решения поставленной задачи возможно применение различных методов в зависимости от доступного оборудования и вычислительных

возможностей. Наиболее простыми и доступными будут методы накопления изменений за отрезок времени и анализ частотного распределения на отрезке вектора, как не требующие дополнительного оборудования и манипуляций с подопытным животным.

Список литературы

1. R.D. Porsolt, G. Anton, N. Blavet and M. Jalfre. Behavioural despair in rats: a new model sensitive to antidepressant treatments // European J. Pharmacol. #47 (1978) 379–391.

2. В.Н. Каркищенко, Г.Д. Капанадзе, С.Е. Деньгина, Н.В. Станкова. Разработка методики оценки физической выносливости мелких лабораторных животных для изучения адаптогенной активности некоторых лекарственных препаратов // Биомедицина, №1, 2011 г. с. 72–74.

УДК 004.942

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ЗАПАСАМИ

Байбуза А.В., Орлов Ю.К.

Донецкий Национальный Технический Университет
кафедра искусственного интеллекта и системного анализа
E-mail: fear-2008@mail.ru

Аннотация

Байбуза А.В., Орлов Ю.К. Разработка системы управления запасами. Рассмотрены задачи управления запасами. Разработанная подсистема содержит результаты расчетов для детерминированной модели, модели со случайным спросом, динамической модели управления запасами. Поиск оптимальной стратегии управления запасами базируется на применении математических методов.

Ключевые слова: *управление запасами, размер партии, время заказа.*

Постановка проблемы. „При управлении запасами любого товара следует ответить на два основных вопроса: когда пополнять запас и какой должен быть размер заказа на пополнение" [1]. С помощью математических методов можно выработать правила управления запасами. Метод решения состоит в построении математической модели рассматриваемой подсистемы и последующем изучении ее свойств. Задача управления запасами является одной из самых сложных в математическом смысле, которые существуют в области управления торговлей [2].

Анализ существующих подходов к решению задач управления запасами. Мировая практика выработала достаточно успешные подходы к решению проблемы планирования запасов, которые условно можно разделить на две группы: «от склада» и «от продаж» [3].

Подход «от склада». В основе классических западных складских систем управления запасами обычно лежит технология управления заказами SIC (Statistical Inventory Control), что использует статистические методы для моделирования спроса и времени пополнения товарных запасов (для производственных компаний – с учетом времени изготовления).

Подход «от продаж» дает более точные прогнозы использования запасов, которые обобщают разные договоренности и наблюдения, позволяют предприятию существенным образом улучшить качество управления запасами за счет дополнения «статистических» моделей планирования.

Назначение подсистемы и поставленные задачи. В процессе разработки подсистемы управления запасами должны быть решены следующие задачи:

- проанализированы существующие модели управления запасами;
- проанализированы методы поиска оптимальной стратегии функционирования управления запасами;
- разработана полная модель подсистемы управления запасами, которая бы включала модели управления запасами как для материалов со стационарным спросом, так и для материалов со динамично развивающимся спросом (соединены два подхода по управлению запасами);
- проверена разработанная подсистема для конкретного промышленного предприятия;
- выполнено сравнение стратегии функционирования, найденные с помощью упрощенной и более точной моделей и выяснить, достаточны ли полученные преимущества для обоснования более сложной модели;
- сделать вывод о выборе рекомендуемой стратегии управления запасами.

Разрабатываемая подсистема управления запасами должна учитывать наблюдение за динамикой запасов, а также сведения о запланированных объемах спроса в будущем, должна быть пригодной к применению как для предприятий со стационарным спросом на запасы, так и для динамично развивающихся предприятий. Подсистема поможет определить предприятию лучшую стратегию функционирования и определить – когда пополнять запас и какой должен быть размер заказа на пополнение, что повысит эффективность используемых ресурсов.

Принципы, использованные для построения системы управления запасами промышленного предприятия.

Связанные с работой складских систем расходы, играют важную роль в выборе стратегии функционирования. На стратегию функционирования влияют, очевидно, только расходы, которые меняются с изменением стратегии функционирования. Не зависящие от применяемой стратегии, расходы не следует учитывать при анализе, если они являются критерием выбора стратегии функционирования.

Существует пять основных типов расходов, которые могут повлиять на выбор стратегии функционирования [4]. К ним относятся:

- расходы, связанные с поставкой;
- расходы содержания запасов;
- расходы выполнения заказов потребителей;
- расходы, связанные с дефицитом запасов, когда требования, которые поступают, не могут быть удовлетворены;
- расходы сбора и обработки данных по управлению складской системой независимо от других.

Вывод: при построению математической модели подсистемы управления запасами следует ограничиться простой складской структурой, учитывать только те затраты, которые зависят от стратегии функционирования, учитывать конкуренцию между запаасаемыми материалами при ограничении на материал, из которого изготавливаются изделия, или при конкуренции на площадь склада. Для этого будем пользоваться удельным показателем.

Разработка прикладного программного обеспечения подсистемы.

Проанализировав поставленные задачи, разработана программа управления запасами промышленного предприятия. Эта программа обращается к серверу баз данных и работает с созданною базой данных Planov, обращается к существующей уже базе данных материальной бухгалтерии Buh_new и серверу Cache для получения учетного состава участка.

При запуске программы появляется главная форма, вверху которой расположенное меню приведенное на рисунке 1.

Управление запасами промышленным предприятием

Справочники Сервис План работ Выбор стратегии управления Отчеты Выход

Рисунок 1 – Главное меню

Пункт меню «Справочники» позволяет вводить данные в следующие справочники:

- Справочник материалов;

- Справочник Основных видов работ;
- Справочник Дополнительных работ;
- Справочник Материалов крепления;
- Справочник Элементов крепления.

А также просматривать Справочник групп материалов. Редактирование не разрешено, поскольку должно существовать полное соответствие данным в материальной бухгалтерии. Этот пункт меню приведен на рисунке 2.

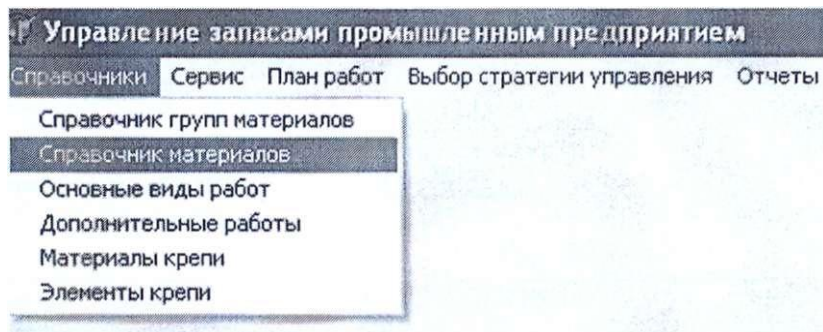


Рисунок 2 – Пункт меню «Справочники»

Следующий пункт меню «Сервис», что приведен на рисунке 3.

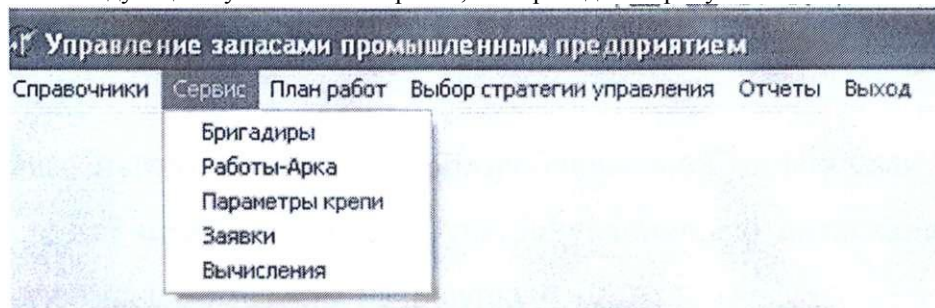


Рисунок 3 – Пункт меню «Сервис»

Он позволяет выполнить:

1. Изменение списка бригадиров на участках.
2. Установить соответствие видов работ и расходов арочного крепления.
3. Изменить расчетные показатели о расходах арочного крепления для проведения выработки бригаидами участков.
4. Пересматривать, менять заявки участков на материалы.

5. Установить точность вычислений, количество металла в наличии, число месяцев для планирования.

Следующий пункт «План работ» позволяет вводить план проведения подготовительных работ и приведен на рисунке 4.

Рисунок 4 – Форма пункта меню «План работ»

Основным пунктом меню для разрабатываемой подсистемы управления запасами является пункт «Выбор стратегии функционирования», по нажатию на который, появляется форма, которая приведена на рисунке 5.

Рисунок 5 – Форма пункта меню «Выбор стратегии функционирования»

Эта форма позволит при отсутствия данных о материале занести их.

По нажатию кнопки «Выполнить» запускаются на выполнение три рабочие потока, которые параллельно ищут стратегию функционирования для детерминированной модели, модели со случайным спросом, динамической модели управления запасами.

По завершению выполнения потоков в базе данных Planov формируются данные соответственно в таблицах Determtable, Sluch, Dynamic. А также текстовые файлы с результатами расчетов.

Основным результатом выполнения рабочих потоков является текстовое сообщение с применяемой рекомендуемой стратегией управления запасами.

Следующий пункт меню «Отчеты», позволяет сформировать отчеты для детерминированной модели управления запасами, модели со случайным спросом, динамической модели (рис. 6).

Пример отчета для модели со случайным спросом приведен на рисунке 7.

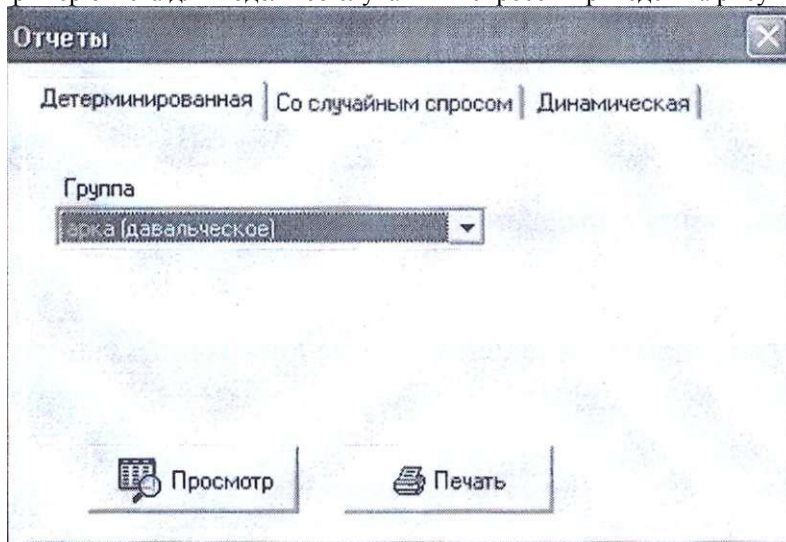


Рисунок 6 – Форма для печати отчетов

**План управления запасами
модель управления запасами со случайным спросом**

для группы материалов

арка (давальческое)

Наименование материала	Оптимальный размер заказа	Точка подачи заказа
АПЗ-18,3 тип.	79	365
АПЗ-18,3 уд.	74	345
АПЗ-13,8 тип.	92	417
АПЗ-13,8 уд.	77	324
АПЗ-11,2	56	113

Рисунок 7 – Отчет для модели управления запасами со случайным спросом

Пример текстового файла, сформированного в процессе поиска стратегии управления для модели со случайным спросом, приведен на рисунке 8.

АПЗ-18,3 тип.	82	365
АПЗ-18,3 уд.	77	345
АПЗ-13,8 тип.	95	417
АПЗ-13,8 уд.	76	325
АПЗ-11,2	54	114

Число итераций для поиска оптимальной стратегии : 0
 Время выполнения : 0:00:00
 Средние годовые издержки : 15812592,6510784

Рисунок 8 Файл с результатами расчетов для модели со случайным спросом

Выводы. Разработанная подсистема управления запасами, базирующаяся на обоснованном выборе оптимальной стратегии функционирования, руководствуясь анализом результатов по применению разных математических моделей управления запасами годится для применения на динамично развивающихся предприятиях, а также для предприятий, которым оптимальнее использовать статические модели управления запасами. Разработанная подсистема содержит результаты расчетов для детерминированной модели, модели со случайным спросом, динамической модели управления запасами. Поиск оптимальной стратегии управления запасами базируется на применении

математических методов. Критерием выбора оптимальной стратегии управления запасами является показатель средних годовых затрат.

Данная подсистема позволит предприятию ответить на два основных вопроса для запаасаемых материалов: „Когда и сколько надо заказывать?“, что позволит минимизировать расходы на содержание, учет материалов и расходы из-за отсутствия материалов на складе.

Список литературы

1. Кремер Н.Ш. Исследование операций в экономике. – М: Юнити, 2002. -407 с.
2. Байбуза А.В., Орлов Ю.К. Разработка моделей оптимизации уровня запасов производства. – Информатика, управляющие системы, математическое и компьютерное моделирование в рамках II форума «Инновационные перспективы Донбасса» (ИУСМКМ - 2016): VII Международная научно-техническая конференция, 26 мая 2016, г.Донецк: / Донец. национал. техн. ун-т. - Донецк: ДонНТУ, 2016. – с. 567-72.
3. Методика оперативного управления запасами/ Электронный ресурс. Способ доступа: URL: www.itrealty.ru/analit/book16.htm
4. Хедли Дж., Уайтин Т. Анализ систем управления запасами. Перевод с англ. –М: Наука, 1969. –511 с.

УДК 004.932.2

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ПРОГРАММНОГО КОДА НА ОСНОВЕ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ СПЕЦИФИКАЦИЙ ТРЕБОВАНИЙ

Небесная В.С., Ермоленко Т.В.

Донецкий национальный технический университет кафедра
искусственного интеллекта и системного анализа E-mail:
nebesny93

Аннотация

Небесная В.С., Ермоленко Т.В. Автоматическая генерация программного кода на основе естественно языковых спецификаций. В работе рассмотрены основные подходы для автоматической генерации программного кода. Подробно рассмотрен предлагаемый алгоритм нахождения генерации кода. Проиллюстрировано выполнение основных шагов алгоритма. По результатам экспериментов можно сделать выводы о точности и быстродействии предложенного подхода. В качестве источника тестовых выбран набор команд на естественном языке для генерации программного кода.

Ключевые слова: автоматическая генерация кода, естественно языковые спецификации, морфологический анализ.

Общая постановка проблемы. Технология генерации кода является эффективной и мощной техникой с потенциально безграничными возможностями.

Идея генерации кода не является новой. Например, формирование машинного кода из программы на ассемблере было придумано давно. Компиляторы языков высокого уровня являются, по сути, генераторами низкоуровневого кода. На этапе появления языков высокого уровня под генерацией кода подразумевалась именно генерация машинного кода. Сейчас же все чаще под этим имеют в виду генерацию исходного кода на одном из языков высокого уровня. Условия эффективного применения генераторов

Генераторы лучше всего применимы в проблемной области, требующей выполнения множества коротких, схожих итераций.

Автоматическая генерация программного кода предоставляет преимущества описанные ниже.

1. структурированность и согласованность исходного кода приложения (сгенерированного машиной). Генератор позволяет с точностью до символа соблюдать правила написания кода, применяемые в организации.

2. высокое качество кода и облегченное исправление ошибок.

3. гибкость к изменениям. Если требования меняются, можно только обновить шаблоны, генератор и/или метаданные и сделать массивированный вывод новой версии кода.

4. единый источник информации о приложении. В приложениях, которые пишутся вручную, простое изменение имени таблицы или столбца может повлечь за собой серию изменений во многих участках кода. При генерации же метаданные находятся в отдельном источнике и поэтому достаточно изменить название таблицы или столбца в том месте, где определяется схема, и пересоздать код;

5. наглядность бизнес логики. В приложениях, которые пишутся вручную, смысл бизнес логики теряется в сотнях и тысячах строк программного кода. Генераторы же используют файлы метаданных, в которых видна вся структура программируемой предметной области, её поведение, исключения и т.д. Гораздо легче понять, что делает приложение на высоком уровне абстракции.

6. Метаданные могут использовать эксперты в предметной области приложения;

7. Код сгенерированный по естественному языку, является сам себе документацией.

Исследование. Для решений задачи автоматической генерации программного кода на основе естественно языковых спецификаций существует несколько методов [1,2,3]. В основном их можно разделить на следующие группы:

Генерация кода с применением шаблонов. В этом случае повторяющийся код хранится в шаблонах. Они также хранят инструкции или программный код для встраивания определенным способом метаданных в шаблон. Примерами являются технологии XSLT и T4, которые мы рассмотрим в нашем курсе;

прямой подстановкой текста. Генерируемый код представляется в виде набора строк, который формируется после манипуляций текстом, результат выводится в текстовый файл. Этот метод является самым простым и интуитивно понятным для программиста. При применении этого метода шаблоны существуют, но они запрограммированы в коде генератора;

Объектным моделированием. Элементы программного кода моделируются в генераторе программными объектами. Примером является технология CodeDom. Такие понятия как классы, типы, объекты, переменные, выражения и т.д. в ней представлены классами из пространства имен **CodeDom**. Манипулируя объектами этих классов можно создавать программный код.

Описание алгоритма. Так как по ряду причин мы не можем выбрать одно единственно верное решение для генерации кода, был разработан алгоритм для генерации программного кода с использованием шаблонов.

На вход поступает текст на естественном языке на котором описывается поведение системы. На рисунке 1 приведены основные шаги алгоритма.

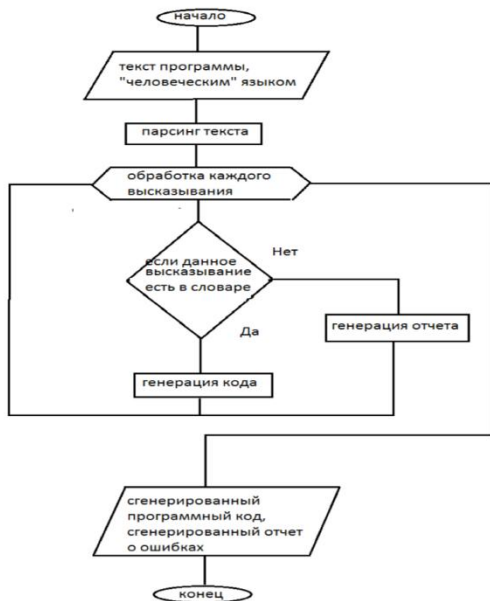


Рисунок 1 – Основные шаги алгоритма

На первом этапе, необходимо разбить весь текст на минимальные точные, понятные и законченные части которые будут соответствовать одному единственному шаблону кода, однако на практике было установлено, что выделение одного предложения целиком не подходит так как конечный пользователь может записать одну и ту же команду двумя похожими предложениями например «Создать переменную «temp» » и «Создать проименованную область памяти с именем «temp»», обе фразы являются синонимами и для них соответствует один шаблон но хранить такие большие предложения со временем станет затруднительно и при расширении базы знаний системы поиск необходимого шаблона затянется. Поэтом имеет смысл сначала искать только главное слово в предложении, как показала практика в большинстве случаев таким словом будит является глагол, например, «Создать, удалить, умножить и т.д.». Так как наше главное слово может фигурировать в нескольких высказываниях, например, (Создать переменную, создать Файл и т.д.) необходимо найти для него самое подходящее высказывание по обязательным атрибутам присутствующих в предложении, поиск ведется от самого сложного до самого легкого высказывания, иначе в противном случае система может обязательные атрибуты распознать как «мусор»

После нахождения необходимого высказывания сохраняем метаданные (имена файлов, переменных и т.д.). Если система не нашла высказывание, то генерируется отчет о ошибке, если система нашла высказывание, то по таблице соответствий находим необходимый шаблон на выбранном языке программирования. На выходе получаем готовый программный код.

На рисунке 2 приведены результаты выполнения описанных выше шагов для одного из изображений.

```

сотворить переменную "name1".
создать переменную "name2".
переменную "name1" инициализировать "1".
Присвоить переменной "name2" значение "2".
Произведение чисел "name2" на число "3".
Сложить "name1" и "1".
Разделить переменную "name2" на "2".
"name1" = "name2" плюс "1".
Сумма чисел "name2" и "1".
Разница чисел "name1" и "name2".
Разделить "name1" на "3".
удалить "name1".
    
```

Рисунок 2 – Результаты выполнения шагов алгоритма.

```

Sname1 = null;//сотворить переменную "name1"
Sname2 = null;// создать переменную "name2"
Sname1 = 1 // переменную "name1" инициализировать "1"
Sname2 = 2 // Присвоить переменной "name2" значение "2"
Sname2 *= 3 // Произведение чисел "name2" на число "3"
Sname1 += 1 // Сложить "name1" и "1"
Sname2 /= 2 // Разделить переменную "name2" на "2"
Sname1 = Sname2 + 1 // "name1" = "name2" плюс "1"
Sname2 += 1 // Сумма чисел "name2" и "1"
Sname1 -= Sname2 // Разница чисел "name1" и "name2"
Sname1 /= 3 // Разделить "name1" на "3"
unset( Sname1 );// удалить "name1"
    
```

Рисунок 3 – Результаты работы системы.

Результаты экспериментов. В качестве источника данных использовался текст на естественно понятном языке (обработано 100 текстов, размер каждого 10 предложений). На сто текстов погрешность была в 6 текстах. Результаты работы алгоритма для автоматической генерации программного кода приведены в таблице 1. Большинство ошибочных определений происходили неоднозначности высказываний.

Таблица 1. Результаты работы алгоритма

Верных, %	Ошибок, %	Min время, мс	Max время, мс	Среднее время, мс
93,6	6,4	125	432	156

Выводы. Технология генерации кода является эффективной и мощной техникой с потенциально безграничными возможностями. Также необходимо отметить, что генератор кода не является волшебной палочкой, решающей все проблемы. Если допущены ошибки на этапе проектирования, или идея проекта, скажем так, не самая лучшая, то никакой генератор не может гарантировать успех проекта. Генератор кода - это инструмент, а грамотное применение инструмента - задача того, кто им пользуется.

Список литературы

1. Wildes, R., Asmuth, J., Green, G., Hsu, S., Kolczynski, R., Matey, J., McBride, S. / A system for automated iris recognition. – Sarasota, FL: Proceedings IEEE Workshop on Applications of Computer Vision. – 1994. – P. 121-128.
2. Ritter, N. Location of the pupil-iris boarder in slit-lamp images of the cornea [Электронный ресурс]. –Режим доступа:
https://www.researchgate.net/publication/221356058_Location_of_the_Pupil-Iris_Border_in_Slit-Lamp_Images_of_the_Cornea.
3. Gui, F. Iris localization scheme based on morphology and gaussian filtering / F. Gui, L. Qiwei // Third Intern. IEEE Conf. on Signal-Image Technologies and Internet-Based System. Shanghai, China. –2007. – P.798-803.

**ИНФОРМАТИКА,
УПРАВЛЯЮЩИЕ СИСТЕМЫ,
МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ
МОДЕЛИРОВАНИЕ
(ИУСМКМ – 2017)**

**Сборник материалов VIII Международной
научно-технической конференции
в рамках
III Международного Научного форума
Донецкой Народной Республики**

25 мая 2017г.

**Редактор Ю.К. Орлов
Дизайнер В.В. Харитонова**

Подписано к печати 25.05.2017 г.

Формат А5. Бумага 80 г/м²

Гарнитура «Times New Roman». Печать лазерная.

Страниц 802. Переплет мягкий.

Уч.-изд. л. 35,3 Усл. печатн. лист 25,2.

Тираж 100 экземпляров

Издательство «Донецкая политехника»

Тел.: +38 (062) 305-35-67

Интернет: <http://science.donntu.org/>