

**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 – 2016)**

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

26 мая 2016г.

г. Донецк, ДонНТУ – 2016

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; редкол. А.Ю. Харитонов и др. - Донецк: ДонНТУ, 2016. - 624 с.

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, проректор по научной работе, и.о. ректора 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; Харитонов А. Ю., ст. пр. каф. КМД; Григорьев А.В. к.т.н., доц. каф. ПИ; Харитонова В. В, ст. пр. каф КМД.

Адрес редакционной коллегии:

83001, г. Донецк, ул. Артема, 58,

Донецкий национальный технический университет,

4 учебный корпус, факультет компьютерных наук и технологий,
кафедра компьютерного моделирования и дизайна, ком.4.41.

E-mail: iuskm@donntu.org

<http://iuskm.donntu.org>

© Донецкий национальный технический университет, 2016

СОДЕРЖАНИЕ

СЕКЦИЯ № 1 ТЕОРЕТИЧЕСКАЯ И ПРИКЛАДНАЯ МАТЕМАТИКА.....10

Матюх Ю.Д., Ефименко К.Н. Особенности применения языка описания аппаратуры SystemC	11
Дегтярев С.П., Коломиец Ю.В. Современные задачи со свободной границей математической физики	16
Прокопенко Е.В., Масло С.В., Уманский В.В. Создание алгоритма управления перемещением породы на отвале	18
Масло С.В., Прокопенко Е.В., Высоцкий С.А. Применение линейного программирования для нахождения максимального компонента состава пород по пластам.....	24
Грановский Я.И. Усреднение случайных функционалов от решений стохастических уравнений	28
Ананьева А.Ю., Будыка В.С. Спектральная теория оператора Бесселя на конечном интервале и полуоси.....	31
Добровольский Н.Н. Об одном вопросе из элементарной теории чисел.....	35
Рудакова О.А. Условия коэрцитивности интегранта Γ -предельного функционала для последовательности интегральных функционалов, определенных на различных весовых пространствах Соболева.....	37
Кузнецов П.А., Степанов О.А., Масло С.В., Клюковкин В.Р. Разработка структуры генератора маркерных сеток для цифрового электронно-оптического преобразователя.	44
Решетников А.В. О конгруэнциях частичных универсальных алгебр	49
Іє О.М. Ігрові методи в теорії статистичних рішень	52
Костенко А. С., Маламуд М. М., Натягайло Д.Д. Индексы дефекта матричного оператора Шредингера с дельта-взаимодействиями.	58
Кожухов И.Б. Некоторые вопросы теории полигонов над полугруппами	62
Свентковский В.А. Построение шестиугольников, состоящих из любого числа n любых равных треугольников, начиная с некоторого; и построение восьмиугольников, состоящих из любых n равных, не равнобедренных треугольников.	69
Вечтомов Е.М., Лубягина Е.Н. О полукольцах непрерывных частичных функций	76
Вечтомов, Е.М. Петров А.А. О конгруэнциях на мультипликативно идемпотентных полукольцах	86

Чупраков Д.В. Криптографические алгоритмы над абстрактными полукольцами. ...93	
СЕКЦИЯ № 2 ПРОГРАММНАЯ ИНЖЕНЕРИЯ.....	101
Егоров А.А., Чернышова А.В. Программная система встраивания информации в изображения с использованием стеганографических и криптографических алгоритмов	102
Чернышова А.В., Кубашевский Д.В. Подсистема защиты авторского права в программном обеспечении.	110
Рычка О.В. Описание и программная реализации методов обработки данных для повышения точности прогнозирования	117
Калиниченко Г.А. Разработка графической оболочки для операционной системы Android.	125
Бычкова Е.В., Нестеренко В.С. Анализ проблем построения ЭВМ с использованием троичной логики	131
Пикалев Я.С. Применение систем синтеза речи.....	136
Фоминых И.Д., Рудак Л.В., Моргайлов Д.Д. Разработка алгоритма генерации венгерских кроссвордов.....	142
Тырса С.В., Хубеджев Д.П., Моргайлов Д.Д. Проектирование архитектуры централизованного кроссплатформенного протокола сетевой печати	146
Немцов Д.И. Автоматизация исследований в области сегментации и распознавания речи	151
Ходич Д.В., Грищенко В.И. Программная система учета НТМ ДонНТУ	154
Багимова А. Д., Губенко Н. Е. Защита web-ресурса оценки и ранжирования стран мира по показателям их развития	158
Воробьёв Л.О., Полетаев В.А., Моргайлов Д.Д. Разработка интегрированной CASE-системы для обучения студентов программированию	163
Маруга М.М. Современные средства кроссплатформенности для разработки и использования ПО	168
Черкасс Д.В., ст. пр. Кравец Т.Н. Разработка и исследование метода контроля карты рабочей среды агента.	174
Сидоренко А.С., Коломойцева И.А. Анализ существующих подходов к разработке программ для анализа прогноза погоды	177
Швыров В.В. Модели информационной безопасности.....	182
Шумский А.А., Бычкова Е.В. Использование облачных технологий в игровой индустрии.....	188
Кацель С.Л. Разработка приложения для организации спортивных мероприятий по системе с одним (кубковая) или двойным выбыванием (double elimination).193	

Медгаус С.В., Чернышова А.В. Проектирование обфускатора для языка JavaScript.....	197
Серёженко О.А., Коломойцева И.А. Обзор программ для чтения текстовых файлов для ОС Android	203
Бакаленко В.С., Федяев О.И. Речевой интерфейс для интеллектуализации ввода компьютерных программ	209

СЕКЦИЯ № 3 СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ219

Калинина Е.С., Чернышова А.В. Разработка модуля аутентификации PDM-подсистемы в CAD-компонентах САПР с открытым кодом	220
Скаженик В.Б., Балагуров А.В. Пространственная визуализация угольной шахты	226
Емец А.И. Системы автоматизированного проектирования (САПР) шахт.	230
Мёд В.П., Григорьев А.В. Проектирование технических объектов с помощью баз знаний в CAD/CAE системах на примере CATIA.....	237
Клокова М.В., Чернышова А.В. Подсистема криптографической защиты файлов проектов для САПР.....	241
Роговец Е.А., Григорьев А.В. Разработка программных средств, повышающих эффективность решения задачи параметризации в САПР AUTOCAD	246
Жусупов Е.В., Жусупов М.В., Григорьев А.В. Графическая база данных в AutoCAD, назначение и структура	248

СЕКЦИЯ № 4 ИНФОРМАЦИОННЫЕ УПРАВЛЯЮЩИЕ СИСТЕМЫ И ТЕХНОЛОГИИ253

Главацкий С.Т., Михалев А.В. Технологии электронного обучения: опыт МГУ имени М.В.Ломоносова	254
Давыденко Д.П., Губенко Н.Е. Метрики оценки качества веб-систем.	260
Штанько А.А., Губенко Н.Е. Политика защиты систем дистанционного обучения. Выполнен анализ сути политики защиты информации	265
Цесаренко Д.С., Харитонов А.Ю. Применение современных web-технологий к автоматизации учета потребления энергоресурсов бюджетной сферы города Донецка	270
Двойкин А.А., Привалов М.В. Архитектура распределённой системы управления целеориентированными проектами.....	274
Толстых В.К., Киселева Л.Н. Распределённая информационная система совместной поддержки единой базы граждан.....	280

Лысенко В.С., Землянская С.Ю. Компьютеризированная подсистема учета и планирования выполнения заказов для малого предприятия по изготовлению печатных плат	287
Харитонов Ю.Е. Спецификации номенклатуры – хранение и обработка в различных ERP-системах	293
Симоненко А.А., Привалов М.В. Разработка веб-ориентированной подсистемы доступа к медицинским изображениям	297
Васильев В.В., Привалов М.В., Пряхин В.В. Разработка веб ориентированной подсистемы планирования и учёта индивидуальной нагрузки преподавателя ВУЗа	301

СЕКЦИЯ № 5 МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ..... 306

Москалюк Т.Г, Харитонов А.Ю. Имитационное моделирование в процессе проектирования станков и станочных систем	307
Якимов А.Н, Мартыненко Н.В. Использование метода векторных конечных элементов для моделирования магнитных полей в неразрушающем контроле .310	
Карабчевский В.В., Кудимов М.И. Сравнение двух способов решения метрических задач средствами графического редактора AutoCAD	319
Давыденко И.П., Малютина Т.П. Точечное уравнение синусоиды с осью в виде окружности и его применение при построении поверхностей технических форм методом подвижного симплекса	324
Степович М.А. Математическое моделирование явлений тепломассопереноса в диагностике материалов	328
Бабенко Б.Г., Беловодский В.Н. Информационная система для первичной обработки временных рядов.....	335
Мартосенко А.Г., Пчёлкин В.Н. 3D-моделирование установки прямого восстановления железа средствами 3ds Max	337
Койбаш А.А., Кривошеев С.В. Прогнозирование траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники	343
Кийко А.В., Кийко С.А. Синтез оптимального узкополосного нерекурсивного цифрового фильтра	347
Дяченко О.Н., Дяченко В.О. Альтернативный метод укорачивания циклических кодов	353

СЕКЦИЯ № 6 СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА..... 360

Рудова Е.А., Жудин А.Ю. Применение генетического алгоритма при построении игрового искусственного интеллекта коллекционной карточной игры.....	361
Васяева Т.А., Струг С.С. Получение продукционных правил на основе ГП...368	368
Нечепуренко М.С., Ольшевский А.И. Применение различных алгоритмов в решении задачи Штейнера.....	372
Потёмкин А.С. Нахождение контура фрагментов лица на мимическом изображении.....	378
Сутковая М.И., Бочаров Д.М. Исследование методов описания контура объекта на изображении.....	382
Лашенко К.С., Гудаев О.А. Применение искусственной нейронной сети для дешифрации лексикографических кодов расширенной реальности.....	388
Савкин М.С., Савицкая И.В. Исследование методов усовершенствования игры «Жизнь».....	394
Захаров А.В., Сорокин Р.А. Исследование методов классификации графических примитивов на фотоизображении.....	398
Минков О.В. Распознавание многоугольников на изображении с помощью алгоритма BFOA.....	404
Потопахин А.А., Гудаев О.А. Разработка аппаратно-программного комплекса ультразвуковой идентификации объектов.....	407
Погребницкий Б.С., Семенова А.П. Двумерная модель компьютерной игры в жанре пошаговой стратегии.....	412
Ольшевский А.И., Стась А.В. Разработка обучающего пакета программ работы с контейнерными классами.....	417
СЕКЦИЯ № 7 МЕДИАИНДУСТРИЯ И ДИЗАЙН.....	421
Кладько Д.С., Харитонова В.В. Особенности профессии проектировщика интерфейсов.....	422
Титаренко О.Ю., Павлий В.А. Разработка клиент-серверного приложения контроля эффективности рабочего времени.....	427
Пересекин А.В. Харитонова В.В. Использование семиотики в печатной рекламе.....	432
Лоленко В.А., Хоруженко А.С. Основные требования к расписанию и методы его составления в условиях высшего учебного заведения.....	438
Ефремченко И.А., Харитонова В.В. Некачественный дизайн как одна из проблем юзабилити сайта.....	443
Сидорчук В.И., Харитонова В.В. Японская мультипликация. Особенности японской мультипликации.....	449

Юлина Ю.А., Харитонов А.Ю. Применение инфографики к отображению аналитической информации энерго- и ресурсопотребления.....	454
Еремеев В.В., Павлий В.А. Анализ приложений для визуализации математических выражений на веб-ресурсах	459
Гаранжа А.В., Губенко Н.Е. Разработка структуры для web-ресурса «Футбольный портал Донецкой Народной Республики»	464
Овсепян Т.Н., Павлий В.А. Разработка информационной системы для верификации и публикации студенческих персональных сайтов	469
Головина Е.О., Губенко Н.Е. Разработка структуры информационного портала «Traditional art».....	474
Ерёменко П.С., Хоруженко А.С. Создание сайта-портфолио Ерёменко П.С., демонстрирующего возможности работы в React.js, Relay, GraphQL, Babylon.js, Node.js	479
Бойко В.В., Губенко Н.Е. Методы защиты веб-сайта журнала «САИТ» от атак через интернет	483
Родненко Е.С., Губенко Н.Е. Анализ систем управления содержимым Joomla, Drupal и WordPress и выбор оптимальной CMS для создания интернет-портала.	487
Мотренко В.В. Создание сайта поклонников творчества донецких художников. Рассмотрена задача создания сайта о творчестве донецких художников	493
Шевляков А.И., Губенко Н.Е. Анализ и выбор алгоритмов наложения ЦВЗ в мультимедийных файлах	499
Жиленкова Е.Ю., Харитонova В.В. Принципы композиционного построения в современных плакатах	503
Давыденко Д.П., Харитонova В.В. Виды инфографики	513
Касияненко Д.И., Харитонova В.В. Некоторые правила применения цветовых комбинаций в Web-дизайне.....	517
Ольшевский А.И., Стась А.В. Разработка обучающего пакета программ работы с контейнерными классами	523
СЕКЦИЯ № 8 КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ	526
Селеменев В.Д., Теплинский С.В. Система контроля микроклимата жилого дома	527
Горин Н.А., Струнилин В.Н. Разработка системы ориентирования в пространстве с использованием модуля GY-531	534
Скиба В.Е., Краснокутский В.А., Меренкова Л.Л. Исследование работы стандартного оптимизатора реляционной СУБД и применение основных методов лексической оптимизации запросов в рамках АСУ ДонНТУ	539

Каховский Д.А., Завадская Т.В. Разработка и исследование алгоритмов управления модуля энергосбережением системы "умный дом".....	544
Мохначёв В.В., Зинченко Ю.Е., Коваль А.В. Разработка базовой модели квадрокоптера на базе FPGA	547
Бурлака М.В., Теплинский С.В. Система приема посетителей жилого дома.	552
Костенко Д.А., Теплинский С.В. Особенности восстановления удаленных файлов и каталогов в файловой системе FAT32	559
Иванов В.Ю., Завадская Т.В., Меренкова Л.Л. Исследование оптимальных запросов при выборке многосвязных данных больших объемов в реляционных базах данных.....	562
Дерксен Е.А., Завадская Т.В., Меренкова Л.Л. Исследование оптимальной структуры базы данных в рамках автоматизированной системы приемной комиссии ДонНТУ	567
Прищепа В.В., Краснокутский В.А., Меренкова Л.Л. Перспективы применения технологий искусственного интеллекта при разработке системы поддержки принятия решений в составе программного комплекса автоматизации формирования нагрузки преподавателей кафедры	572
Плахин С.В., Яременко П.А., Краснокутский В.А. Вопросы организации аналого-цифрового преобразования в устройствах на базе микроконтроллеров ADuC841	577
СЕКЦИЯ № 9 СИСТЕМНЫЙ АНАЛИЗ И УПРАВЛЕНИЕ.....	584
Громоклицкая М.Н. Основные функции и задачи системы учета реестра продаж мототехники в мотосалоне.	585
Ванжа Т.В., Орлов Ю.К. Разработка моделей и методов управления горным предприятием в условиях экономических рисков	591
Бровкина А.С., Орлов Ю.К. Разработка моделей и методов синтеза информационно-справочной системы диагностирования и лечения болезней для медицинских заведений.....	597
Милая А.С., Савкова Е.О. Формализация задачи принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости.	603
Вергелес А.А., Копытова О.М. О сравнении поведения ОДк-эталона и автоматов, порождаемых его локальными преобразованиями	614
Байбуза А.В., Орлов Ю.К. Разработка моделей оптимизации уровня запасов производства	618

Секция № 1
Теоретическая и
прикладная математика



УДК 004.274

Особенности применения языка описания аппаратуры SystemC

Матюх Ю.Д., Ефименко К.Н.

Донецкий национальный технический университет
студент гр. ЭЛМ-156¹, доцент кафедры прикладной математики²
E-mail: opera613@gmail.com

Аннотация

Матюх Ю.Д., Ефименко К.Н. Особенности применения языка описания аппаратуры SystemC. Рассмотрены особенности применения языка описания аппаратуры SystemC при реализации аппаратных и программных моделей высокого уровня абстракции. Приведен пример программного кода описания логического элемента и результаты моделирования.

Общая постановка проблемы. В настоящее время цифровые устройства находят массовое применение во всех сферах жизнедеятельности человечества. Поэтому проектирование и разработка электронных цифровых устройств остается актуальной научной и практической задачей. Одним из важнейших этапов проектирования является верификация разрабатываемых логических интегральных схем [1]. При решении этой задачи широко применяется язык описания аппаратуры HDL (Hardware Description Language) [2]. В зависимости от метода последующей реализации проектируемой интегральной системы выбирается наиболее удобный язык описания проекта. К ним относятся VHDL, Verilog, AHDL и др. Одним из основных отличий языков описания аппаратуры от традиционных языков программирования (C++ и др.) является возможность ветвления, распараллеливания потоков данных, без чего, как правило, невозможно проектирование высокопроизводительных микросхем. В результате при проектировании систем высокого уровня сложности возникает необходимость использования наряду с текстовыми описаниями графического интерфейса, позволяющего визуализировать параллельные потоки информации и тем самым сократить количество ошибок, время проектирования, значительно облегчить труд разработчика. Однако сложность электронных схем возрастает с каждым годом. В связи с этим разрабатывается все больше подобных языков, одним из которых является SystemC.

SystemC – язык описания аппаратуры, отличительной особенностью которого является то, что он представлен надстройкой языка C++, реализованной в виде отдельных библиотек [3]. Помимо стандартных конструкций и команд C++ в SystemC можно использовать конструкции, присущие HDL и позволяющие моделировать электронные системы разной степени абстрактности.

Таким образом, язык SystemC рекомендуется для описания абстрактных

аппаратных и программных моделей на высоком уровне абстракции, так как именно они легко могут быть использованы для быстрого создания интегральных схем.

Исследования. Язык описания SystemC является мощным средством для создания моделей цифровой аппаратуры. Библиотека классов включает в себя все необходимые конструкции, благодаря чему SystemC ничем не уступает языкам VHDL и Verilog.

В тоже время язык имеет ряд особенностей, отличающих его от C++. Например, функции позволяющие работать со временем, моделируя задержки реальных действий, таких как механическое воздействие и распространение сигнала, или подсчитывая время работы алгоритма в мили-, микро-, нано-, пико- или фемтосекундах. Или типы данных, такие как `sc_logic`, способный принимать значения 0, 1, X и Y; векторные типы данных, содержащие в себе определенное количество бит типа `logic (sc_lv<n>)` или целых чисел (`sc_int<n>/sc_uint<n>/sc_bigint<n>` и др.) [4].

Для иллюстрации особенностей и структуры языка опишем логический элемент «2И». Принцип работы элемента «2И» заключается в том, что на выходе элемента будет логическая единица только в одном случае – когда на оба входа подается логическая единица. Если хотя бы на одном из входов ноль, то и на выходе будет ноль.

В SystemC каждый модуль содержится в отдельном файле с расширением “.h”, который в дальнейшем подключается в виде библиотеки в файле с главной функцией (файл с расширением .cpp).

В файле, который содержит описание модуля элемента «2И», назовем его `2and.h`, описаны входящие и выходящие порты. Они служат для взаимодействия модулей между собой, как контакты на микросхемах. За портами следует описание события, с продолжительностью 20 наносекунд и функция определения значения на выходящем сигнале. Ниже приведен фрагмент описания модуля элемента:

```
#include "systemc.h"
SC_MODULE(and2)
{
    sc_in <bool> A, B;
    sc_out <bool> F;
    sc_event delay;
    void do_delay()
    {delay.notify(20, SC_NS);}
    void do_and2()
    {F.write( A.read() && B.read() );}
    SC_CTOR(and2)
    { SC_METHOD(do_delay);
      sensitive << A << B;
```

```
    SC_METHOD(do_and2);  
    sensitive << delay; }  
}
```

Для проверки корректности работы модуля «2И», необходимо организовать подачу входных сигналов, то есть сгенерировать входящие на этот модуль сигналы. Для генерации входных сигналов используется следующая функция:

```
#include "systemc.h"  
SC_MODULE(stim)  
{  
    sc_out <bool> A, B;  
    sc_in <bool> Clk;  
    void do_stim()  
    {  
        while(1)  
        {  
            A.write(false);  
            B.write(false);  
            wait();  
            A.write(true);  
            B.write(false);  
            wait();  
            A.write(false);  
            B.write(true);  
            wait();  
            A.write(true);  
            B.write(true);  
            wait();  
        }  
    }  
    SC_CTOR(stim)  
    {  
        SC_THREAD(do_stim);  
        sensitive << Clk.neg();  
    }  
}
```

Данный модуль чувствителен к срезу синхросигнала Clk, который будет принимать значения 0 и 1. Модуль реализован в main.cpp. Задача данного модуля – менять значения входных сигналов А и В, поступающих на логический элемент «2И». Метод wait() в данном случае приостанавливает работу процесса модуля до следующего переключения Clk из 1 в 0. Поэтому, задержка переключения сигналов А и В будет равна периоду синхросигнала.

Далее следует описание файла main.cpp, реализующего функцию sc_main(). Главная задача данной функции – это объявление всех объектов проекта (объект Stim и and2), соединение этих объектов посредством сигналов, а также запуск фазы моделирования:

```
#include "systemc.h"  
#include "and2.h"
```

```

#include "stim.h"
int sc_main(int argc, char* argv[])
{
    sc_signal <bool> A, B, F;
    sc_clock Clk("TestClock", 50, SC_NS, 0, 0, SC_NS);
    // Создание объекта класса stim и назначение его портам сигналов
    stim Stim1("Stimulus");
    Stim1(A, B, Clk);
    // Создание объекта класса and2 и назначение его портам сигналов and2
    a("AND2");
    a(A, B, F);
    // Создание файла трассировки и добавление сигналов
    sc_trace_file* Tf;
    Tf = sc_create_vcd_trace_file("traces");
    Tf -> set_time_unit(-12);
    sc_trace(Tf, A , "A" );
    sc_trace(Tf, B , "B" );
    sc_trace(Tf, F , "F" );
    sc_trace(Tf, Clk , "Clk" );
    // Запуск фазы моделирования от 0 до 300 нс
    sc_start(300, SC_NS);
    // Закрытие файла трассировки
    sc_close_vcd_trace_file(Tf);
    return 0;
}

```

Для завершения моделирования необходимо откомпилировать созданный проект и запустить исполняемый файл. В результате работы программы получена временная диаграмма работы логического элемента «2И», изображенная на рис.1.

Выводы. Преимуществом рассмотренного языка описания аппаратуры SystemC является возможность использования всех средств языка C++, который является наиболее популярным среди системных разработчиков и разработчиков программного обеспечения, что делает его привлекательным также и для разработок в области описания аппаратуры. В результате этого язык описания SystemC, интегрированный в C++, может быть использован как общий язык системных проектировщиков, инженеров программного обеспечения и проектировщиков аппаратных средств.

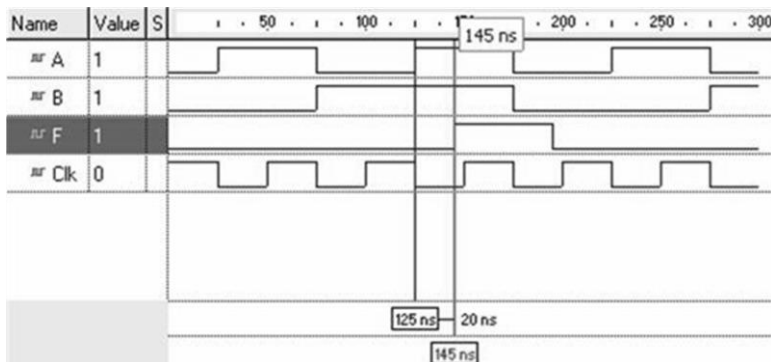


Рис.1. Временная диаграмма работы логического элемента «2И»

Однако основным недостатком SystemC, является то, что описание схем на языке C++ в отдельных случаях является слишком громоздким, например, описание одной и той же модели на языке Verilog может иметь в несколько раз меньше строк кода, чем на языке SystemC. Объем в особенности велик в случае создания конструкций в моделях, позволяющих имитировать задержки реальных элементов. Тем не менее, изучение и совершенствование SystemC имеет большое практическое значение.

Список литературы

1. Кравченко, В. Виртуальное прототипирование для аппаратно-программной верификации СБИС/ В. Кравченко, Д. Радченко// ЭЛЕКТРОНИКА: Наука, Технология, Бизнес, №7. – 2003. – С. 34-37.
2. Руководство к лабораторной работе «Разработка VHDL-описаний СБИС в подсистеме Renoir САПР FPGA Advantage». – Таганрог: Изд-во ТРТУ, 2002. – 24 с.
3. Black, David C. Systemc: From the Ground Up/ David C. Black, Jack Donovan. – Kluwer Academic, 2004. – 264р.
4. Язык SystemC [Электронный ресурс]/ Интернет-ресурс. – Режим доступа: <http://systemc.dax.ru/book/2.html>. – Загл. с экрана.

УДК 517.9

**Современные задачи со свободной границей
математической физики**

Дегтярев С.П., Коломиец Ю.В.

ГУ «Институт прикладной математики и механики», Донецк

E-mail: degtyar@i.ua

Аннотация

Дегтярев С.П., Коломиец Ю.В. Современные задачи со свободной границей математической физики. Приведены постановки некоторых современных задач со свободной границей в области дифференциальных уравнений с частными производными и математической физики и описаны открытые проблемы.

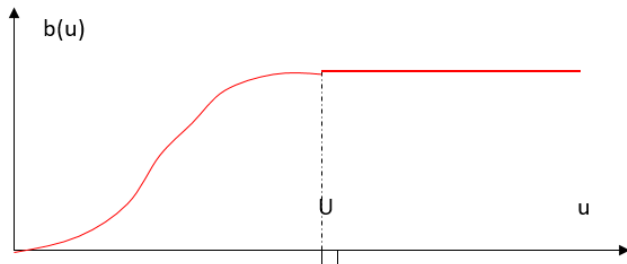
Общая постановка проблемы. Мы рассмотрим две открытые проблемы корректности современных математических моделей в теории фильтрации в пористой среде и в теории тонких пленок. Обе эти проблемы касаются корректности так называемых задач со свободной границей.

Эллиптико-параболическое уравнение и соответствующая задача со свободной границей.

Первая модель касается процесса фильтрации, например, жидкости в частично насыщенной пористой среде. Она описывается следующим уравнением

$$\frac{\partial b(u)}{\partial t} - \sum_{i=1}^N \frac{\partial^2 u(x, t)}{\partial x_i^2} = f(x, t, u). \quad (1)$$

Здесь t – время, x – пространственные координаты, неизвестная функция $u(x, t)$ имеет смысл давления в фильтрующейся жидкости, а заданная нелинейная функция $b(u)$ представляет собой зависимость плотности жидкости от давления в ней и имеет следующий вид

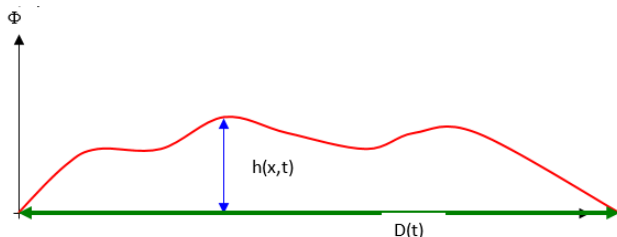


То есть, при $u > U$ функция $b(u)$ уже не изменяется и является константой. Область, где давление $u > U$ соответствует полному насыщению пористой среды, когда никаким повышением давления нельзя увеличить плотность находящейся в этой области жидкости. Пусть фильтрующаяся жидкость занимает объем Q . Обозначим область, где в момент времени t выполнено неравенство $u > U$ через $Q1(t)$ (это область полного насыщения), а оставшуюся область – через $Q2(t)$. Так как в области $Q1(t)$ функция $b(u)$ является константой, то ее производная равна нулю в этой области и первое слагаемое в уравнении (1) пропадает, а само уравнение становится эллиптическим в этой области. В то же время, в области $Q2(t)$ первое слагаемое сохраняется и само уравнение является параболическим. Сама граница раздела между областями $Q1(t)$ и $Q2(t)$ является одним из главных неизвестных задачи. К настоящему времени неизвестно даже, следует ли из описанной математической модели, что такая граница вообще существует.

Этот вопрос можно сформулировать несколько иначе. Предположим, что в начальный момент времени $t > 0$ области $Q1(0t)$ и $Q2(0)$ разделены четкой гладкой границей. Следует ли из описанной модели, что эти области, изменяясь с течением времени, будут и далее разделены четкой границей, или они перемешаются с течением времени? Ответ на этот вопрос остается пока открытой математической проблемой.

Задача со свободной границей для уравнения тонких пленок.

Другая модель описывает поведение жидкой тонкой пленки на некоторой поверхности. Она получена из системы уравнений Навье - Стокса асимптотическим предельным переходом по толщине пленки. Пусть в момент времени $t > 0$ пленка занимает на поверхности область $D(t)$ и пусть толщина пленки в этот момент времени в точке из $D(t)$ с координатами $x=(x1,x2)$ равна $h(x,t)$.



Функция $h(x,t)$ описывает поведение пленки с течением времени и удовлетворяет уравнению четвертого порядка

$$\frac{\partial h}{\partial t} + \nabla(h^n \nabla \Delta h) = 0, h|_{D(t)} = 0, \frac{\partial h}{\partial n} \Big|_{D(t)} = 1.$$

Отметим, что область $D(t)$ также относится к числу неизвестных задачи. Пусть в начальный момент времени $t=0$ пленка занимает область $D(0)$, ограниченную вполне определенной гладкой кривой. Вопрос: обеспечивает ли описанная модель такое решение $(D(t), h(x,t))$, что и при $t>0$ область $D(t)$ остается ограниченной определенной кривой, или ее границы «размываются»? Ответ на этот вопрос также остается открытой математической проблемой.

Выводы. Отметим, что в последние годы в отделе Уравнений математической физики ГУ «Институт прикладной математики и механики» разработаны некоторые подходы и методы для решения описанных задач. Эти подходы в общем идейном плане основаны на абстрактных вариантах теоремы о неявной функции. Для конкретного же применения этих абстрактных вариантов были разработаны подходы к изучению краевых задач для параболических и эллиптических уравнений с вырождением на границе области в классах гладких функций. Кроме того были разработаны необходимые методы изучения неклассических постановок краевых задач, включая краевые задачи с различного вида динамическими граничными условиями. На основании имеющихся подходов были рассмотрены достаточно широкие важные подклассы описанных задач. Все это позволяет надеяться на успешное исследование упомянутых проблем в полном объеме в будущем.

Мы не приводим списка литературы по упомянутым вопросам, так как он очень обширен и содержит множество аспектов рассматриваемых проблем, по каждому из которых существует своя собственная библиография.

УДК 622.14

**Создание алгоритма управления перемещением породы на отвале
Прокопенко Е.В., Масло С.В., Уманский В.В.**

Донецкий национальный технический университет, г. Донецк
кафедра прикладной математики
E-mail: prokopenko1515@rambler.ru

Аннотация

Прокопенко Е.В., Масло С.В., Уманский В.В. Создание алгоритма управления перемещением породы на отвале. В статье предлагается концепция по применению теории графов для создания хронолитологической модели породного отвала, которая позволит выявить интервалы локализации вредных химических компонент на отвале, и дать рекомендации по дальнейшему использованию этих компонент.

***Ключевые слова:** шахта, отвал, порода, граф, модель отвала*

На сегодняшний день в регионе накоплено 4 млрд. тонн отходов, которыми занято чуть менее 2 % территории области. Основные отрасли, образующие отходы - угольная промышленность (около 120 шахт и горнодобывающих предприятий). [1] Отходы угольных предприятий в основном сосредотачиваются на отвалах.

На территории Донбасса находится 1257 терриконов, которые занимают площадь 5526, 3 га. Большинство отвалов Донбасса являются горящими. В их недрах держится высокая температура, так как там, в избытке накоплен мышьяк, ртуть, цианиды, сера и другие вредные вещества и их соединения. Отвалы принимают породу от отдельной шахты, обогатительной фабрики или от группы угольных предприятий. [2]

В отвалах угольных шахт много запасов некоторых металлов, соизмеримых по объему с природными месторождениями полезных ископаемых, тем более что в настоящее время многие из минеральных ресурсов уже исчерпаны, что является одной из глобальных проблем. В ближайшем будущем все запасы благородных и цветных металлов, железа будут исчерпаны. Поэтому именно сейчас актуально рассматривать отходы горного производства, как альтернативный вариант пополнения природных ресурсов. Само по себе возведение террикона требует значительных усилий, ведь вначале надо построить собственно шахту, пройти километры горных выработок. Вся поднятая на-гора порода попадает в террикон, туда же идут и всевозможные отходы из шахты, это могут быть металлоконструкции, железобетон, дерево, кабели, и, небольшая часть угля (до 30 процентов). В модели отвала должна быть отражена локализация определенных типов (видов) пород и связанных с ними химические компоненты. Единственным достоверным источником такой информации являются результаты маркшейдерско-геологических съемок в горных выработках периодически с маркшейдерскими съемками отвалов. 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. Исходя из вышесказанного, можно сделать вывод, что маркшейдерский план это хронология деятельности "живого организма", и данную хронологию можно использовать для построения динамической модели формирования породного отвала

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. Горные работы ведутся по определенной системе и планированию, используя планограмму развития горных работ.

Исходя из выше сказанного, можно проследить динамику насыпки породы на отвал за определенной период времени, состоящую из следующих этапов:

Горные работы ведутся по определенной системе и планированию, используя планогамму развития горных работ.

Данная планогамма отображается на планах горных работ по каждому пласту, на котором ведутся работы.

Развитие горных работ осуществляется за определенный интервал времени (t) и в определенном месте проходческого забоя, т. е. осуществляется во времени и в пространстве.

Данная информация отображается на маркшейдерских планах горных работ в виде подвигания каждой выработки за определенное время.

Зная динамику насыпки пород, можно выявить опасные очаги выбросов вредных веществ на отвале, так как загрязнение атмосферы вредными веществами оказывает значительное воздействие на здоровье населения и экосистему области. [1]

Разработка данной модели предусматривает использование элементов теории графов, так как данная теория рассматривает постановку и решение задач управления организационными системами. [3] Одной из таких систем и является отсыпка породы на отвал.

Граф представляет собой систему, которая интуитивно может быть рассмотрена как множество кружков и множество соединяющих их линий. Кружки называются вершинами графа, линии со стрелками - дугами, без стрелок - ребрами.

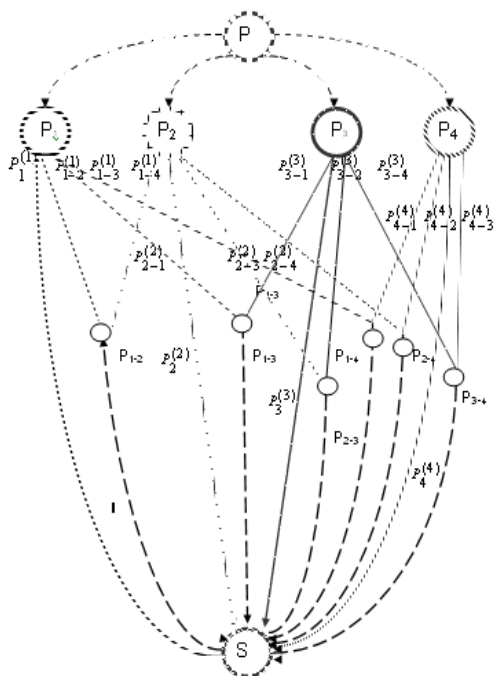


Рис.1. Модель формирования отвала в виде графа

Таблица 1-Характеристика элементов, входящих в граф

Обозначение вершины	Назначение вершины
P	Наименование пласта
P ₁	Вид выработок- квершлагги
P ₂	Вид выработок- уклоны
P ₃	Вид выработок- транспортные штреки
P ₄	Вид выработок- штреки за лавами
P ₁₋₂	Суммарный объем пород квершлаггов и уклонов
P ₁₋₃	Суммарный объем пород квершлаггов и транспортных штреков

P_{1-4}	Суммарный объем пород квершлагов и штреков за лавами
P_{2-3}	Суммарный объем пород уклонов и транспортных штреков
P_{2-4}	Суммарный объем пород уклонов и штреков за лавами
P_{3-4}	Суммарный объем пород транспортных штреков и штреков за лавами
Обозначение дуг	Назначение дуг
$P_1^{(1)}$	Отсыпка породы только из квершлагов
$P_2^{(2)}$	Отсыпка породы только из уклонов
$P_3^{(3)}$	Отсыпка породы только из транспортных штреков
$P_4^{(4)}$	Отсыпка породы только из штреков за лавами
Обозначение ребер	Назначение ребер
$P_{1-2}^{(1)}, P_{1-3}^{(1)}, P_{1-4}^{(1)}$	Одновременная работа квершлагов с уклонами, транспортными штреками и штреками за лавой
$P_{2-1}^{(2)}, P_{2-3}^{(2)}, P_{2-4}^{(2)}$	Одновременная работа уклонов с квершлагами, транспортными штреками и штреками за лавой
$P_{3-1}^{(3)}, P_{3-2}^{(3)}, P_{3-4}^{(3)}$	Одновременная работа транспортных штреков с квершлагами, уклонами и штреками за лавой
$P_{4-1}^{(4)}, P_{4-2}^{(4)}, P_{4-3}^{(4)}$	Одновременная работа штреков за лавами с квершлагами, уклонами и штреками

На основании графиков ввода-вывода проходческих забоев, порода попадает на отвал в основном из квершлагов, уклонов, транспортных штреков и штреков за лавами. Используя эти данные можно составить граф. На рисунке 1 представлен граф, реализующий доставку породы на отвал в пределах одного пласта.

Структура данного графа и все обозначения, представленные на схеме, могут быть сведены в таблицу 1, которая отражает полную характеристику данного графа.

Данная структура графа рассмотрена только в пределах одного пласта, т.е. по схеме можно рассмотреть динамику насыпки пород в зависимости от планограммы развития горных работ, которая отражена на маркшейдерских

планах. По каждому пласту составляется геологический разрез и осуществляется характеристика состава пород, входящих в пласт.

Для остальных пластов составляется такой же граф.

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 хронолитологическую модель насыпки породы, в результате которой могут быть выявлены места с накоплением вредных веществ [4]. По данной модели можно составить рекомендации по контролю за данными веществами, а также дать рекомендации по дальнейшему использованию этих веществ.

Список литературы

1. Земля тривоги нашої. За матеріалами доповіді про стан навколишнього природного середовища у Донецькій області у 2008-2009 роках / Під ред.С.В. Трет'якова, Г.Аверіна– Донецьк: Новий світ.-2009.-124с.
2. Оценка влияния породных отвалов шах."Горняк" ПО "Селидовуголь" на окружающую среду и перспективы их рекультивации/ Соловьева Е.А. (магистерская работа) - Руководитель: доцент кафедры "Полезные ископаемые и экологическая геология" Проскурня Юлия Анатольевна
3. Теория графов в управлении организационными системами/ Бурков В.Н., Заложнев А.Ю., Новиков Д.А.// М.: Синтег, 2001. -124с.
4. Прокопенко Е.В. Использование теории графов для создания хронолитологической модели формирования породного отвала/ Науковий вісник НГУ. - Національний гірничий університет, 2011. №5, т. С.28-31.

УДК 622.14

Применение линейного программирования для нахождения максимального компонента состава пород по пластам

Масло С.В., Прокопенко Е.В., Высоцкий С.А.

Донецкий национальный технический университет, г. Донецк

кафедра прикладной математики

E-mail: prokopenko1515@rambler.ru

Аннотация

Масло С.В., Прокопенко Е.В., Высоцкий С.А. Применение линейного программирования для нахождения максимального компонента состава пород по пластам. В статье предлагается концепция по применению теории графов для создания хронолитологической модели породного отвала, которая позволит выявить интервалы локализации вредных химических компонент на отвале, и дать рекомендации по дальнейшему использованию этих компонент

Ключевые слова: отвал, порода, целевая функция, матрица.

В отвалах угольных шахт много запасов некоторых металлов, соизмеримых по объему с природными месторождениями полезных ископаемых, 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 единственным достоверным источником информации за весь период эксплуатации шахты, и, следовательно, за весь период отсыпки породы на отвал.[1]

Крупные скопления отходов угольной промышленности оказывают негативное воздействие на окружающую среду. В частности происходит загрязнение атмосферного воздуха, особенно в случаях возгорания отвалов, а также почв и водных объектов.

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

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

где

$$\begin{aligned} V_1 &= (x_1^{(1)}, x_2^{(1)}, \dots, x_j^{(1)}, \dots, x_k^{(1)}) \\ V_2 &= (x_1^{(2)}, x_2^{(2)}, \dots, x_j^{(2)}, \dots, x_k^{(2)}) \\ &\dots \\ V_i &= (x_1^{(i)}, x_2^{(i)}, \dots, x_j^{(i)}, \dots, x_k^{(i)}) \\ &\dots \\ V_m &= (x_1^{(m)}, x_2^{(m)}, \dots, x_j^{(m)}, \dots, x_k^{(m)}) \end{aligned} \quad (2)$$

при ограничениях

$$\begin{aligned} g_1(x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(i)}, \dots, x_1^{(m)}) &\leq b_1 \\ g_2(x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(i)}, \dots, x_2^{(m)}) &\leq b_2 \\ &\dots \\ g_j(x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(i)}, \dots, x_j^{(m)}) &\leq b_j \\ &\dots \\ g_k(x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i)}, \dots, x_k^{(m)}) &\leq b_k \end{aligned} \quad (3)$$

Где $f(v_1, v_2, \dots, v_m)$ - 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,

$$g_1(x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(i)}, \dots, x_1^{(m)}) \leq b_1$$

...

$g_k(x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i)}, \dots, x_k^{(m)}) \leq b_k$ - 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.

$$\begin{array}{cccccc}
 V_1 & V_2 & \dots & V_i & \dots & V_m \\
 x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(i)} & \dots & x_1^{(m)} \\
 x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(i)} & \dots & x_2^{(m)} \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 x_j^{(1)} & x_j^{(2)} & \dots & x_j^{(i)} & \dots & x_j^{(m)} \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 x_k^{(1)} & x_k^{(2)} & \dots & x_k^{(i)} & \dots & x_k^{(m)}
 \end{array} \quad (4)$$

Для 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

$$V^{(l)} = \sum_{i=1}^m x_l^{(i)} \cdot V_i \quad (5)$$

Используя найденные объемы, можно составить 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.

В результате применения данного алгоритма было рассчитано максимальное значение химического компонента среди работающих пластов одной из шахт Донбасса. На рисунке 1 приведена таблица с характеристикой состава компонент и значением объемов пород по пластам.

Элемент	пласты			максим	миним
	к5	к3	к2		
титан	1125	3200	3500	3500	1125
свинец	9,8	17,8	16,9	17,8	9,8
олово	1,9	1,9	2	2	1,9
мышьяк	150	150	150	150	150
ванадий	60	56,2	97,5	97,5	56,2
марганец	286	143,8	267,5	286	143,8
галлий	7,5	6,9	9,8	9,8	6,9
никель	21,2	30	27,5	30	21,2
хром	17,4	13,1	25,6	25,6	13,1
кобальт	21,5	15,6	20,6	21,5	15,6
висмут	1	1	1	1	1
барий	350	350	312,5	350	312,5
бериллий	4,9	2,6	6,4	6,4	2,6
молибден	4,9	3,2	3,8	4,9	3,2
литий	20	21,9	95,6	95,6	20
индий	23,8	36,2	37,6	37,6	23,8
лантан	17,5	20	25	25	17,5
цирконий	130	268,8	180	268,8	130
медь	32,5	32,5	32,5	32,5	32,5
серебро	0,01	0,05	0,4	0,4	0,01
ниобий	5,5	4,8	6	6	4,8
цинк	40	146,2	40	146,2	40
скандий	10	20,6	26,5	26,5	10
фтор	117	124	150	150	117
макс	1125	3200	3500	3500	1125
мин	0,01	0,05	0,4	0,4	0,01

Рис.1 Нахождение максимального компонента по пластам

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 хронолитологическую модель насыпки породы, в результате которой могут быть выявлены места с накоплением вредных веществ. По данной модели можно составить рекомендации по контролю за данными веществами, а также дать рекомендации по дальнейшему использованию этих веществ.

Список литературы

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. Проскурня Ю.А., Бредихина Н.П. Породные отвалы Донецка и их влияние на экологическую ситуацию. -Донецк, 2002.
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 519.2

Усреднение случайных функционалов от решений стохастических уравнений

Грановский Я.И.

Отдел уравнений в частных производных ГУ ИПММ

E-mail: yarvodoley@mail.ru

Аннотация

Грановский Я.И. Усреднение случайных функционалов от решений стохастических уравнений. Рассматривается интегральный функционал от случайного стационарного поля с перемешиванием и от решения стохастического уравнения, зависящих от малого параметра. Вид функционала обусловлен вероятностным представлением решения задачи Коши и решения первой граничной задачи для линейного параболического уравнения второго порядка в недевергентной форме с неограниченными случайными быстрыми осцилляциями у слагаемого нулевого порядка производной. Основной результат – центральная предельная теорема о сходимости функционала.

Постановка задачи. Рассмотрим функционал, зависящий от малого параметра ε ,

$$F^\varepsilon(t, \theta, \omega) = \frac{1}{\sqrt{\varepsilon}} \int_0^t c\left(\frac{\xi^\varepsilon(s, \omega)}{\varepsilon}, \theta\right) ds.$$

Здесь $c(x, \theta)$ – стационарный процесс с перемешиванием, $\xi^\varepsilon(s, \omega)$ – решение одномерного стохастического уравнения Ито, заданные на разных вероятностных пространствах. Исследуем сходимость функционала в смысле распределений при стремлении малого параметра к нулю. Выбор такого функционала для исследования обусловлен тем, что через такой функционал выражаются решения задачи Коши и первой граничной задачи для одномерных линейных параболических уравнений в частных производных второго порядка недивергентного вида, коэффициенты которых зависят от малого параметра, и с сингулярно возмущённым случайным коэффициентом при слагаемом нулевого порядка производной неизвестной функции:

$$\frac{\partial u^\varepsilon(t, x)}{\partial t} = \frac{a^\varepsilon(x)}{2} \frac{\partial^2 u^\varepsilon(t, x)}{\partial x^2} + b^\varepsilon(x) \frac{\partial u^\varepsilon(t, x)}{\partial x} + \frac{1}{\sqrt{\varepsilon}} c\left(\frac{x}{\varepsilon}, \theta\right) u^\varepsilon(t, x).$$

Аналогичная проблема рассматривалась в работах [1, 2], но методы исследования существенно подстроены под уравнения, в которых оператор, содержащий частные производные, является самосопряжённым в соответствующем функциональном пространстве, т.е. $b^\varepsilon(x) = \frac{1}{2} \frac{\partial a^\varepsilon(x)}{\partial x}$.

Вспомогательные результаты и основные предположения.

В следующей лемме собраны известные оценки.

Лемма 1. 1. [3]. Для стационарного процесса *p.c.n.* при $h > 0$,

$$|\mathbb{E}c(x)c(x+h)| \leq 2\varphi^{\frac{1}{2}}(h)\mathbb{E}c^2(0).$$

2. [4]. Если для стационарного процесса *c.n.* при некоторой постоянной K , $|c(x)| \leq K$, то для $h > 0$,

$$|\mathbb{E}c(x)c(x+h)| \leq 6K^2\alpha^{\frac{1}{2}}(h).$$

3. [4]. Если для стационарного процесса *c.n.* при некотором $\delta > 0$, $\mathbb{E}|c(0)|^{2+\delta} < \infty$, то для $h > 0$,

$$|\mathbb{E}c(x)c(x+h)| \leq 12\alpha^{\frac{\delta}{2+\delta}}(h)[\mathbb{E}c^{2+\delta}(0)]^{\frac{2}{2+\delta}}.$$

Пусть процесс $\xi^\varepsilon(t)$ является решением стохастического уравнения

$$\xi^\varepsilon(t) = x + \int_0^t b^\varepsilon(\xi^\varepsilon(s))ds + \int_0^t \sigma^\varepsilon(\xi^\varepsilon(s))dB^\varepsilon(s), x \in \mathbb{R}, t \geq 0.$$

Относительно коэффициентов этого уравнения сделаем следующие предположения, полагая $a^\varepsilon(x) = (\sigma^\varepsilon(x))^2$:

(A1) Существует постоянная $\lambda \geq 1$ такая, что для $x \in \mathbb{R}$,

$$|b^\varepsilon(x)| + a^\varepsilon(x) \leq \lambda, \quad a^\varepsilon(x) \geq \frac{1}{\lambda}.$$

(A2) Существует функция $a(x)$ такая, что для любого $N < \infty$,

$$\lim_{\varepsilon \rightarrow 0} \sup_{x \in [-N, N]} |a^\varepsilon(x) - a(x)| = 0.$$

(A3) Существует функция $b(x)$ такая, что для любого $x \in \mathbb{R}$,

$$\lim_{\varepsilon \rightarrow 0} \int_0^x b^\varepsilon(y)dy = \int_0^x b(y)dy.$$

(A4) Функция $b(x)$ непрерывна, а функция $a(x)$ имеет ограниченную производную.

Основной результат.

Теорема 1. Пусть выполнено одно из условий леммы 1 и условия (A1) -- (A4). Тогда меры, порождённые на пространстве $\mathbb{C}[0, T]$ функционалом $F^\varepsilon(t, \theta, \omega)$, слабо сходятся к мере, порождённой функционалом

$$F(t, \theta, \omega) = \frac{c}{2} \int_{-\infty}^{+\infty} \frac{L(t, y)}{a(y)} dW(y).$$

Примеры. В рассматриваемых ниже примерах обыкновенных линейных дифференциальных уравнений первого порядка, записанных в интегральной форме, предполагаются выполненными условия теоремы.

1. Рассмотрим уравнение с мультипликативным неограниченным случайным возмущением

$$x^\varepsilon(t) = x + \frac{1}{\sqrt{\varepsilon}} \int_0^t x^\varepsilon(s)c\left(\frac{\xi^\varepsilon(s)}{\varepsilon}\right) ds.$$

В условиях теоремы $x^\varepsilon(t) \Rightarrow x$:

$$x(t) = x \exp \left\{ \frac{c}{2} \int_{-\infty}^{\infty} \frac{L(t, y)}{a(y)} dW(y) \right\}.$$

Поэтому можем рассматривать процесс $x(t)$ как решение стохастического интегро-дифференциального уравнения

$$x(t) = x + \frac{c}{2} \int_{-\infty}^{\infty} \frac{1}{a(y)} \int_0^t x(s) L(ds, y) dW(y).$$

2. Рассмотрим уравнение с аддитивным неограниченным случайным возмущением

$$x^\varepsilon(t) = x + \int_0^t \lambda^\varepsilon(s) x^\varepsilon(s) ds + \frac{1}{\sqrt{\varepsilon}} \int_0^t h^\varepsilon(\xi^\varepsilon(s)) \int_0^s c\left(\frac{\xi^\varepsilon(v)}{\varepsilon}\right) dv ds.$$

Предположим, что существует функция $\lambda(t)$ такая, что $\int_0^t \lambda^\varepsilon(s) ds \rightarrow \int_0^t \lambda(s) ds$ для любого $t \in [0, T]$ при $\varepsilon \rightarrow 0$, функции $h^\varepsilon(x)$ ограничены постоянной Λ и равномерно на компактах сходятся к непрерывной функции $h(x)$. Заметим, что $h^\varepsilon(\xi^\varepsilon(s)) \Rightarrow h(\xi(t))$. Следовательно, процесс $x(t)$ является решением уравнения

$$x(t) = x + \int_0^t \lambda(s) x(s) ds + \frac{c}{2} \int_0^t h(\xi(s)) \int_{-\infty}^{\infty} \frac{L(t, y)}{a(y)} dW(y) ds.$$

Список литературы

1. E. Pardoux, A. Piatnitski. Homogenization of a singular random one-dimensional PDE // Math. Sci. Appl., 24 (2005), 292 – 302.
2. B. Iftimie, E. Pardoux, A. Piatnitski. Homogenization of a singular random one-dimensional PDE // Annals de l'Institut H. Poincare, 44 (2008), No. 3, 512 – 543.
3. И. А. Ибрагимов, Ю. В. Линник. Независимые и стационарно связанные величины, М.: Наука, 1965.
4. Ю. А. Давыдов. О сходимости распределений, порождённых стационарными случайными процессами // Теория вероятностей и её применения, 13 (1968), No. 4, 730 – 737.

УДК 517.984.46

**Спектральная теория оператора бесселя на конечном интервале и полуоси
Ананьева А.Ю., Будыка В.С.**

Институт прикладной математики и механики, г. Донецк
отдел уравнений в частных производных
E-mail: budyka.vik@gmail.com

Аннотация

Ананьева А.Ю., Будыка В.С. Спектральная теория оператора Бесселя на конечном интервале и полуоси. Изучены минимальный и максимальный операторы, порожденные дифференциальным выражением Бесселя на конечном интервале и полуоси. Описаны области определения фридрихсова и крейновского расширений минимального оператора, а также все неотрицательные самосопряженные расширения минимального оператора

Ключевые слова: граничная тройка, функция Вейля, фридрихсово и крейновское расширения, квадратичная форма, спектральная функция.

Постановка проблемы. Рассмотрим однопараметрическое семейство дифференциальных выражений Бесселя

$$\tau_v = -\frac{d^2}{dx^2} + \frac{v^2 - 1/4}{x^2}, \quad v \in [0,1) \setminus \{1/2\}. \quad (1)$$

Спектральному анализу граничных задач для выражения (1) посвящено много работ (см. [1, с. 535] – [7] и литературу в них). Особо отметим работы Кальфа и Эверита [6], [3], в которых найдена явная форма m -коэффициента Вейля–Титчмарша выражения τ_v в $L^2(R_+)$.

В [6], [3], [2] описаны области определения фридрихсова расширения минимального оператора $A_{v,\infty}$ ассоциированного с выражением (1) в $L^2(R_+)$, а в [3] все самосопряженные расширения оператора $A_{v,\infty}$. Кроме того, в [2] описаны области определения соответствующих квадратичных форм. Однако, это описание апеллирует к области определения максимального оператора $A_{v,\infty}^*$, явное выражение для которой отсутствует в литературе.

В настоящей работе изучаются минимальный оператор Бесселя и его расширения на конечном интервале и полуоси.

Цель статьи – изучить спектральные свойства операторов, порожденных дифференциальным выражением Бесселя на конечном интервале и полуоси.

Оператор Бесселя на конечном интервале. Пусть $S_{\nu,b} := S_{\nu,b,\min}$ и $S_{\nu,b,\max}$ – минимальный и максимальный операторы Бесселя, соответственно, порожденные выражением (1) в $L^2(0,b)$; $b < \infty$ (см. [1, с. 535]). В дальнейшем $\text{span}\{F\}$ обозначает линейную оболочку множества F .

Теорема 1. Пусть $\nu \in (0,1)$. Справедливы следующие утверждения:

(i) оператор $S_{\nu,b}$ является неотрицательным и его индексы дефекта $n_{\pm}(S_{\nu,b}) = 2$;

(ii) область определения оператора $S_{\nu,b}$ задается соотношением

$$\text{dom}(S_{\nu,b}) = H_0^2[0,b] := \{f \in H^2[0,b] : f(0) = f(b) = f'(0) = f'(b) = 0\};$$

(iii) $S_{\nu,b,\max} = S_{\nu,b}^*$ и

$$\text{dom}(S_{\nu,b}^*) = \begin{cases} \tilde{H}_0^2[0,b] + \text{span}\{x^{1/2+\nu}, x^{1/2-\nu}\}, & \nu \neq 0, \\ \tilde{H}_0^2[0,b] + \text{span}\{x^{1/2}, x^{1/2} \ln(x)\}, & \nu = 0, \end{cases}$$

где $\tilde{H}_0^2[0,b] := \{f \in H^2[0,b] : f(0) = f'(0) = 0\}$.

В дальнейшем важную роль играет специальное расширение оператора Бесселя $A_{\nu,b}$ задаваемое дифференциальным выражением (1) в $L^2(0,b)$ на области

$$\text{dom}(A_{\nu,b}) = \{f \in S_{\nu,b}^* : f(0) = f'(0) = f(b) = 0\}, \quad \nu \in [0,1]. \quad (2)$$

Предложение 1. Пусть $\nu \in (0,1)$. Справедливы следующие утверждения:

(i) оператор $A_{\nu,b}$ имеет равные индексы дефекта $n_{\pm}(A_{\nu,b}) = 1$;

(ii) $\text{dom}(A_{\nu,b}) = \{f \in H^2[0,b] : f(0) = f'(0) = f(b) = 0\}$;

(iii) $\text{dom}(A_{\nu,b}^*) = \{f \in \text{dom}(S_{\nu,b}^*) : f(b) = 0\}$.

Оператор Бесселя на полуоси. Пусть $A_{\nu,\infty} := A_{\nu,\infty,\min}$ и $A_{\nu,\infty,\max}$ – минимальный и максимальный операторы Бесселя, порожденные выражением (1) в $L^2(\mathbb{R}_+)$.

Теорема 2. Пусть $\nu \in (0,1)$. Справедливы следующие утверждения:

(i) оператор $A_{\nu,\infty}$ имеет равные индексы дефекта $n_{\pm}(A_{\nu,\infty}) = 1$;

(ii) область определения оператора $A_{\nu,\infty}$ имеет вид:

$$\text{dom}(A_{\nu,\infty}) = H_0^2(\mathbb{R}_+) = \{f \in H^2(\mathbb{R}_+) : f(0) = f'(0) = 0\};$$

(iii) $A_{v,\infty,\max} = A_{v,\infty}^*$ и

$$\text{dom}(S_{v,b}^*) = \begin{cases} H_0^2(R_+) \dot{+} \text{span}\{x^{1/2+v}\xi(x), x^{1/2-v}\xi(x)\}, & v \neq 0, \\ H_0^2(R_+) \dot{+} \text{span}\{x^{1/2}\xi(x), x^{1/2}\ln(x)\xi(x)\}, & v = 0, \end{cases} \quad (3)$$

где $\xi(x) \in C_0^1(R_+)$ некоторая функция такая, что $\xi(x) = 1$ при $x \in [0,1]$.

Предложение 2. Пусть $v \in (0,1)$. Тогда:

(i) граничную тройку оператора $A_{v,\infty}^*$ можно выбрать в виде

$$H = C, \quad \Gamma_0^{v,\infty} f = [f, x^{1/2+v}]_0, \\ \Gamma_1^{v,\infty} f = \begin{cases} -(2v)^{-1} [f, x^{1/2+v}]_0, & v \in (0,1), \\ [f, x^{1/2} \ln(x)]_0, & v = 0; \end{cases} \quad (4)$$

(ii) соответствующая функция Вейля $M_{v,\infty}(\cdot)$ имеет вид:

$$M_{v,\infty}(z) = \begin{cases} e^{i(1-v)\pi} \frac{\Gamma(1-v)}{2v4^v \Gamma(1+v)} z^v, & v \in (0,1), \\ -\ln\left(\frac{\sqrt{z}}{2}\right) + \frac{i\pi}{2} - \gamma, & v = 0, \end{cases} \quad z \in C \setminus R,$$

где γ – константа Эйлера.

Это позволило получить явное описание фридрихсова и крейнковского расширений и соответствующих квадратичных форм.

Предложение 3. Пусть $v \in (0,1)$ и $\Pi_{v,\infty} = \{H, \Gamma_0^{v,\infty}, \Gamma_1^{v,\infty}\}$ – граничная тройка вида (4) для оператора $A_{v,\infty}^*$. Тогда:

(i) область определения фридрихсова расширения $A_{v,\infty,F}$ оператора $A_{v,\infty}$ имеет вид:

$$\text{dom}(A_{v,\infty,F}) = \ker(\Gamma_0^{v,\infty}) = \{f \in \text{dom}(A_{v,\infty}^*) : [f, x^{1/2+v}]_0 = 0\};$$

(ii) область определения крейнковского расширения $A_{v,\infty,K}$ оператора $A_{v,\infty}$ имеет вид:

$$\text{dom}(A_{v,\infty,K}) = \{f \in \text{dom}(A_{v,\infty}^*) : [f, x^{1/2-v}]_0 = 0\}.$$

Теорема 3. Пусть $v \in [0,1)$ и $A_{v,\infty,F}$ фридрихсово расширение оператора $A_{v,\infty}$. Пусть также функция $\xi(x) \in C_0^1(R_+)$ такова, что

$$\xi(x) = \begin{cases} 1, & x \in (0, 1/2), \\ 0, & x \geq 3/4. \end{cases}$$

Тогда: (i) при $\nu \in (0,1)$ квадратичная форма $a_{\nu,\infty}$ ассоциированная с фридриховым расширением $A_{\nu,\infty,F}$ имеет вид:

$$a_{\nu,\infty}[u] = \int_0^\infty |u'(x)|^2 dx + \left(\nu^2 - \frac{1}{4}\right) \int_0^\infty \frac{|u(x)|^2}{x^2} dx, \quad \text{dom}(a_{\nu,\infty}) = H_0^1(\mathbb{R}_+);$$

(ii) при $\nu = 0$ квадратичная форма $a_{0,\infty}$ ассоциированная с фридриховым расширением $A_{0,\infty,F}$ имеет вид:

$$a_{0,\infty}[u] = \int_0^\infty \left| u'(x) - \frac{u(x)}{2x} \right|^2 dx, \quad \text{dom}(a_{0,\infty}) \supset H_0^1(\mathbb{R}_+) \dot{+} \text{span}\{u_\alpha(x)\};$$

где $u_\alpha(x) := x^{1/2} |\ln(x)|^{-\alpha} \xi(x)$, $0 < \alpha \leq 1/2$. Функции $u_\alpha(x)$

являются линейно независимыми. При этом $\dim(\text{dom}(a_{0,\infty} / H_0^1(\mathbb{R}_+))) = \infty$.

Выводы. Теорема 3 усиливает и дополняет результаты работ [6] и [2]. Например, при $\nu \in (0,1)$ в [2] показано лишь, что $\text{dom}(A_{\nu,\infty,F})$ плотна в $H_0^1(\mathbb{R}_+)$.

Список литературы

1. Ахиезер Н.И., Глазман И.М. Теория линейных операторов в гильбертовом пространстве. – М.: Наука, 1966. 544 с.
2. Bruneau L., Dereziński J., Georgescu V. Homogeneous Schrödinger Operators on Half-Line. Ann. Henri Poincaré. 2011. V. 12. P. 547–590.
3. Everitt W.N., Kalf H. The Bessel differential equation and the Hankel transform. J. of Comput. and App. Math. 2007. V. 208. P. 3–19.
4. Fulton C. Titchmarsh–Weyl m -functions for second-order Sturm–Liouville problems with two singular endpoints. Math. Nachr. 2008. V. 281. 10. P. 1418–1475.
5. Fulton C., Langer H. Sturm–Liouville operators with singularities and generalized Nevanlinna functions. Comp. Anal. and Opera. Th. 2010. V. 4. 2. P. 179–243.
6. Kalf H. A Characterization of the Friedrichs Extension of Sturm–Liouville Operators. J. London Math. Soc. 1978. V. 17. 2. P. 511–521.
7. Kostenko A., Teschl G. On the singular Weyl–Titchmarsh function of perturbed spherical Schrödinger operators. J. Diff. Eqs. 2011. V. 250. P. 3701–3739.

УДК 512.3

**Об одном вопросе из элементарной теории чисел
Добровольский Н.Н.**

Тульский государственный педагогический университет им. Л. Н. Толстого, г.
Тула, Россия
E-mail: nikolai.dobrovolsky@gmail.com

Аннотация

Добровольский Н.Н. Об одном вопросе из элементарной теории чисел.
В работе строится контрпример к одному вопросу из элементарной теории чисел, который возник в общей алгебре в связи с исследованиями по продолжению частичных операций.

Ключевые слова: показатель числа 2 по модулю n , функция Эйлера $\varphi(n)$.

Введение. В связи с исследованием возможности продолжения частичных операций И. Б. Кожуховым был поставлен следующий вопрос.

Пусть m , n — различные нечётные числа. Верно ли, что существует такое натуральное i , что $m + 2^i$ взаимно просто с n ?

По своей постановке этот вопрос относится к элементарной теории чисел и связан с таким понятием, как показатель числа 2 по модулю n . Как хорошо известно (см. [1]), показатель δ , которому принадлежит 2 по модулю n , является делителем мультипликативной функции Эйлера $\varphi(n)$.

Нам удалось построить контрпример, который даёт отрицательный ответ на вопрос И. Б. Кожухова.

Построение контрпримера

Рассмотрим два числа $m = 9937\ 637$ и $n = 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 241 = 5\ 592\ 405$.

Заметим что $2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^4 - 1 = 5 \cdot 3$, $2^8 - 1 = 17 \cdot 15$,
 $2^{12} - 1 = 13 \cdot 315$, $2^{24} - 1 = 241 \cdot 3^2 \cdot 5 \cdot 7 \cdot 13 \cdot 17 = 16\ 777\ 215 = 3n$.

Таким образом, 2 принадлежит показателю 24 по модулю n .

Рассмотрим 24 числа $m + 2^i, i = 0, \dots, 23$:

$$9\ 937\ 637 + 2^0 = 3 \cdot 3\ 312\ 546$$

$$9\ 937\ 637 + 2^1 = 17 \cdot 584\ 567$$

$$9\ 937\ 637 + 2^2 = 7 \cdot 1\ 419\ 663$$

$$9\ 937\ 637 + 2^3 = 5 \cdot 1\ 987\ 529$$

$$9\ 937\ 637 + 2^4 = 9\ 937\ 637 + 2^0 + (2^2 - 1)(2^2 + 1) = 3(3\ 312\ 546 + 5)$$

$$9\,937\,637 + 2^5 = 9\,937\,637 + 2^2 + 2^2(2^3 - 1) = 7(1\,419\,663 + 4)$$

$$9\,937\,637 + 2^6 = 9\,937\,637 + 2^0 + (2^2 - 1)(2^4 + 2^2 + 1) = 3(3\,312\,546 + 21)$$

$$9\,937\,637 + 2^7 = 9\,937\,637 + 2^3 + 2^3(2^4 - 1) = 5(1\,987\,529 + 24)$$

$$9\,937\,637 + 2^8 = 9\,937\,637 + 2^0 + (2^2 - 1)(2^2 + 1)(2^4 + 1) = 3(3\,312\,546 + 85)$$

$$9\,937\,637 + 2^9 = 13 \cdot 764\,473$$

$$9\,937\,637 + 2^{10} = 9\,937\,637 + 2^0 + (2^2 - 1)(2^2(2^2 + 1)(2^4 + 1) + 1) = 3(3\,312\,546 + 341)$$

$$9\,937\,637 + 2^{11} = 9\,937\,637 + 2^2 + (2^3 - 1)2^2(2^6 + 2^3 + 1) = 7(1\,419\,663 + 292)$$

$$9\,937\,637 + 2^{12} = 9\,937\,637 + 2^0 + (2^2 - 1)(2^6 + 1)(2^4 + 2^2 + 1) = 3(3\,312\,546 + 1\,365)$$

$$9\,937\,637 + 2^{13} = 241 \cdot 41\,269$$

$$9\,937\,637 + 2^{14} = 9\,937\,637 + 2^0 + (2^2 - 1)(2^2(2^6 + 1)(2^4 + 2^2 + 1) + 1) = \\ = 3(3\,312\,546 + 5\,461)$$

$$9\,937\,637 + 2^{15} = 9\,937\,637 + 2^3 + 2^3(2^4 - 1)(2^8 + 2^4 + 1) = 5(1\,987\,529 + 6\,552)$$

$$9\,937\,637 + 2^{16} = 9\,937\,637 + 2^0 + (2^2 - 1)(2^2 + 1)(2^4 + 1)(2^8 + 1) = 3(3\,312\,546 + 21\,845)$$

$$9\,937\,637 + 2^{17} = 7(1\,419\,663 + 18\,724)$$

$$9\,937\,637 + 2^{18} = 3(3\,312\,546 + 87\,381)$$

$$9\,937\,637 + 2^{19} = 5(1\,987\,529 + 104\,856)$$

$$9\,937\,637 + 2^{20} = 3(3\,312\,546 + 349\,525)$$

$$9\,937\,637 + 2^{21} = 13(764\,473 + 161\,280)$$

$$9\,937\,637 + 2^{22} = 3(3\,312\,546 + 1\,398\,101)$$

$$9\,937\,637 + 2^{23} = 7(1\,419\,663 + 119\,837)$$

Выводы. Из этого примера следует отрицательный ответ на вопрос И. Б. Кожухова.

Построение данного контрпримера выполнялось с помощью компьютерного эксперимента с использованием пакетов Mathcad и Excel, которые были необходимы для подбора модуля n и нужного значения m .

Повидимому, аналогичный контрпример можно построить не только для 2, но и для любого числа a найдутся натуральные n и m взаимно простые с a такие, что для любого натурального i справедливо неравенство $(m + a^i, n) > 1$.

Работа выполнена по гранту РФФИ № 15-41-03263.

Список литературы

1. Виноградов И. М. Основы теории чисел. М.: Наука, 1981.

УДК 517.9

**Условия коэрцитивности интегранта Γ -предельного функционала для
последовательности интегральных функционалов, определенных на
различных весовых пространствах Соболева**

Рудакова О.А.

Донецкий национальный технический университет

E-mail: olga.a.rudakova@gmail.com

Аннотация

Рудакова О.А. *Условия коэрцитивности интегранта Γ -предельного функционала для последовательности интегральных функционалов, определенных на различных весовых пространствах Соболева. Рассматривается последовательность интегральных функционалов с интегрантами, удовлетворяющими определенным условиям роста и коэрцитивности, с вырождением по пространственным переменным. Кроме того, предполагается, что интегранты зависят не только от градиентов функций из областей определения функционалов, но и от самих функций. При некоторых условиях относительно пространств, на которых определены функционалы, и связанных с ними n -мерных областей, а также при условии Γ -сходимости последовательности функционалов, устанавливается коэрцитивность Γ -предельного интегранта.*

Общая постановка проблемы. Интерес к изучению Γ -сходимости функционалов и в особенности интегральных функционалов связан с тем, что во многих важных случаях она сопровождается сходимостью решений соответствующих вариационных задач для этих функционалов. Само понятие Γ -сходимости было введено Э. Де Джорджи [1] в 70-е годы прошлого столетия и на данный момент существует большое количество работ, посвященных различным вопросам, связанным с этим типом сходимости функционалов, в том числе интегральных функционалов с вырожденными интегрантами и переменными областями определения (см., например, [2]–[8], и библиографию в [2], [3]). Для различных ситуаций формулируется адекватное понятие Γ -сходимости, учитывающее специфику рассматриваемых задач и связанное с определенной сходимостью последовательностей из областей задания функционалов, а по своей структуре требующее выполнение двух свойств, характеризующих поведение функционалов на соответствующих последовательностях.

В настоящей статье рассматриваются весовые пространства Соболева, связанные с последовательностью n -мерных областей и интегральные функционалы, определенные на этих пространствах. Условие, характеризующее поведение интегрантов данных функционалов, содержит весовую функцию и

некоторую, вообще говоря, неограниченную последовательность функций. Особенностью предлагаемой ситуации является то, что интегранты рассматриваемых функционалов зависят не только от градиентов функций из соответствующих областей определения функционалов, но и от самих функций. Для такой последовательности функционалов устанавливаются условия, при которых имеет место коэрцитивность интегранта Γ -предельного функционала.

Исследования. Введем функциональные пространства, определения регулярной сильной связанности и Γ -сходимости функционалов. Пусть $n \in \mathbb{N}$, $n \geq 2$, Ω – ограниченная область в \mathbb{R}^n и $p \in (1, n)$. Пусть V – неотрицательная функция на Ω , причем $V > 0$ почти всюду в Ω , $v \in L^1_{loc}(\Omega)$, $(1/v)^{1/(p-1)} \in L^1_{loc}(\Omega)$. Через $L^p(v, \Omega)$ обозначим множество всех измеримых функций $u: \Omega \rightarrow \mathbb{R}$ таких, что $v|u|^p \in L^1(\Omega)$. $L^p(v, \Omega)$ есть банахово пространство с нормой $\|u\|_{L^p(v, \Omega)} = \left(\int_{\Omega} v|u|^p dx \right)^{1/p}$. Через $W^{1,p}(v, \Omega)$ обозначим множество всех функций $u \in L^p(v, \Omega)$ таких, что для любого $i \in \{1, \dots, n\}$ существует обобщенная производная $D_i u$, $D_i u \in L^p(v, \Omega)$. $W^{1,p}(v, \Omega)$ есть рефлексивное банахово пространство с нормой $\|u\|_{1,p,v} = \left(\int_{\Omega} v|u|^p dx + \sum_{i=1}^n \int_{\Omega} v|D_i u|^p dx \right)^{1/p}$. Через $\overset{\circ}{W}^{1,p}(v, \Omega)$ обозначим замыкание множества функций $C_0^\infty(\Omega)$ в $W^{1,p}(v, \Omega)$. $\overset{\circ}{W}^{1,p}(v, \Omega)$ есть рефлексивное банахово пространство с индуцированной нормой пространства $W^{1,p}(v, \Omega)$.

Далее, пусть $\{\Omega_s\}$ – последовательность областей в \mathbb{R}^n , содержащихся в Ω . Аналогично пространствам, введенным выше, определим функциональные пространства, соответствующие областям Ω_s . Пусть $s \in \mathbb{N}$. Через $L^p(v, \Omega_s)$ обозначим множество всех измеримых функций $u: \Omega_s \rightarrow \mathbb{R}$ таких, что $v|u|^p \in L^1(\Omega_s)$. $L^p(v, \Omega_s)$ есть банахово пространство с нормой $\|u\|_{L^p(v, \Omega_s)} = \left(\int_{\Omega_s} v|u|^p dx \right)^{1/p}$. Через $W^{1,p}(v, \Omega_s)$ обозначим множество всех функций $u \in L^p(v, \Omega_s)$ таких, что для любого $i \in \{1, \dots, n\}$ существует обобщенная производная $D_i u$, $D_i u \in L^p(v, \Omega_s)$. $W^{1,p}(v, \Omega_s)$ есть банахово пространство с

нормой $\|u\|_{1,p,\nu,s} = \left(\int_{\Omega_s} \nu |u|^p dx + \sum_{i=1}^n \int_{\Omega_s} \nu |D_i u|^p dx \right)^{1/p}$. Через $\tilde{C}_0^\infty(\Omega_s)$ обозначим множество всех сужений на Ω_s функций из $C_0^\infty(\Omega)$. Через $\tilde{W}_0^{1,p}(\nu, \Omega_s)$ обозначим замыкание множества $\tilde{C}_0^\infty(\Omega_s)$ в $W^{1,p}(\nu, \Omega_s)$. Отметим, что если $u \in \tilde{W}_0^{1,p}(\nu, \Omega)$ и $s \in N$, то $u|_{\Omega_s} \in \tilde{W}_0^{1,p}(\nu, \Omega_s)$.

Определение 1. Если $s \in N$, то q_s – отображение $\tilde{W}_0^{1,p}(\nu, \Omega)$ в $\tilde{W}_0^{1,p}(\nu, \Omega_s)$ такое, что для любой функции $u \in \tilde{W}_0^{1,p}(\nu, \Omega)$ $q_s u = u|_{\Omega_s}$.

Определение 2. Будем говорить, что последовательность пространств $\tilde{W}_0^{1,p}(\nu, \Omega_s)$ сильно связана с пространством $\tilde{W}_0^{1,p}(\nu, \Omega)$, если существует последовательность линейных непрерывных операторов $l_s : \tilde{W}_0^{1,p}(\nu, \Omega_s) \rightarrow \tilde{W}_0^{1,p}(\nu, \Omega)$ такая, что $\sup_{s \in N} \|l_s\| < +\infty$ и для любых $s \in N$ и $u \in \tilde{W}_0^{1,p}(\nu, \Omega_s)$ имеем $q_s(l_s u) = u$ почти всюду на Ω_s .

О понятии сильной связанности пространств следует сказать особо. Оно играет важную роль в вопросах усреднения задач Неймана в областях сложной структуры. Впервые условие сильной связанности было введено в работе Е.Я. Хруслова [9]. Сильная связанность последовательности пространств, используемых при исследовании сходимости решений краевых и вариационных задач в переменных (например, сильно перфорированных областях), позволяет перейти от последовательности решений, каждое из которых содержится в "своем", пространстве, к ограниченной последовательности в некотором едином пространстве. А это, в свою очередь, является первым шагом к выделению некоторого предельного элемента исходной последовательности и последующему доказательству того, что этот элемент есть решение соответствующей усредненной задачи. Понятие сильной связанности последовательности пространств $\tilde{W}_0^{1,p}(\nu, \Omega_s)$ с пространством $\tilde{W}_0^{1,p}(\nu, \Omega)$ достаточно подробно исследовано в работе [10]. В дальнейшем понадобится более тонкое понятие по сравнению с понятием сильной связанности данных пространств.

Для любых $y \in R^n$ и $t \in N$ положим $Q_t(y) = \{x \in R^n : |x_i - y_i| < 1/(2t), i=1, \dots, n\}$.

Определение 3. Будем говорить, что последовательность пространств $\tilde{W}_0^{1,p}(\nu, \Omega_s)$ регулярно сильно связана с пространством $W^{1,p}(\nu, \Omega)$, если существуют последовательность линейных непрерывных операторов $l_s : \tilde{W}_0^{1,p}(\nu, \Omega_s) \rightarrow W^{1,p}(\nu, \Omega)$ и постоянная $\bar{c} > 0$ такие, что: (i) $\sup_{s \in N} \|l_s\| < +\infty$; (ii) для любых $s \in N$ и $u \in \tilde{W}_0^{1,p}(\nu, \Omega_s)$ имеем $q_s(l_s u) = u$ почти всюду на Ω_s ; (iii) для любых $y \in \Omega$ и $t \in N$, удовлетворяющих условию $\overline{Q_t(y)} \subset \Omega$, существует $s_0 \in N$, такое что для любых $s \in N$, $s \geq s_0$, и $u \in \tilde{W}_0^{1,p}(\nu, \Omega_s)$ имеем

$$\int_{Q_{t+1}(y)} |\nabla(l_s u)|^p dx \leq \bar{c} \int_{Q_t(y) \cap \Omega_s} |\nabla u|^p dx$$

Введем теперь определение Γ -сходимости последовательности функционалов, определенных на соответствующих весовых пространствах Соболева.

Определение 4. Пусть для любого $s \in N$ I_s – функционал на $\tilde{W}_0^{1,p}(\nu, \Omega_s)$, I – функционал на $W^{1,p}(\nu, \Omega)$. Будем говорить, что последовательность $\{I_s\}$ Γ -сходится к функционалу I , если выполняются условия:

- 1) для любой функции $u \in W^{1,p}(\nu, \Omega)$ существует последовательность $w_s \in \tilde{W}_0^{1,p}(\nu, \Omega_s)$ такая, что $\lim_{s \rightarrow \infty} \|w_s - q_s u\|_{L^p(\nu, \Omega_s)} = 0$, $\lim_{s \rightarrow \infty} I_s(w_s) = I(u)$;
- 2) для любой функции $u \in W^{1,p}(\nu, \Omega)$ и любой последовательности $u_s \in \tilde{W}_0^{1,p}(\nu, \Omega_s)$ такой, что $\lim_{s \rightarrow \infty} \|u_s - q_s u\|_{L^p(\nu, \Omega_s)} = 0$, имеем $\liminf_{s \rightarrow \infty} I_s(u_s) \geq I(u)$.

Перейдем к изложению основного результата работы.

Пусть $b \in L^1(\Omega)$, $b \geq 0$ в Ω , и пусть $\{\psi_s\}$ – последовательность функций, удовлетворяющая следующим условиям: для любого $s \in N$ имеем

$\psi_s \in L^1(\Omega_s)$ и $\psi_s \geq 0$ в Ω_s ; последовательность норм $\|\psi_s\|_{L^1(\Omega_s)}$ ограничена; для любого открытого куба $Q \subset R^n$ имеем $\limsup_{s \rightarrow \infty} \int_{Q \cap \Omega_s} \psi_s dx \leq \int_{Q \cap \Omega} b dx$.

Пусть $c_1, c_2, c_3, c_4 > 0$ и $f_s : \Omega_s \times R \times R^n \rightarrow R, s \in N$ – последовательность функций такая, что: для любых $s \in N, \xi \in R$ и $\eta \in R^n$ функция $f_s(\cdot, \xi, \eta)$ измерима на Ω_s ; для любого $s \in N$ и почти все $x \in \Omega_s$ функция $f_s(x, \cdot, \cdot)$ непрерывна на $R \times R^n$; для любого $s \in N$, почти всех $x \in \Omega_s$ и любого $\xi \in R$ функция $f_s(x, \xi, \cdot)$ выпукла на R^n ; для любого $s \in N$, почти всех $x \in \Omega_s$, и любых $\xi \in R$ и $\eta \in R^n$ имеем

$$c_1 \nu(x)(|\xi| + |\eta|)^p - \psi_s(x) \leq f_s(x, \xi, \eta) \leq c_2 \nu(x)(|\xi| + |\eta|)^p + \psi_s(x);$$

для любого $s \in N$, почти всех $x \in \Omega_s$, и любых $\xi, \xi' \in R$ и $\eta \in R^n$ имеем

$$|f_s(x, \xi, \eta) - f_s(x, \xi', \eta)| \leq c_3 \nu(x)(1 + |\xi| + |\xi'| + |\eta|)^{p-1} |\xi - \xi'| + c_4 [\psi_s(x)]^{(p-1)/p} [\nu(x)]^{1/p} |\xi - \xi'|$$

Обозначим через Λ совокупность всех непустых открытых множеств λ в R^n таких, что $\lambda \subset \Omega$ и $meas \partial \lambda = 0$.

Определим функционал $J_{\lambda, s}$ следующим образом: если $\lambda \in \Lambda, s \in N$, то

$J_{\lambda, s}$ – функционал на $\tilde{W}_0^{1,p}(\nu, \Omega_s)$ такой, что для любой функции $u \in \tilde{W}_0^{1,p}(\nu, \Omega_s)$

$$J_{\lambda, s}(u) = \int_{\lambda \cap \Omega_s} f_s(x, u, \nabla u) dx, u \in \tilde{W}_0^{1,p}(\nu, \Omega_s).$$

Через F обозначим множество всех функций $f : \Omega \times R \times R^n \rightarrow R$, удовлетворяющих условиям: для любых $\xi \in R$ и $\eta \in R^n$ функция $f(\cdot, \xi, \eta)$ измерима на Ω ; для почти всех $x \in \Omega$ функция $f(x, \cdot, \cdot)$ непрерывна на $R \times R^n$; для почти всех $x \in \Omega$ и любого $\xi \in R$ функция $f(x, \xi, \cdot)$ выпукла на R^n ; для почти всех $x \in \Omega$ и любых $\xi \in R$ и $\eta \in R^n$ имеем

$$-b(x) \leq f(x, \xi, \eta) \leq c_2 \nu(x)(|\xi| + |\eta|)^p + b(x);$$

для почти всех $x \in \Omega$, любых $\xi, \xi' \in R$ и $\eta \in R^n$ имеем

$$|f(x, \xi, \eta) - f(x, \xi', \eta)| \leq c_3 \nu(x)(1 + |\xi| + |\xi'| + |\eta|)^{p-1} |\xi - \xi'| + c_4 [b(x)]^{(p-1)/p} [\nu(x)]^{1/p} |\xi - \xi'|$$

Введем следующее обозначение: если $f \in F$, $\lambda \in \Lambda$, то J_λ^f – функционал на $\overset{\circ}{W}^{1,p}(\nu, \Omega)$ такой, что для любой функции $u \in \overset{\circ}{W}^{1,p}(\nu, \Omega)$

$$J_\lambda^f(u) = \int_\lambda f(x, u, \nabla u) dx, \quad u \in \overset{\circ}{W}^{1,p}(\nu, \Omega).$$

Через F_0 обозначим множество всех функций $f \in F$, удовлетворяющих условию: существует постоянная $c' > 0$ и ограниченная измеримая положительная функция σ на Ω такие, что для почти всех $x \in \Omega$, любых $\xi \in R$ и $\eta \in R^n$

$$f(x, \xi, \eta) \geq c' \nu(x) (\sigma(x) |\xi| + |\eta|)^p - b(x).$$

Теорема 1. Пусть выполняются следующие условия:

(*) функция $\nu \in L^1(\Omega)$; (*₂) пространство $\overset{\circ}{W}^{1,p}(\nu, \Omega)$ компактно вложено в пространство $L^p(\nu, \Omega)$; (*₃) последовательность пространств $\overset{\circ}{W}_0^{1,p}(\nu, \Omega_s)$ регулярно сильно связана с пространством $\overset{\circ}{W}^{1,p}(\nu, \Omega)$; (*₄) существует ограниченная измеримая положительная функция σ на Ω такая, что для любого открытого куба $Q \subset \Omega$ $\liminf_{s \rightarrow \infty} \text{meas}(Q \cap \Omega_s) \geq \int_Q \sigma dx$; (*₅) для любого $\lambda \in \Lambda$ последовательность $\{J_{\lambda,s}^f\}$ Γ -сходится к функционалу J_λ^f , $f \in F$. Тогда функция $f \in F_0$.

Выводы. Представленный результат означает, что интегрант Γ -предельного функционала “наследует” свойства интегрантов f_s исходных функционалов, и, в частности, является коэрцитивным. Что касается некоторых требований данной теоремы, то условия на весовую функцию ν , при которых вложение $\overset{\circ}{W}^{1,p}(\nu, \Omega)$ в $L^p(\nu, \Omega)$ компактно, изложены в [6]. Регулярная сильная связанность пространства $\overset{\circ}{W}_0^{1,p}(\nu, \Omega_s)$ с пространством $\overset{\circ}{W}^{1,p}(\nu, \Omega)$ и условие (*₄) имеют место, например, в случае специальным образом перфорированных областей. Такие области также рассмотрены в [6]. Вопрос об условиях Γ -сходимости последовательности рассматриваемых функционалов остается пока открытым, заметим лишь, что Γ -сходимость интегральных функционалов в случае, когда интегранты зависят только от градиентов функций из областей определения функционалов, и эффективное представление для интегранта соответствующего Γ -предела имеют место, например, в случае определенной

периодичности интегрантов исходных функционалов по пространственной переменной или (и) периодичности структуры областей. В общем же случае особый интерес заключается в теоремах о Γ -компактности. Результаты такого типа представлены в работах [4], [5] и [8].

Список литературы

1. De Giorgi E. Su un tipo di convergenza variazionale [Текст] / E. De Giorgi, T. Franzoni // Atti Accad. Naz. Lincei. Rend. Cl. Sci. Fis. Mat. e Natur. – 1975. – 58, № 6. – P. 842-850.
2. Dal Maso G. An introduction to Γ -convergence [Текст] / G. Dal Maso. – Boston: Birkhäuser, 1993. – 337 p.
3. Braides A., Defranceschi A. Homogenization of multiple integrals // Oxford Lect. Ser. Math. and Appl. – New York: Clarendon Press, 1998. – 12. – 298 p.
4. Kovalevsky A.A., Rudakova O.A. Γ -convergence of integral functionals with degenerate integrands in periodically perforated domains // Труды ИПММ НАН Украины. – 2009. – 19. – С. 101-109.
5. Рудакова О.А. О Γ -сходимости интегральных функционалов, определенных на различных весовых пространствах Соболева [Текст] / О.А. Рудакова // Укр. мат. журн. – 2009. – 61, №1. – С. 99-115.
6. Ковалевский А.А. О связности подмножеств соболевских пространств и Γ -сходимости функционалов с переменной областью определения [Текст] / А.А. Ковалевский // Нелинейн. граничн. задачи. – 1989. – Вып. 1. – С. 48-54.
7. Ковалевский А.А. Условия Γ -сходимости и усреднение интегральных функционалов с различными областями определения [Текст] / А.А. Ковалевский // Докл. АН УССР. – 1991. – № 4. – С. 5-8.
8. Рудакова О.А. О коэрцитивности интегранта Γ -предельного функционала последовательности интегральных функционалов, определенных на различных весовых пространствах Соболева [Текст] / О.А. Рудакова // Труды ИПММ НАН Украины. – 2007. – 15. – С. 171-180.
9. Хруслов Е.Я. Асимптотическое поведение решений второй краевой задачи при измельчении границы области [Текст] / Е.Я. Хруслов // Мат. сб. – 1978. – 106, № 4. – С. 604-621.
10. Ковалевский А.А. О сильной связности весовых пространств Соболева и компактности последовательностей их элементов [Текст] / А.А. Ковалевский, О.А. Рудакова // Труды ИПММ НАН Украины. – 2006. – 12. – С. 85-99.

УДК 62-523.8

Разработка структуры генератора маркерных сеток для цифрового электронно-оптического преобразователя

Кузнецов П.А., Степанов О.А., Масло С.В., Клюковкин В.Р.

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

Кафедра электроэнергетики и электротехники
Донецкий национальный технический университет
Кафедра теоретической и прикладной математики
kuznetsovpvael@inbox.ru

Аннотация:

Кузнецов П.А., Степанов О.А., Масло С.В., Клюковкин В.Р. Разработка структуры генератора маркерных сеток для цифрового электронно-оптического преобразователя. Рассмотрена возможность совершенствования модели генератора маркерных сеток для улучшения его характеристик. Разобрана модель электрического принципиальной схемы

Общая постановка проблемы. Цифровой электронно-оптический преобразователь (ЭОП) предназначен для усиления слабых световых потоков и преобразования излучения в видеосигнал стандарта CCIR. Он является основным элементом приборов ночного видения. Генератор маркерных сеток формирует подвижные и неподвижные маркерные сетки, с помощью которых могут формироваться различные элементы дальномерных штрихов, штрихов боковых поправок, прицельных марок, служебная информация и пр.

Основное направление использования данной разработки – военное (прицелы ночного видения для стрелкового оружия), однако цифровой ЭОП также может использоваться в системах наблюдения, либо как модуль, встраиваемый в другие системы.

Массовое производство функциональных составляющих ЭОП и использование широко применяющихся комплектующих позволит существенно сократить стоимость конечного продукта.

Проведенные исследования. Электронно-оптический преобразователь – вакуумный фотоэлектронный прибор для преобразования невидимого глазом изображения объекта (в ИК, УФ и рентгеновских лучах) в видимое либо для усиления яркости видимого изображения. Он представляет собой стеклянную колбу с размещёнными в ней пластинами фотокатода и экрана, покрытого люминофором. В ЭОП изображение объекта проецируется с помощью объектива на фотокатод (при использовании рентгеновских лучей теневое изображение объекта проецируется на фотокатод непосредственно). Излучение от объекта

вызывает фотоэлектронную эмиссию с поверхности фотокатода, причём величина эмиссии с различных участков последнего изменяется в соответствии с распределением яркости спроецированного на него изображения [1].

В данной разработке изображение, получаемое с помощью ЭОП, преобразуется ПЗС-матрицей в аналоговый видеосигнал и поступает на микродисплей. ПЗС - матрица (прибор с зарядовой связью) - твердотельное светочувствительное электронное устройство, преобразующую энергию фотонов в электрический заряд.

Такие устройства, представляющие собой узел, в котором сопряжены ЭОП и матрица ПЗС, называются гибридно-модульные преобразователи (ГМП). Возможны 2 варианта построения ГМП (рис. 1 и 2):

экран ЭОП стыкуется с внешней матрицей ПЗС (ICCD);

матрица ПЗС располагается внутри ЭОП вместо его экрана (ЕВССД).

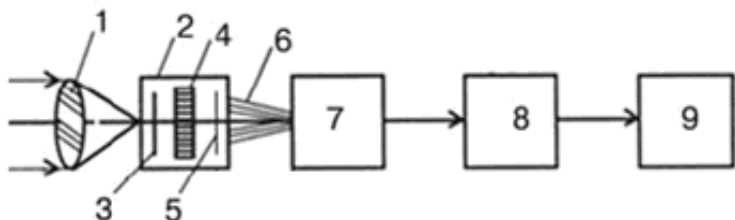


Рис. 1 – Схема системы на базе ГМП ICCD: 1 – объектив; 2 – ЭОП; 3 – фотокатод; 4 – микроканальная пластина; 5 – экран; 6 – оптика переноса; 7 – матрица ПЗС; 8 – видеоусилитель; 9 – монитор (дисплей)

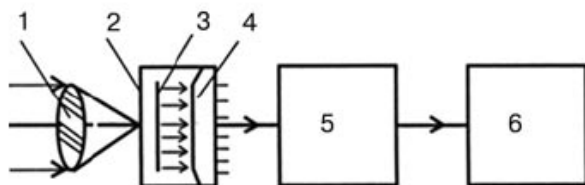


Рис.2 – Схема системы на базе ГМП ЕВССД: 1 – объектив; 2 – ЭОП; 3 – фотокатод; 4 – матрица ПЗС; 5 – видеоусилитель; 6 – монитор

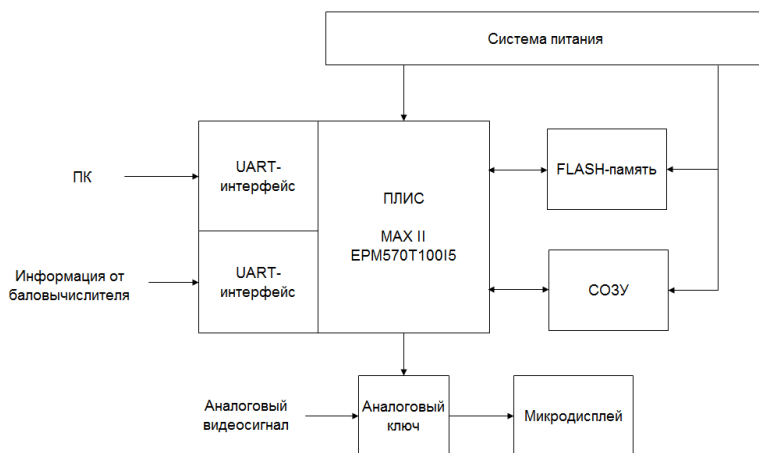


Рис. 3 – Структурная схема генератора маркерных сеток

Работа устройства начинается с передачи содержимого FLASH-памяти в СОЗУ (статическое оперативное запоминающее устройство). В памяти хранится неподвижная и подвижная сетка и символы индикации служебной информации (все цифры и некоторые буквы). После передачи содержимого в СОЗУ включается отображение всех составляющих маркерных сеток. Видеосигнал включается раньше, сразу после сигнала сброса, автоматически генерируемого внутри ПЛИС.

Для программирования FLASH-памяти к изделию через разъем подключается персональный компьютер. Через интерфейс UART в пакетном режиме информация о конфигурации сеток и символов записывается сначала в СОЗУ, а затем, автоматически, во FLASH. Информация на служебные поля индикатора поступает только от баловычислителя также через интерфейс UART.

ПЛИС, FLASH-память и СОЗУ входят в узел обработки информации, который осуществляет генерацию подвижной и неподвижной маркерных сеток синхронно с поступающим видеосигналом. Под управлением ПЛИС аналоговый ключ переключается на изображение сеток и служебной информации в заданные моменты времени телевизионного раstra.

Узел обработки информации. Аналоговый видеосигнал поступает с контроллера ПЗС-матрицы через разъём XP3. Синхронизация с поступающим видеосигналом осуществляется по четырем служебным сигналам FLD, HD, VD и CLK_V полукадровая, горизонтальная, вертикальная и тактовая синхронизация соответственно. Управление подвижной сеткой осуществляется от пятикнопочной клавиатуры (влево-вправо-вверх-вниз-зафиксировать).

Клавиатура подключается через разъем XP1, который одновременно является и технологическим разъёмом для программирования ПЛИС.

DD2 – микросхема FLASH-памяти. DD3 – микросхема СОЗУ. DA5 – кварцевый генератор. DD1 – микросхема ПЛИС. XP2 – разъем, через который подключается ПК для программирования FLASH-памяти. XP4 – разъем, к которому подключён баловычислитель. Через разъём XP5 устройство подключается к микродисплею.

Описание электрической принципиальной схемы

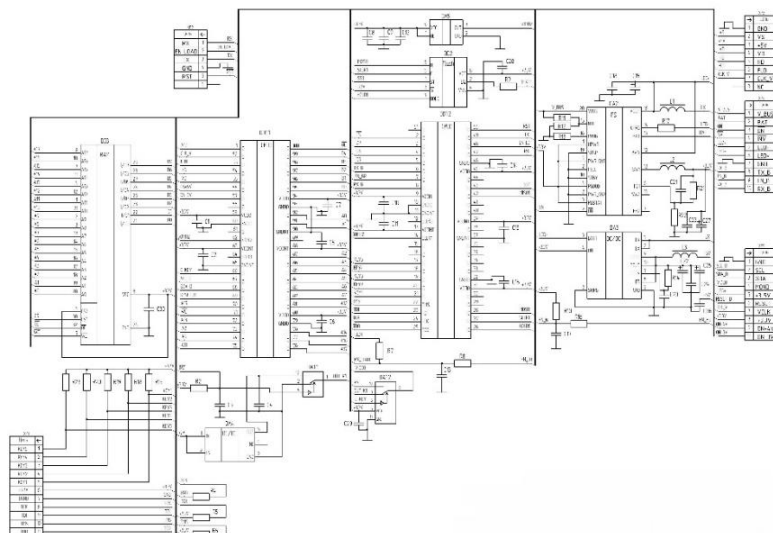


Рис. 4 Схема электрическая принципиальная

Узел сопряжения с микродисплеем. Узел сопряжения с микродисплеем представляет собой ключ DA1 и линейный стабилизатор DA4. Под управлением ПЛИС ключ переключается на изображение сеток и служебной информации в заданные моменты времени телевизионного раstra. Для согласования двух источников видеосигнала: с контроллера ПЗС-матрицы и генератора сеток применен линейный стабилизатор, который удерживает уровень яркости независимо от нагрузки. Источник же видеосигнала согласован с входным сопротивлением драйвера микродисплея. Драйвер микродисплея содержит также источник вторичного электропитания на +4 и -3 В, включением которых управляет ПЛИС. Согласно спецификации микродисплея питание подается на него последовательно: +3.3 В, программирование параметров дисплея, и затем +4 В, -3 В.

Узел питания. Система электропитания генератора маркерных сеток построена на микросхемах DA2 типа LTC3554 и DA3 типа MAX8815A. Микросхема LTC3554 содержит зарядное устройство для литий-ионного аккумулятора и два импульсных понижающих конвертера напряжения, из которых в данной конфигурации задействован один для выработки напряжения +3.3 В. Первичное постоянное напряжение величиной от 4.35 до 5.5 В от внешнего источника через контакт 1 разъема XP4 поступает на вход VBUS DA2 и используется для заряда внешнего аккумулятора номинальным напряжением 3.7 В, а также для выработки системного напряжения LED+ и выходного +3.3 В. Уровень выходного напряжения +3,3 В задается резистивным делителем R21-R22 так, чтобы на выводе FB1 микросхемы было напряжение обратной связи, равное ее внутреннему опорному напряжению (0.8+/-0.02 В). При отсутствии первичного напряжения для выработки системного и выходного напряжений используется энергия аккумулятора. Состояние заряда аккумулятора индицируется свечением внешнего светодиода, сигнал на который (LED-) подается с вывода CHRГ через токоограничивающий резистор R17 и контакт 6 разъема XP4 [4].

Микросхема DA3 типа MAX8815A представляет собой одноканальный конвертер постоянного напряжения повышающего типа и используется в данном устройстве для выработки напряжения +5 В. Микросхема включена по типовому варианту (силовой дроссель L1, силовой конденсатор C22+C24+C25), но для лучшего сглаживания пульсаций на выходе дополнена однозвенным LC-фильтром (L3, C26). Конвертер на DA3 получает питание от системного напряжения LED+, вырабатываемого микросхемой DA2.

Элементная база. Определяющими параметрами при выборе элементной базы послужили такие параметры, как быстродействие, миниатюризация, теплостойкость и вибростойкость (т.к. устройство мобильное и эксплуатируется на открытом воздухе).

Для улучшения блокировочных свойств конденсаторов их следует располагать как можно ближе к ПЛИС. Для этого они должны быть наименьшего размера, однако их установка не должна при этом сильно усложнять технологический процесс. Исходя из этих соображений, выбираем блокировочные конденсаторы фирмы Murata с типоразмером 0603.

Информация по компонентам представлена в виде краткого описания и основных технических характеристик.

SG-210SCB-L – однократно программируемый кварцевый генератор с диапазоном выходных частот от 2 до 60 МГц. Напряжение питания 3.3 В. Габаритные размеры: 2.5x2.0x0.8 мм. Диапазон рабочих температур от -40 до +125 °С.

Как отмечено выше микросхема LTC3554 содержит зарядное устройство для литий-ионного аккумулятора и два импульсных понижающих конвертера напряжения. Разработанная специально для USB приложений, она автоматически

ограничивает входной ток до 100 мА или 500 мА. Габаритные размеры: 3х3х0.75 мм. Диапазон рабочих температур от -40 до +85 °С.

Кроме этого, принципиальная схема содержит конденсаторы, резисторы, дроссели, микросхемы и разъёмы. Вся эта элементная база была подобрана таким образом, чтобы уменьшить габариты, массу изделия, обеспечить хорошее быстродействие, теплостойкость и вибростойкость (т.к. устройство мобильное и эксплуатируется на открытом воздухе).

Выводы. В данном разделе были представлены и описаны структурная и принципиальная схема генератора маркерных сеток. Принципиальная схема содержит конденсаторы, резисторы, дроссели, микросхемы и разъёмы. Вся эта элементная база была подобрана таким образом, чтобы уменьшить габариты, массу изделия, обеспечить хорошее быстродействие, теплостойкость и вибростойкость (т.к. устройство мобильное и эксплуатируется на открытом воздухе).

Список литературы

1. Яроцкий В.Г., Печаткин А.В., Гусев И.М. Проектирование и технология радиоэлектронных средств: Учебное пособие в 3-х частях – Рыбинск: РГАТА, 1997
2. Медведев А.В., Гринкевич А.В., Князева С.Н. Перспективные направления развития оптико-электронной техники и техники ночного видения. 2012
3. Стещенко В.Б. ПЛИС фирмы Altera: элементная база, система проектирования и языка описания аппаратуры. – М.: Издательский дом «Додэка-XXI», 2007
4. Разработка и оформление конструкторской документации радиоэлектронной аппаратуры: Справочник / Э.Т. Романычева, А.К. Иванова, А.С. Куликов и др.; Под ред. Э.Т. Романычевой. – М.: Радио и связь, 1989. – 448 с.

УДК 512.579

О конгруэнциях частичных универсальных алгебр Решетников А.В. (МИЭТ, Москва)

При рассмотрении различных алгебраических систем нередко приходится иметь дело с частичными операциями. Извлечение корня из действительного числа, деление элементов в каком-либо поле, вычитание натуральных чисел – всё это примеры частичных операций, которые не являются операциями в том смысле, в котором это понятие используется в алгебре. С другой стороны, классические разделы общей алгебры (теория групп, теория колец и т.д.) в основном посвящены изучению свойств полных (не частичных) операций. Поэтому для алгебраических систем с полными операциями

естественно представляют интерес вопросы обобщения тех или иных результатов на аналогичные системы, содержащие частичные операции.

Множества с введёнными на них частичными операциями – частичные универсальные алгебры – изучались различными авторами. Было получено немало интересных результатов, многие из которых представлены в монографиях [1] (Г. Гретцер) и [2] (Е. С. Ляпин и А. Е. Евсеев); последняя посвящена частичным бинарным группоидам, но в её библиографический список включены работы по самым разным вопросам теории частичных универсальных алгебр. Во вторых главах обеих монографий есть сведения о конгруэнциях: из теории универсальных алгебр было известно, что конгруэнции играют важную роль при изучении гомоморфизмов, и оказалось, что при изучении гомоморфизмов частичных алгебр их конгруэнции продолжают играть такую же важную роль¹.

Данный доклад посвящён изучению частичных универсальных алгебр, у которых каждое отношение эквивалентности является конгруэнцией. Обозначим класс таких алгебр через K . В общем случае, если частичная алгебра A принадлежит классу K , её строение не известно, но если A – полная алгебра, то её характеристику даёт теорема 1, приведённая в работе [3]. Нами показано, что на самом деле аналогичная характеристика справедлива в более общем случае – а именно, она имеет место для некоторых частичных алгебр специального вида, которые будут описаны в данном докладе.

Для получения основных результатов были активно использованы R_i -конгруэнции частичных n -арных группоидов, не смотря на то, что в формулировках теорем R_i -конгруэнции не участвуют. Понятие R_i -конгруэнции было введено автором в работе [4] как обобщение понятия односторонней конгруэнции бинарного группоида. В той же работе были описаны частичные n -арные группоиды, у которых для некоторого значения i все отношения эквивалентности являются R_i -конгруэнциями. В теории полугрупп

¹ Заметим, что понятие конгруэнции можно перенести с полных универсальных алгебр на частичные двумя неэквивалентными способами (использование второго способа приводит к сильным конгруэнциям, которые в данном докладе не рассматриваются). Гомоморфизм для частичных алгебр также определяется неоднозначно: у Г. Гретцера предложено три способа, а Е. С. Ляпин и А. Е. Евсеев в своей монографии в главе 2, §2 рассматривают и другие специальные способы, «отражающие ту или иную из черт единого понятия гомоморфизма теории полных действий».

односторонние конгруэнции играют важную роль при рассмотрении полигонов (автоматов). В случае полного n -арного группоида его отношение эквивалентности ρ является конгруэнцией в том и только том случае, если оно является R_i -конгруэнцией при всех i (следствие из предложений 1.1 и 1.2 работы [4]).

Пусть A – произвольное множество. Отображение $f : A' \rightarrow A$ называется *частичной n -арной операцией* на множестве A , если $A' \subseteq A^n$. Если $\Sigma = \{f_\alpha \mid \alpha \in I\}$ – какое-либо множество частичных операций, заданных на одном и том же множестве A , то A называется *частичной универсальной алгеброй с набором операций Σ* .

Теорема. Пусть A – частичная универсальная алгебра с набором частичных операций Σ , удовлетворяющая следующему условию: для любой частичной операции $f \in \Sigma$ (обозначим её арность через n) для любых $x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n \in A$ значение выражения $f(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n)$ определено по крайней мере для трёх различных элементов $y \in A$. Тогда все отношения эквивалентности на A являются её конгруэнциями в том и только том случае, если каждая её частичная операция f является либо константой (т.е. $|f(A, \dots, A)| \leq 1$), либо проекцией (т.е. $f(x_1, \dots, x_n) = x_i$ для некоторого i и всех $x_1, \dots, x_n \in A$).

Частный случай этой теоремы был рассмотрен в работе [5].

Список литературы

1. G. Grätzer. Universal algebra. Second Edition. Springer. 2008, 2nd ed. with updates. 1979, Second Edition. Springer Science + Business Media, LLC. 586 p.
2. Ляпин Е. С., Евсеев А. Е. Частичные алгебраические действия: монография. С.-Петербург, Росс. гос. пед. ун-т им. А. И. Герцена: Образование, 1991. 163 с. (английский перевод: Springer, Springer Science + Business Media, B.V., 1997, 237 p.)
3. Решетников А. В. Об односторонних конгруэнциях n -арного группоида. // ЭИС. 2016. №1(8). С. 89 – 95.
4. Решетников А. В. О конгруэнциях частичных n -арных группоидов. // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2011. Т. 11, вып. 3, ч. 2. С. 46 – 51.

5. Решетников А. В. О частичных n -арных группоидах, у которых каждое отношение эквивалентности является конгруэнцией. // Чебышевский сб. 2016. Т. 17, вып. 1. С. 232 – 239.

УДК 519.83

Ігрові методи в теорії статистичних рішень

Іє О. М.

Луганський державний університет імені Тараса Шевченка, м. Луганськ
 кафедра фундаментальної математики
 E-mail: olgaie@mail.ru

Анотація

Іє О.М. Ігрові методи в теорії статистичних рішень. Досліджено ігрові методи в теорії статистичних рішень, визначено істотні відмінності ігор з природою від антагоністичних ігор, детально описано основні критерії вибору рішень: критерій Байєса, Лапласа, Вальда, Севіджа, Гурвіца. Також розроблено алгоритм дій дослідника для визначення оптимальної системи в умовах невизначеності з вихідною інформацією про середовище і без неї.

Ключові слова: теорія ігор, виграш, стратегія, прийняття рішень, критерій, ймовірність, ризик.

Загальна постановка проблеми. Дослідження теоретичних і практичних проблем управління в даний час стає все більш актуальним внаслідок того, що відбувається перехід до інформаційного суспільства, а також спостерігається часте виникнення конфліктних ситуацій економічного характеру. Управлінські стратегії повинні бути спрямовані сьогодні на моментальне рішення несподівано виниклих проблем, так як конкуренція зростає з кожним днем. До того ж в умовах конфлікту інтересів та невизначеності на конкурентному ринку необхідно вміти виробити оптимальну поведінку, яка забезпечила б мінімальні втрати, підвищення економічної та технологічної ефективності управління, стабільність і добробут. На практиці це неможливо здійснити без науково обґрунтованих методів. Саме такі методи розроблені математичною теорією конфліктних ситуацій.

Сьогодні ведеться робота, спрямована на розширення сфери застосування теорії ігор. В даний час ця теорія є ефективним інструментом для аналізу та дослідження економічних, політичних, правових, військових, технічних та інших проблемних ситуацій. В останні роки спостерігається успішна реалізація методів теорії ігор в управлінській практиці. В якості прикладів застосування теорії ігор в управлінні можна назвати рішення з приводу проведення принципової цінової

політики, вступу на нові ринки, кооперації та створення спільних підприємств для здійснення проектів, розробки стратегій розвитку компаній, ведення переговорів. Ось чому тема роботи досить актуальна в даний час.

Мета роботи – дослідження особливостей застосування методів теорії ігор в теорії статистичних рішень.

У багатьох задачах, що приводяться до ігрових, невизначеність проявляється як результат дії тих чи інших "стихийних сил" (непізваної природи). Такі ігри називаються іграми з природою. При цьому термін "природа" може бути використаний як у традиційному сенсі, що означає довкілля, погодні умови в даному районі, так і умови ринку, що визначають попит на продукцію, обсяг перевезень, деяке поєднання виробничих факторів і т.д.

Дослідження. У теорії статистичних рішень невизначена для особи, приймаючої рішення (ОПР), ситуація у її "грі" з природою не має вираженого конфліктного забарвлення. Іншими словами, у цій теорії невідомі умови, обставини зовнішнього середовища залежать не від свідомої і активної протидії супротивника, а від об'єктивної дійсності. У грі з природою ОПР ніхто активно не заважає, але їй важче обґрунтувати свій вибір, ніж в антагоністичній грі, де противник також дотримується певних правил вибору оптимальної поведінки і його дії якимось чином можуть бути передбачені або прогнозовані.

Людина (гравець А) в іграх з природою намагається діяти обачно, використовуючи деяку оптимальну стратегію, що дозволяє отримати найбільший вигравш або найменший програвш. Гравець В (природа) діє абсолютно випадково, можливі стратегії визначаються як її стани.

Гра з природою описується матрицею H , як і кінцева антагоністична гра (зрозуміло, за умови, що розглядається кінцеве число "стратегій" природи, тобто реалізацій тих чи інших обставин). ОПР – гравець 1, максимізує свій вигравш. Тому можна говорити про виділення множини стратегій ОПР, оптимальних в сенсі Парето, тобто про викреслювання домінуючих стратегій у процесі спрощення гри. Але виділяти паретівську множину стратегій природи (гравця 2) не можна, тому що вона не керується ніякими принципами оптимальності – це перша істотна відмінність гри з природою від антагоністичної. Припустимо, що вигравш $H_1(i, j) > H_1(k, s)$. Це може відбутися не тільки за рахунок того, що ОПР вдало обрала стратегію x_1^i замість x_1^k . Просто виявилось, що реалізований стан природи x_2^j вигідніший для ОПР, ніж стан x_2^s . Іноді ж перевага одного стану природи перед іншим може виглядати для ОПР цілком очевидною. Тут під словом "перевага" розуміється, що особа, яка приймає рішення буде орієнтувати свою поведінку на якийсь цілком певний стан природи як найбільш ймовірне або як найбільш небажане, яке слід по можливості відбити своїми діями.

Другою відмінною рисою гри з природою є той факт, що стратегії природи за рідкісним винятком не можна змішувати, тобто слід вважати, що

природа оперує тільки чистими стратегіями. Змішування стратегій ОПР доцільно тільки тоді, коли гра з природою може багаторазово повторюватися, що має місце вельми нечасто.

Розглянемо детальніше критерії вибору рішень.

1. Критерій Байеса. Якщо невизначеність реалізації станів природи носить стохастичний характер, то для кожного j -го стану природи можна вказати ймовірність його реалізації η_j :

$$j = \overline{1, n}; \quad \sum_{j=1}^n \eta_j = 1. \quad (1)$$

Тоді для ОПР виявляється доцільним вибирати ту стратегію, яка дає максимальний зважений ймовірностями виграш по рядку матриці H :

$$H_1 = \max_i \sum_{j=1}^n \eta_j \cdot h_{ij}. \quad (2)$$

2. Критерій Лапласа. Якщо ймовірності в принципі існують, але невідомі, тоді можна припустити, що всі стани природи рівноймовірні за принципом недостатньої підстави Лапласа. В такому випадку:

$$H_1 = \max_i \frac{1}{n} \sum_{j=1}^n h_{ij}. \quad (3)$$

3. Якщо визначеність природи нестохастична, то для прийняття оптимального рішення умови дії ОПР істотно несприятливі. Завдання відшукування дійсно оптимального рішення в цьому випадку підміняється завданням відшукування не найбільш гіршого рішення. До цієї проблеми існує кілька підходів, які виражаються через критерії вибору рішення в умовах повної невизначеності. Основними з цієї групи критеріїв є критерій Вальда, Севіджа і Гурвіца.

Нижче ці критерії розглядаються детально.

а) критерій Вальда (критерій обережного спостерігача). Це максимальний критерій, згідно з яким передбачається, що гра з природою ведеться як з розумним і агресивним противником, тобто середовище знаходиться в самому несприятливому для ОПР стані. Оптимальною для ОПР вважається стратегія, яка дає гарантований результат, тобто нижнє значення гри α :

$$Wa = \max_i \min_j h_{ij}. \quad (4)$$

б) критерій Севіджа. Це мінімакський критерій, який застосовується не до значень елементів матриці H виграшів ОПР, а до елементів матриці R ризиків (по іншій термінології – матриці жалю):

$$Sa = \min_i \max_j r_{ij}, \quad \text{де } r_{ij} = \beta_j - h_{ij} = \max_j h_{ij} - h_{ij}. \quad (5)$$

Таким чином, ризик, нажаль виступає як плата за відсутність інформації. У сенсі ступеня песимізму цей критерій подібний з критерієм Вальда, але сам песимізм тут розуміється по-іншому;

в) *критерій Гурвіца*. Цей критерій рекомендує при прийнятті рішення не керуватися ні крайнім песимізмом, ні нестримним оптимізмом. Для реалізації цих рекомендацій призначається коефіцієнт $0 < L < 1$, який можна назвати коефіцієнтом оптимізму. Критерій Гурвіца має вид:

$$Hu = \max_i (L \max_j h_{ij} + (1-L) \min_j h_{ij}). \quad (6)$$

При $L=0$ цей критерій вироджується в критерій Вальда; при $L=1$ – у максімакний критерій абсолютного оптиміста. Коефіцієнт L в реальності призначають, виходячи з досить суб'єктивних міркувань або на основі експертних оцінок ситуації, що складається. Чим небезпечніше ситуація, чим менше схильність до ризику, тим ближче до 0 призначається L і навпаки.

Існує модифікація цього критерію, в якій він формується не на основі матриці виграшів H , а на основі матриці ризиків R .

При розробці технічних або будь-яких інших систем необхідні дані про умови застосування та експлуатації проєктованих систем, які можуть залежати від стану середовища, в якому вони будуть експлуатуватися. Однак часто дослідник стикається з ситуацією, коли інформація про середовище відсутня або можна зробити деякі припущення про його стан.

Умови роботи системи, як правило, сильно залежать від стану середовища, яке, в свою чергу, часто має ймовірнісний характер, що ускладнює процес визначення оптимальних систем. У таких ситуаціях доцільно використовувати байєсовський метод прийняття рішень про змінні, що не спостерігаються, заснований на знанні апіорного розподілу ймовірностей і на умовному розподілі інших змінних при заданому значенні змінних, що не спостерігаються.

Сформулюємо загалом завдання пошуку оптимальної системи в умовах невизначеності. Припустимо, проєктувальник має або може визначити:

$A = (A_1, A_2, \dots, A_i, \dots, A_n)$ – множину всіх можливих систем;

$B = (B_1, B_2, \dots, B_j, \dots, B_m)$ – множину всіх можливих станів середовища;

$C = (C_{11}, C_{12}, \dots, C_{ij}, \dots, C_{mm})$ – витрати при використанні всіх можливих систем для всіх можливих станів середовища, де C_{ij} – витрати при використанні системи A_i при стані середовища B_j . (див. табл. 1)

Варіант системи A_i	Стан середовища					
	B_1	B_2	...	B_j	...	B_m
A_1	C_{11}	C_{12}	...	C_{1j}	...	C_{1m}
A_2	C_{21}	C_{22}	...	C_{2j}	...	C_{2m}
...
A_i	C_{i1}	C_{i2}	...	C_{ij}	...	C_{im}
...
A_n	C_{n1}	C_{n2}	...	C_{nj}	...	C_{nm}

Потрібно спроектувати таку систему, яка забезпечує мінімальні витрати, тобто:

- обчислити очікувані витрати для кожної системи A_i ($i = 1, 2, \dots, n$), використовуючи апіорний розподіл;
- вибрати систему з найменшими витратами.

Можливі дві ситуації:

1) відомо апіорний розподіл ймовірності стану середовища $P(B = B_j)$. На початкових етапах проектування систем проектувальник може мати деяку інформацію про середовище чи інтуїтивно здогадатися на підставі проектування попередніх систем про ймовірнісні параметри середовища, тобто проектувальник має справу з апіорною інформацією.

Витрати обчислюються за формулою:

$$C(A_i; B_j) = \sum_{i=1}^m (C_{ij} \cdot P(B = B_j)), \quad (i = \overline{1, n}) \quad (7)$$

2) дослідник не має інформації про стан середовища, але він має або може отримати додаткову інформацію з того чи іншого непрямого параметру середовища, в якому передбачається працювати системі. При цьому бувають відомі:

B – випадкова змінна, що характеризує стан середовища;

X – випадкова змінна, пов'язана так чи інакше з змінною B ;

$P(B_j)$ – апіорний розподіл B_j ;

$Q(x | B = B_j)$ – умовний розподіл змінної B при $X = x$.

Для визначення оптимальної системи потрібно знайти умовний розподіл змінної B при $X = x$. Розподіл $H(B_j | X = x)$ називається апостеріорним розподілом змінної B , який отриманий на підставі відомої інформації про змінну X .

Для визначення апостеріорної ймовірності можна використовувати формулу Байеса, яка пов'язує апіорну і апостеріорну ймовірності:

$$H(B_j | X = x) = \frac{Q(x | B = B_j) \cdot P(B_j)}{\sum_{k=1}^m Q(x | B = B_k) \cdot P(B_k)}. \quad (8)$$

Витрати для кожної системи в цьому випадку будуть дорівнювати:

$$C(A_i; B_j) = \sum_{i=1}^m (C_{ij}(B_i | X = x)), \quad (i = \overline{1, n}) \quad (9)$$

Остаточно можна представити наступний алгоритм дії дослідника при наявності додаткової інформації.

А. Маючи таблицю витрат (табл. 1), апіорний розподіл $P(B_j), (j = \overline{1, m})$, визначити за формулою (7) очікувані витрати для кожної системи для різних станів середовища, вибрати систему, що має найменші витрати.

Б. Якщо апіорний розподіл невідомо, то по умовному розподілу змінної X для даного стану середовища $B = B_j$, тобто $Q(x | B = B_j)$, необхідно:

- а) визначити апостеріорний розподіл змінної B за формулою (8);
- б) отримати надійне значення випадкової змінної X , скажімо, x ;
- в) обчислити очікувані витрати для кожної системи, використовуючи апостеріорний розподіл $H(B_j | X = x)$ за формулою (9);
- г) вибрати систему, яка має мінімальні витрати.

Для визначення оптимальної системи в умовах невизначеності з вихідною інформацією про середовище і без неї доцільно використовувати ЕОМ, яка значно прискорює процес пошуку оптимальної системи.

Висновки. В роботі досліджено ігрові методи в теорії статистичних рішень, визначено істотні відмінності ігор з природою від антагоністичних ігор, детально описано основні критерії вибору рішень: критерій Байеса, Лапласа, Вальда, Севіджа, Гурвіца.

Також розроблено алгоритм дій дослідника для визначення оптимальної системи в умовах невизначеності з вихідною інформацією про середовище і без неї.

Література

1. Вентцель Е. С. Элементы теории игр / Е. С. Вентцель. – М.: Государственное издательство физико–математической литературы, 1961. – 482 с.
2. Воробьев Н. Н. Основы теории игр. Бескоалиционные игры / Н. Н. Воробьев. – М.: Наука. Главная редакция физико–математической литературы, 1984. – 496 с.
3. Дубров А. М. Моделирование рискованных ситуаций в экономике и бизнесе: Учеб. пособие / А. М. Дубров, Б. А. Лагоша, Е. Ю. Хрусталева; под ред. Б. А. Лагоши. – М.: Финансы и статистика, 2000. – 176 с.
4. Протасов И. Д. Теория игр и исследование операций: Учебное пособие / И. Д. Протасов. – М.: Гелиос АРВ, 2003. – 368 с.

5. Шикин Е. В. От игр к играм. Математическое введение. Изд. 2-е, исправл. / Е. В. Шикин. – М.: Едиториал УРСС, 2003. – 112 с.

УДК 517.984.4

Индексы дефекта матричного оператора Шредингера с дельта-взаимодействиями

Костенко А.С., Маламуд М.М., Натягайло Д.Д.

Институт прикладной математики и механики

Отдел уравнений в частных производных

E-mail: delthink@mail.ru

Аннотация

Костенко А. С., Маламуд М. М., Натягайло Д.Д. Индексы дефекта матричного оператора Шредингера с дельта-взаимодействиями. Рассмотрен матричный оператор Шредингера с дельта-взаимодействиями. Определена взаимосвязь между его индексами дефектов и индексами дефектов специального класса граничных операторов, ассоциированных с блочными матрицами Якоби. Приведено достаточное условие для случая промежуточных индексов дефекта.

Общая постановка проблемы. Рассматривается матричный оператор Шредингера с точечными взаимодействиями $H_{X,\Lambda}$, ассоциированный в пространстве вектор-функций $L^2(\mathbb{R}_+; \mathbb{C}^p)$ с формальным дифференциальным выражением

$$l_{X,\Lambda} := -\frac{d^2}{dx^2} \otimes I_p + \sum_{n=1}^{\infty} \Lambda_n \delta(x - x_n). \quad (1)$$

Здесь $\Lambda_k = \Lambda_k^* \in \mathbb{C}^{p \times p}$, $k \in \mathbb{N}$, $X = \{x_k\}_{k=0}^{\infty} \subset \mathbb{R}_+$ – строго возрастающая последовательность, $\lim_{k \rightarrow \infty} x_k = \infty$, а δ – функция Дирака.

Скалярные операторы Шредингера с дельта-взаимодействиями впервые возникли в квантовой механике как точно решаемые модели [8], и их спектральные свойства достаточно хорошо изучены (см., например, [9, 10]). Операторы Шредингера с матрично-значными потенциалами, задаваемые выражением (1), представляют из себя актуальную тему для исследований, и в качестве основного источника информации и дальнейших ссылок укажем [1, 2] (в частности, в этих монографиях изучается случай операторно-значных потенциалов). А также отметим недавние работы [3, 4, 5, 6, 7], содержащие, в частности, достаточно подробную библиографию.

Настоящая работа посвящена описанию взаимосвязи индексов дефекта оператора $H_{X,\Lambda}$, которые мы будем обозначать $n_{\pm}(H_{X,\Lambda})$, и оператора, ассоциированного в пространстве $\ell^2(\mathbf{N}; \mathbf{C}^p)$ с блочной якобиевой матрицей $B_{X,\Lambda}$ следующего вида:

$$B_{X,\Lambda} = \begin{pmatrix} \frac{1}{d_1 d_2} I_p + \frac{1}{d_1 + d_2} \Lambda_1 & \frac{-1}{r_1 r_2 d_2} I_p & O_p & \dots \\ \frac{-1}{r_1 r_2 d_2} I_p & \frac{1}{d_2 d_3} I_p + \frac{1}{d_2 + d_3} \Lambda_2 & \frac{-1}{r_2 r_3 d_3} I_p & \dots \\ O_p & \frac{-1}{r_2 r_3 d_3} I_p & \frac{1}{d_3 d_4} I_p + \frac{1}{d_3 + d_4} \Lambda_3 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}, \quad (2)$$

где O_p и I_p – нулевая и единичная матрицы в $\mathbf{C}^{p \times p}$ соответственно.

Исследования. Следующая теорема показывает связь индексов дефектов минимальных операторов, ассоциированных с выражениями вида (1) и (2).

Теорема 1. Пусть $B_{X,\Lambda}$ – минимальный оператор, задаваемый матрицей (2). Тогда $n_{\pm}(H_{X,\Lambda}) = n_{\pm}(B_{X,\Lambda}) \leq p$.

Применяя известные результаты об индексах дефекта блочных матриц Якоби, **Теорема 1** позволяет изучать индексы дефекта оператора $H_{X,\Lambda}$. Например, возможно получить следующее утверждения.

Предложение 1. Если $\sum_{k=1}^{\infty} d_k^2 = \infty$, то $H_{X,\Lambda}$ самосопряжен, т.е. $n_{\pm}(H_{X,\Lambda}) = 0$.

Предложение 2. [11] Пусть $\sum_{k=1}^{\infty} d_k^2 < \infty$, и $d_k d_{k+2} \geq d_{k+1}^2$, $k \in \mathbf{N}$. Если для всех $k \in \mathbf{N}$

$$\sum_{k=1}^{\infty} d_{k+1} \left\| \Lambda_k + (d_k^{-1} + d_{k+1}^{-1}) I_p \right\|_{\mathbf{C}^p} < \infty, \quad (3)$$

то $n_{\pm}(H_{X,\Lambda}) = p$.

После рассмотрения утверждений, дающих представление о минимальности и максимальности индексов дефектов оператора $H_{X,\Lambda}$ перейдем к случаю промежуточных индексов: $n_{\pm}(H_{X,\Lambda}) = p_1$, $0 < p_1 < p$.

Пусть $p = p_1 + p_2$ с некоторыми натуральными p_1 и p_2 . Представим каждую матрицу $\Lambda_k \in \mathbf{C}^{p \times p}$ в виде

$$\Lambda_k = \begin{pmatrix} \Lambda_k^{11} & \Lambda_k^{12} \\ \Lambda_k^{21} & \Lambda_k^{22} \end{pmatrix}, \quad \Lambda_k^{ij} \in \mathbf{C}^{p_i \times p_j}, \quad i, j \in \{1, 2\}.$$

Теорема 2. Пусть $\{d_k\}_{k=1}^\infty \in \ell^2(\mathbf{N})$ – невозрастающая последовательность и $d_k d_{k+2} \geq d_{k+1}^2$, $k \in \mathbf{N}$. Если матрицы $\Lambda_k = \Lambda_k^*$ удовлетворяют следующим условиям:

1. матрицы Λ_k^{11} удовлетворяют условию (3) с p_1 вместо p ;
2. $\|\Lambda_k^{12}\|_{\mathbf{C}^{p_1 \times p_2}} = \mathbf{O}(d_k)$, $k \rightarrow \infty$;
3. $\Lambda_k^{22} = \tilde{\Lambda}_k^{22} + \hat{\Lambda}_k^{22}$,

причем $\|\hat{\Lambda}_k^{22}\|_{\mathbf{C}^{p_1 \times p_2}} = \mathbf{O}(d_k)$, $k \rightarrow \infty$, а $\tilde{\Lambda}_k^{22}$ – диагональные матрицы такие, что либо

$$\sum_{k=1}^\infty |\alpha_{ij}| d_k^3 = \infty, \quad j \in \{1, \dots, p_2\},$$

либо найдется постоянная $M > 0$ такая, что для всех $k \in \mathbf{N}$ и $j \in \{1, \dots, p_2\}$

$$\frac{4}{d_{k+1}^2} + \frac{\alpha_{k,j}}{d_{k+1}} \leq M;$$

то $n_\pm(H_{X,\Lambda}) = p_1$.

Пример 1. Пусть $d_k = \frac{1}{k}$ и Λ_k имеют диагональный вид,

$\Lambda_k = \text{diag}(\alpha_{k,j})_{j=1}^p$, для всех $k \in \mathbf{N}$. Пусть также

$$\begin{aligned} \alpha_{k,j} &= -2k - 1 + \mathbf{O}(k^{-1}), \quad j \in \{1, \dots, p_1\} \text{ и} \\ \alpha_{k,j} &= -4k - 4 + \mathbf{O}(k^{-1}), \quad j \in \{p_1 + 1, \dots, p\}. \end{aligned}$$

Нетрудно видеть, что матрицы $\Lambda_k^{11} = \text{diag}(\alpha_{k,j})_{j=1}^{p_1}$, $k \in \mathbf{N}$, удовлетворяют (3). Кроме того, для всех $j \in \{p_1 + 1, \dots, p\}$ имеем

$$\frac{4}{d_{k+1}^2} + \frac{\alpha_{k,j}}{d_{k+1}} = 4(k+1)^2 - (k+1)(4k+4+O(k^{-1})) = O(1), \quad k \rightarrow \infty.$$

В силу **Теоремы 2** заключаем, что $n_{\pm}(H_{x,\lambda}) = p_1$.

Выводы. Взаимосвязь индексов дефектов гамильтонианов вида (1) и якобиевых матриц вида (2) справедлива не только для скалярного случая ($m=1$), но и для матричного ($m=p$). Случаи минимальности и максимальности дефектных чисел этих операторов, рассматриваемых в многомерных пространствах, хорошо известны. Мы выделили достаточные условия, позволяющие указать операторы Шредингера вида (1), которые имели бы промежуточный индекс дефекта ($0 < n < p$)

Список литературы

1. Rofe F. S. Rofe-Beketov, A. M. Kholkin Spectral analysis of differential operators. Interplay between spectral and oscillatory properties, World Scientific, Singapore, 2005
2. Горбачук В. И., Горбачук М. Л. Граничные задачи для операторно-дифференциальных уравнений. Киев, Наукова думка, 1984;
3. Eckhardt J., Gesztesy F., Nichols R., Teschl G. Weyl–Titchmarsh theory for Sturm–Liouville operators with distributional potentials, *Opuscula Math.*, 33, 2013, p. 467–563
4. Савчук А. М., Шкалик А. А. Операторы Штурма–Лиувилля с потенциалами-распределениями, *Труды ММО*, 64, 2003, с. 159–212
5. Мирзоев К. А., Сафонова Т. А. Сингулярные операторы Штурма–Лиувилля с негладкими потенциалами в пространстве вектор-функций, *Уфимск. матем. журн.*, 3:3, 2011, с. 105–119
6. Мирзоев К. А., Сафонова Т. А. Об индексе дефекта векторного оператора Штурма–Лиувилля, *Матем. Заметки*, 99:2, 2016, с. 262–277
7. Мирзоев К. А. Операторы Штурма–Лиувилля, *Труды ММО*, 75:2, 2014, с. 335–359
8. Albeverio S., Gesztesy F., Hoegh-Krohn R., Holden H. Solvable models in quantum mechanics. With an appendix by P. Exner. 2nd edn., Providence, Amer. Math. Soc., 2004
9. Костенко А. С., Маламуд М. М. Об одномерном операторе Шредингера с \mathcal{D} -взаимодействиями, *Функц. анализ и его прил.*, 44:2, 2010, с. 87–91
10. Kostenko A. S., Malamud M. M. 1–D Schrödinger operators with local point interactions on a discrete set, *J. Differential Equations*, 249, 2010
11. Костюченко А. Г., Мирзоев К. А. Признаки вполне неопределенности якобиевых матриц с матричными элементами, *Функц. анализ и его прил.*, 35, 2001, с. 265–269.

УДК 512.579

Некоторые вопросы теории полигонов над полугруппами

Кожухов И.Б.

НИУ МИЭТ, Москва, Россия

Полигон над полугруппой (см. [1]) определяется как множество, на котором действует полугруппа. А именно, пусть S – полугруппа, X – множество; если определено отображение $X \times S \rightarrow X$, (x, s) а xs такое, что $x(ss') = (xs)s'$ при всех $x \in X$, $s, s' \in S$, то X называется *правым полигоном* над полугруппой S (или *правым S -полигоном*). *Левый S -полигон* определяется двойственным образом: как множество Y вместе с отображением $S \times Y \rightarrow Y$, (s, y) а sy таким, что $(ss')y = s(s'y)$. Чтобы подчеркнуть, что X – правый S -полигон, а Y – левый, пишут $X = X_S$, $Y = {}_S Y$. Если множество Z является одновременно левым S -полигоном и правым T -полигоном для некоторых полугрупп S и T и выполняется равенство $(sz)t = s(zt)$, то Z называют *(S, T) -биполигоном* и пишут $Z = {}_S Z_T$. Левый полигон над полугруппой S можно рассматривать как правый полигон над полугруппой S^{op} , полученной из S инвертированием операции умножения: $S^{op} = (S, *)$, где $s*s' = s's$. Понятие биполигона может быть существенно обобщено: *мультиполигоном* над семейством полугрупп $\{S_i \mid i \in I\}$ будем называть множество X , на котором действуют полугруппы S_i (справа) и выполняются равенства $(xS_i)S_j = (xS_j)S_i$ при $x \in X$, $s_i \in S_i$, $s_j \in S_j$, где $i \neq j$.

Полигон X над полугруппой S является *алгебраической моделью автомата* (см. [2], [3]), при этом X – множество состояний автомата, а S – полугруппа входных сигналов. Алгебраическая теория автоматов – это составная часть кибернетики, основы этой теории были заложены В.М.Глушковым [4]. Биполигон и мультиполигон можно рассматривать как автомат, на который поступают сигналы из нескольких источников, и действия этих сигналов перестановочны друг с другом.

Исторически более ранним было рассмотрение действия групп на множествах: это группы подстановок, лежащие в основе теории Галуа, группы автоморфизмов алгебраических систем [5], группы гомеоморфизмов

топологических пространств, группы диффеоморфизмов гладких многообразий и т.д. Широко известна Эрлангенская программа Клейна, классифицирующая различные геометрии по их группам преобразований. В случае групп обычно предполагают, что их действие на множествах являются взаимно однозначными преобразованиями. Однако, в случае полугрупп мы отказываемся от требования биективности этих действий. Таким образом, всякая универсальная алгебра – полигон над полугруппой её эндоморфизмов, частично упорядоченное множество – полигон над полугруппой изотонных (т.е. сохраняющих порядок) преобразований, топологическое пространство X – полигон над полугруппой непрерывных отображений $X \rightarrow X$ и т.д.

Следует отметить, что полигоны над полугруппами являются унарными алгебрами, т.е. универсальными алгебрами с сигнатурой, состоящей из унарных операций (для полигона над полугруппой S унарными операциями сигнатуры являются отображения $x \text{ и } xs$ для каждого $s \in S$). Обратное тоже верно: унарную алгебру нетрудно сделать полигоном над естественно возникающей полугруппой. Унарные алгебры и их важный частный случай – унары (т.е. алгебры с одной унарной операцией) являются одной из основных тем Волгоградской алгебраической школы [6,7].

Среди основных направлений теории полигонов можно выделить следующие: 1) структурную теорию полигонов, 2) гомологическую теорию полугрупп и полигонов, 3) изучение полигонов специального вида.

К структурной теории относятся работы, где выясняется строение полигонов над теми или иными классами полугрупп, строение их подполигонов, конгруэнций и т.д. Для полугрупп сравнительно простого строения все полигоны над ними могут быть полностью описаны. Так, в работе [8] авторы описали полигоны над вполне простой $M(G, I, \Lambda, P)$ вполне 0-простой $M^0(G, I, \Lambda, P)$ (определения и обозначения см. в [9, гл.3]) полугруппами. В [10,11] были описаны конгруэнции полигонов над группами и над полугруппами правых или левых нулей. Назовём *частичным полигоном* множество X , для которого произведения xs определены, возможно, не для всех $x \in X$, $s \in S$, но равенство $(xs)s' = x(ss')$ выполнено, если определена какая-либо из частей этого равенства. Полигоны и частичные полигоны над полурешётками (т.е. коммутативными полугруппами идемпотентов) изучались в [12,13]. Там было замечено, что полигон над полурешёткой (даже частичный) является частично упорядоченным множеством относительно порядка $x \leq y \Leftrightarrow x = y$ или $x \in yS$. При этом были полностью описаны полигоны над конечной цепью. В работе [14] были полностью описаны биполигоны и мультиполигоны над

семейством полугрупп, каждая из которых является полугруппой правых или левых нулей.

Гомологической теории полугрупп посвящена большая часть монографии [1]. Суть этой теории в исследовании полугруппы по свойствам категории правых полигонов над ней. Начало гомологической теории полугрупп было положено Л.А.Скорняковым [15]. Чаще эту теорию называют омологической теорией моноидов (моноид – это полугруппа с единицей), так как в целом ряде её мест требуется наличие в полугруппе единицы, а на полигоны накладывается требование унитарности – т.е. чтобы $x \cdot 1 = x$ для любого элемента x полигона. Гомологическая теория полугрупп развивалась под большим влиянием гомологической теории колец, беря от неё многие понятия – такие, как тензорное произведение, инъективность, проективность и т.д. В [16] было доказано, что всякий полигон имеет инъективную оболочку – наименьшее инъективное расширение, и эта оболочка единственна с точностью до изоморфизма. В [17] были получены условия, при которых всякий полигон над данным моноидом имеет проективное накрытие – минимальный проективный полигон, имеющий сюръективный гомоморфизм на данный полигон. Описание инъективных полигонов над полугруппами левых нулей и построение инъективных оболочек произвольных полигонов над этими полугруппами было осуществлено в [18]. В [19] над полугруппами левых нулей были описаны проективные полигоны и были построены проективные накрытия. Кроме того, в [19] получены исчерпывающие результаты об инъективности и проективности для полигонов над полугруппами правых нулей.

Подпрямо неразложимые полигоны, т.е. полигоны, неразложимые в нетривиальное подпрямое произведение, исследовались в [20]. Интерес к таким полигонам объясняется известной теоремой Биркгофа, утверждающей, что всякая универсальная алгебра является подпрямым произведением подпрямо неразложимых алгебр. Таким образом, подпрямо неразложимые алгебры являются “строительным материалом”, из которого строятся все алгебры. В [20] было замечено, что подпрямо неразложимый полигон имеет не более двух нулей, и были охарактеризованы подпрямо неразложимые полигоны с двумя нулями. Таким образом, осталось невыясненным строение подпрямо неразложимых полигонов без нуля и с одним нулём. Такие полигоны были охарактеризованы в [21] “с точностью до ядра полигона”. Ядром называется наименьший нетривиальный подполигон (ядро существует не у всякого полигона, но у подпрямо неразложимого существует обязательно). Итак, зная строение ядра, можно написать необходимые и достаточные условия подпрямой неразложимости данного полигона. А для полигонов над прямоугольными связками, т.е. полугруппами, являющимися прямым произведением полугруппы левых и полугруппы правых нулей, в [21] получена исчерпывающая характеристика подпрямо неразложимых полигонов.

Интересными вопросами теории полугрупп и полигонов являются следующие:

(*) над какими полугруппами S все S -полигоны аппроксимируются конечными и

(**) над какими полугруппами S все S -полигоны аппроксимируются конечными, состоящими каждый не более, чем из l элементов?

Начало исследованию полугрупп со свойствами (*), (**) было положено в [22], а в [23] следования были продолжены и существенно усилены. Было доказано, что полугруппа S с условием (**) равномерно локально конечна, т.е. существует функция $f(k)$ такая, что k -порождённые подполугруппы имеют не более $f(k)$ элементов. На самом деле равномерная локальная конечность полугруппы S имеет место уже в случае, когда полигон S_S аппроксимируется конечными ограниченными в совокупности порядков. Если же S – абелева группа, то вопрос о том, выполняются ли (*) или (**), решается до конца. В [24] доказано, что условие (**) равносильно ограниченности группы (т.е. $a^m = 1$ для некоторого натурального m и всех $a \in S$). При этом порядки подпрямо неразложимых S -полигонов, аппроксимирующих произвольный S -полигон, равны p^α или $p^\beta + 1$, где p – простое число и $p^\alpha, p^\beta + 1 \leq n$. Необходимые и достаточные условия для (*) выглядят сложнее. Отметим лишь, что группа S имеет конечный ранг.

Конгруэнции универсальной алгебры – одно из центральных алгебраических понятий. Хорошо известно, что конгруэнции – это в точности ядра гомоморфизмов, поэтому знание всех конгруэнций алгебры означает знание всех её и гомоморфных образов. Множество классов конгруэнции с определёнными на нём естественным образом операциями называется фактор-алгеброй данной алгебры. Для полигона X над полугруппой S фактор-полигон X / ρ – это множество ρ -классов $x\rho$ со следующим действием на нём полугруппы S : $(x\rho)s = (xs)\rho$ ($s \in S$). Известная конструкция приведённого автомата означает переход от данного автомата (полигона) к некоторому фактор-полигону. Конгруэнции универсальной алгебры A образуют решётку, в которой частичный порядок – это теоретико-множественное включение. Решётка называется дистрибутивной, если $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$ для любых x, y, z , и модулярной, если это равенство выполняется при $x \leq z$ (здесь $x \wedge y = \inf\{x, y\}$, $x \vee y = \sup\{x, y\}$ – решёточные операции). Несмотря на

сходство теории модулей над кольцами и теории полигонов над полугруппами условия модулярности решётки конгруэнций весьма далеки друг от друга. В то время как решётка конгруэнций модуля – т.е. решётка подмодулей – всегда модулярна, модулярность решётки конгруэнций полигона – редкое явление.

Пусть X – полигон над полугруппой S . Для каждого $a \in S$ мы имеем отображение $\varphi_a : X \rightarrow X$, $x \mapsto xa$. Очевидно, отображение $\Phi : a \mapsto \varphi_a$ ($a \in S$) является гомоморфизмом полугруппы S в полугруппу $T(X)$ всех преобразований множества X . Обозначим через $\ker \Phi$ ядро гомоморфизма Φ . Тогда фактор-полугруппу $\bar{S} = S / \ker \Phi$ можно рассматривать как подполугруппу полугруппы $T(X)$. Будем говорить, что S действует на X эффективно, если Φ – вложение; иными словами, если $\varphi_a \neq \varphi_b$, при $a \neq b$. Легко видеть, что конгруэнции у полигонов X_S и $X_{\bar{S}}$ одни и те же.

В работе [25] исследовались условия модулярности и дистрибутивности решётки конгруэнций полигонов. Оказывается, полигон с модулярной решёткой конгруэнций, рассматриваемый как граф, имеет не более трёх компонент связности, полигон с дистрибутивной решёткой конгруэнций – не более двух. В [26] были описаны полигоны над полугруппой левых / правых нулей, у которых решётка конгруэнций модулярна, или дистрибутивна, или является цепью. Элементов в этих полигонах не более 9, полугруппа S , если она действует эффективно, содержит не более 27 элементов, а решётка конгруэнций – не более 160 элементов.

В заключение рассмотрим специальный вид полигонов – диагональные полигоны. Правым диагональным полигоном над полугруппой S называется полигон $(S \times S)_S$, левым – полигон ${}_S(S \times S)$, диагональным биполигоном – биполигон ${}_S(S \times S)_S$. Центральным вопросом здесь является наличие или отсутствие конечной системы образующих этого (би)полигона, а также системы образующих из одного элемента (в первом случае (би)полигон называется конечно порождённым, во втором – циклическим). Конечно, над конечной полугруппой S все диагональные (би)полигоны конечно порождены, а циклическими могут быть лишь в случае $|S| = 1$. Однако, в бесконечном случае возможны любые варианты. Так, в [27] доказано, что для бесконечного множества X полугруппа $T(X)$ всех преобразований, полугруппа $P(X)$ частичных преобразований и полугруппа $B(X)$ бинарных отношений на множестве X имеют циклические диагональные полигоны. Ряд результатов о циклическости и конечной порождённости диагональных (би)полигонов получен в [28]. В [29]

многие из этих результатов были усилены; например, было доказано, что у бесконечной полугруппы, удовлетворяющей нетривиальному тождеству, правый и левый диагональные полигоны не могут быть циклическими. Ряд результатов о цикличности и конечной порождённости диагональных полигонов над полугруппами изотонных преобразований частично упорядоченных множеств был получен в [30].

Список литературы

1. Kilp M., Knauer U., Mikhalev A.V. Monoids, acts and categories. N.Y. – Berlin, W. de Gruyter, 2000.
2. Плоткин Б.И., Гринглас Л.Я., Гварамия А.А. Элементы алгебраической теории автоматов. М., Высш. школа, 1994.
3. Лаллеман Ж. Полугруппы и комбинаторные приложения. М., Мир, 1985.
4. Глушков В.М. Абстрактная теория автоматов. Успехи мат. наук, 1961, 16 : 5 (101), 3-62.
5. Плоткин Б.И. Группы автоморфизмов алгебраических систем. М., Наука, 1966.
6. Карташова А.В. О решётках конгруэнций и топологий унарных алгебр Чебыш. сб., 2011, 12 : 2, 27-33.
7. Карташов В.К. О некоторых результатах и нерешённых задачах теории унарных алгебр. Чебыш. сб., 2011, 12 : 2, 18-26.
8. Avdeyev A.Yu., Kozhukhov I.B. Acts over completely 0-simple semigroups. Acta Cybernetica, 2000, 14, № 4, 523-531.
9. Клиффорд А., Престон Г. Алгебраическая теория полугрупп: М., Мир, 1972 г. , 2 т.
10. Халиуллина А.,Р. Конгруэнции полигонов над группами. Известия Саратов. ун-та. Новая серия. Серия Математика. Механика. Информатика, 2013, 13, вып. 4, ч. 2, 133-137.
11. Халиуллина А.Р. Конгруэнции полигонов над полугруппами правых нулей. Чебыш. сборник, 2013, 13, вып. 3, 142-146.
12. Максимовский М.Ю. О полигонах над полурешётками. Фунд. и прикл. матем., 2008, 14 : 7, 151-156.
13. Апраксина Т.В., Максимовский М.Ю. Полигоны и частичные полигоны над полурешётками. Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика, 2012, 12:1, 3-7.
14. Максимовский М.Ю. О биполигонах и мультиполигонах над полугруппами. Матем. заметки, 2010, 87 : 6, 855-866.
15. Скорняков Л.А. О гомологической классификации моноидов. Сиб. матем. ж., 1969, 10 : 5, 1139-1143.
16. Berthiaume P. The injective envelope of S-sets. Canad. Math. Bull., 1967, 10, 261-273.

17. Isbell J.R. Perfect monoids. *Semigroup Forum*, 1971, 2, 95-118.
18. Moghaddasi Gh. On injective and subdirectly irreducible S -acts over left zero semigroups. *Turk J. Math.*, 2012, 36, 359-365.
19. Кожухов И.Б., Халиуллина А.Р. Инъективность и проективность полигонов над сингулярными полугруппами. *Электронные информационные системы*, 2014, № 2(2), 45-56.
20. Ройз Е.Н. О подпрямо неразложимых монарах. Межвуз. научн. сборник “Упорядоченные множества и решёнки. Саратов, 1974, вып. 2, с. 80-84.
21. Кожухов И.Б., Халиуллина А.Р. Характеризация подпрямо неразложимых полигонов. *Прик. дискр. матем.*, 2015, № 1, 5-16.
22. Кожухов И.Б. Полугруппы, над которыми все полигоны резидуально конечны. *Фунд. и прикл. матем.*, 1998, 4 : 4, 1335-1344.
23. Кожухов И.Б., Халиуллина А.Р. Полугруппы с финитно аппроксимируемыми полигонами. *Матем. заметки СВФУ*, 2014, 21, № 3(83), 60--67.
24. Кожухов И.Б., Царёв А.В. Абелевы группы с финитно аппроксимируемыми полигонами. *Фунд. и прикл. матем. (в печати)*.
25. Птахов Д.О., Степанова А.А. Решётки конгруэнций полигонов. *Дальневост. матем. ж.*, 2013, 13, № 1, 107-115.
26. Халиуллина А.Р. Условия модулярности решётки конгруэнций полигона над полугруппой правых или левых нулей. *Дальневост. матем. ж.*, 2015, 15, № 1, 102-120.
27. Gallagher P., Ruškuc N. Finite generation of diagonal acts of some infinite semigroups of transformations and relations. *Bull. Austral. Math. Soc.*, 2005, 72, 139-146.
28. Gallagher P. On the finite and non-finite generation of diagonal acts. *Comm. Algebra*, 2006, 34, 3123-3137.
29. Apraksina T.V., Barkov I.V., Kozhukhov I.B. Diagonal ranks of semigroups. *Semigroup Forum*, 2015, 90, 2, 386-400.
30. Апраксина Т.В. Диагональные полигоны над полугруппами преобразований. *Чебыш. сб.*, 2011, 12, № 1, 10-16.

УДК 514.112.4

Построение шестиугольников, состоящих из любого числа n любых равных треугольников, начиная с некоторого; и построение восьмиугольников, состоящих из любых n равных, не равнобедренных треугольников.

Свентковский В.А.

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

Введение. Работа посвящена доказательству неизвестного факта комбинаторной геометрии, а именно: что из любого числа n , начиная с некоторого n_0 , из n произвольных одинаковых треугольников, можно сложить выпуклый шестиугольник. На основе этого факта доказано, что любое натуральное число, начиная с некоторого, можно выразить разностью между квадратом натурального числа и суммой квадратов трех натуральных чисел.

Методом построения этих шестиугольников выбран метод отрезания треугольников – уголков, подобных исходному треугольнику, от некоторого параллелограмма, со сторонами, кратными двум сторонам исходного треугольника. Доказывается также, что построенный шестиугольник можно деформировать в выпуклый восьмиугольник, составленный из того же числа треугольников, если треугольник неравнобедренный.

Методом построения этих восьмиугольников выбран метод такой деформации шестиугольника в восьмиугольник, при котором две известные части шестиугольника меняют свое расположение, а остальные части не меняются. В статье доказано также, что любое натуральное число, большее 670, можно представить в форме $n = 26N - a^2 - b^2$ с натуральными числами N, a, b , где $0 < a, b < 13$.

1. Построение шестиугольников, состоящих из любого числа n любых равных треугольников, начиная с некоторого

Теорема 1. Из любого числа $n \geq 671$ одинаковых треугольников можно сложить выпуклый шестиугольник.

Доказательство. Пусть исходный произвольный треугольник ABC имеет острыми углами углы A и C (см. рис.1). Достроим его до параллелограмма $ABCD$. AC растянем в M раз по горизонтали, AB растянем в N раз вдоль AB , так, что,

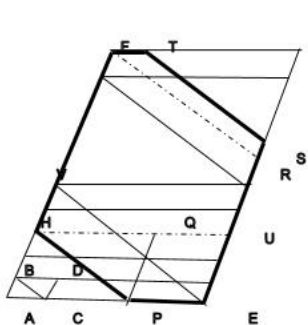


Рис. 1

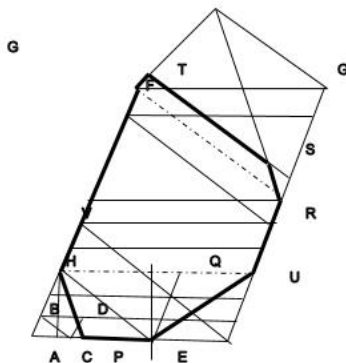


Рис. 2

M, N - натуральные числа. Достроим угол $\angle FAE$ до параллелограмма $AFGE$ составленного из MN параллелограммов $ABDC$, и, значит, из $2MN$ треугольников ABC . AC растянем в a раз по горизонтали, AB растянем в a раз вдоль AB , так, что $AH = a \cdot AB$ $AP = a \cdot AC$, a - натуральное число. Достроим угол $\angle HAP$ до параллелограмма $AHQP$, составленного из $a \cdot a$ параллелограммов $ABDC$, и, значит, из $2a^2$ треугольников ABC . Поэтому треугольник AHP состоит из a^2 треугольников ABC . Аналогично, если $GT = b \cdot AC, GS = b \cdot AB$, то получаем, что треугольник TGS состоит из b^2 треугольников ABC . Тогда шестиугольник $PHFTSE$ состоит из $n = 2MN - a^2 - b^2$ треугольников. Из рисунка следует, что если $0 < a, b < M, N$, $a + b < N$. то выпуклый шестиугольник, состоящий из n треугольников ABC строится отрезанием двух треугольников от параллелограмма. Шестиугольник, состоящий из $n = 2MN - a^2 - b^2$ треугольников ABC можно задать вектором $\overset{1}{F} = (M, N, a, b)$. Построение такого шестиугольника просто повторяет указанный ранее алгоритм. Осталось только для каждого вида n

подобрать вектор $\overset{1}{F} = (M, N, a, b)$, удовлетворяющий условию $0 < a, b < M, N$, $a + b < N$. Разобьем множество всех натуральных чисел на 26 подмножеств, дающие одинаковые остатки при делении на 26. И для каждого подмножества натуральных чисел, начиная с некоторого номера, укажем алгоритм построения шестиугольника, то есть вектор $\overset{1}{F} = (M, N, a, b)$ и проверку неравенств: $0 < a, b < M, N$, $a + b < N$, $a, b < M$. Потребуем, чтобы

$M = 13, N \geq 26, 0 < a, b < 13$, поэтому требуемые неравенства выполняются:
 $a + b < 13 + 13 = 26 \leq N, 0 < a, b < M, N$. Значит, все неравенства верны. И
проверять их не надо.

Строим алгоритм.

I. При нечетном n : $M = 13$

$$1. n = 26T - 1 = 26K - 53 = 2 \cdot 13 \cdot (T + 2) - 7^2 - 2^2, T + 2 \geq 26.$$

$$T \geq 24, N = T + 2, \overset{1}{F} = (M, N, a, b) = (13, T + 2, 7, 2)$$

$$n_{\min} = 26 \cdot 24 - 1 = 624$$

$$2. n = 26T - 3 = 26(T + 1) - 29 = 2 \cdot 13 \cdot (T + 1) - 5^2 - 2^2$$

$$T + 1 \geq 26, T \geq 25, N = T + 1$$

$$\overset{1}{F} = (M, N, a, b) = (13, T + 1, 5, 2), n_{\min} = 26 \cdot 25 - 3 = 650 - 3 = 647$$

$$3. n = 26T - 5 = 2 \cdot 13 \cdot T - 2^2 - 1^2, T \geq 26, N = T$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 2, 1), n_{\min} = 26 \cdot 26 - 5 = 676 - 5 = 671$$

$$4. n = 26T - 7 = 26(T + 3) - 85 = 2 \cdot 13 \cdot (T + 3) - 9^2 - 2^2$$

$$T + 3 \geq 26,$$

$$T \geq 23, \overset{1}{F} = (M, N, a, b) = (13, T + 3, 9, 2), N = T + 3$$

$$\overset{1}{F} = (M, N, a, b) = (13, T + 3, 9, 2)$$

$$n_{\min} = 26 \cdot 23 - 7 = 598 - 7 = 591$$

$$5. n = 26T - 9 = 26(T + 2) - 61 = 2 \cdot 13 \cdot (T + 2) - 6^2 - 5^2$$

$$N = T + 2 \geq 26,$$

$$T \geq 24, \overset{1}{F} = (M, N, a, b) = (13, T + 2, 6, 5)$$

$$n_{\min} = 26 \cdot 24 - 9 = 624 - 9 = 615$$

$$6. n = 26T - 11 = 26K - 37 = 2 \cdot 13 \cdot (T + 1) - 6^2 - 1^2, N = T + 1 \geq 26$$

$$T \geq 25, \overset{1}{F} = (M, N, a, b) = (13, T + 1, 6, 1)$$

$$n_{\min} = 26 \cdot 25 - 11 = 650 - 11 = 639$$

$$7. n = 26T - 13 = 2 \cdot 13 \cdot T - 3^2 - 2^2. \quad N = T \geq 26.$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 3, 2) \quad n_{\min} = 26 \cdot 26 - 13 = 676 - 13 = 663$$

$$8. n = 26T - 15 = 26(T + 1) - 41 = 2 \cdot 13 \cdot (T + 1) - 5^2 - 4^2$$

$$N = T + 1 \geq 26, \quad T \geq 25 \quad \overset{1}{F} = (M, N, a, b) = (13, T + 1, 5, 4)$$

$$n_{\min} = 26 \cdot 25 - 15 = 650 - 15 = 635$$

$$9. n = 26T - 17 = 2 \cdot 13 \cdot T - 4^2 - 1^2. \quad N = T + 1 \geq 26, \quad T \geq 25$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 4, 1) \quad n_{\min} = 26 \cdot 25 - 17 = 650 - 17 = 643$$

$$10. n = 26T - 19 = 26(T + 1) - 45 = 2 \cdot 13 \cdot (T + 1) - 6^2 - 3^2$$

$$N = T + 1 \geq 26, \quad T \geq 25 \quad \overset{1}{F} = (M, N, a, b) = (13, T + 1, 6, 3)$$

$$n_{\min} = 26 \cdot 25 - 19 = 650 - 19 = 631$$

$$11. n = 26T - 21 = 26(T + 2) - 73 = 2 \cdot 13 \cdot (T + 2) - 8^2 - 3^2$$

$$N = T + 2 \geq 26, \quad T \geq 24 \quad \overset{1}{F} = (M, N, a, b) = (13, T + 2, 8, 3)$$

$$n_{\min} = 26 \cdot 24 - 21 = 624 - 21 = 603$$

$$12. n = 26T - 23 = 26(T + 3) - 101 = 2 \cdot 13 \cdot (T + 3) - 10^2 - 1^2$$

$$N = T + 3 \geq 26, \quad T \geq 23 \quad \overset{1}{F} = (M, N, a, b) = (13, T + 3, 10, 1)$$

$$n_{\min} = 26 \cdot 23 - 23 = 598 - 23 = 575$$

$$13. n = 26T - 25 = 2 \cdot 13 \cdot T - 4^2 - 3^2. \quad N = T \geq 26,$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 4, 3)$$

$$n_{\min} = 26 \cdot 26 - 25 = 676 - 25 = 651 \quad \max(n_{\min}) = 671$$

II. При четном n

$$1. n = 26T - 2 = 2 \cdot 13 \cdot T - 1^2 - 1^2. \quad N = T \geq 26.$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 1, 1)$$

$$n_{\min} = 26 \cdot 26 - 2 = 676 - 2 = 674$$

$$2. n = 26T - 4 = 26(T + 3) - 82 = 2 \cdot 13 \cdot (T + 3) - 9^2 - 1^2$$

$$N = T + 3 \geq 26, T \geq 23, \overset{1}{F} = (M, N, a, b) = (13, T + 3, 9, 1)$$

$$n_{\min} = 26 \cdot 23 - 4 = 598 - 4 = 594$$

$$3. n = 26T - 6 = 2 \cdot 13 \cdot (T + 1) - 4^2 - 4^2, N = T + 1 \geq 26, T \geq 25$$

$$\overset{1}{F} = (M, N, a, b) = (13, T + 1, 4, 4), n_{\min} = 26 \cdot 25 - 6 = 650 - 6 = 644$$

$$4. n = 26T - 8 = 2 \cdot 13 \cdot (T + 1) - 5^2 - 3^2, N = T + 1 \geq 26,$$

$$T \geq 25, \overset{1}{F} = (M, N, a, b) = (13, T + 1, 5, 3)$$

$$n_{\min} = 26 \cdot 25 - 8 = 650 - 8 = 642$$

$$5. n = 26T - 10 = 2 \cdot 13 \cdot T - 3^2 - 1^2, T \geq 26$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 3, 1)$$

$$n_{\min} = 26 \cdot 26 - 10 = 676 - 10 = 666$$

$$6. n = 26T - 12 = 2 \cdot 13 \cdot (T + 3) - 9^2 - 3^2, N = T + 3 \geq 26, T \geq 23$$

$$\overset{1}{F} = (M, N, a, b) = (13, T + 3, 9, 3), n_{\min} = 26 \cdot 23 - 12 = 598 - 12 = 586$$

$$7. n = 26T - 14 = 26(T + 1) - 40 = 2 \cdot 13 \cdot (T + 1) - 6^2 - 2^2$$

$$N = T + 1 \geq 26, T \geq 25, \overset{1}{F} = (M, N, a, b) = (13, T + 1, 6, 2)$$

$$n_{\min} = 26 \cdot 25 - 14 = 650 - 14 = 636$$

$$8. n = 26T - 16 = 26(T + 2) - 68 = 2 \cdot 13 \cdot (T + 2) - 8^2 - 2^2$$

$$N = T + 2 \geq 26, T \geq 24, \overset{1}{F} = (M, N, a, b) = (13, T + 2, 8, 2)$$

$$n_{\min} = 26 \cdot 24 - 16 = 624 - 16 = 608$$

$$9. n = 26T - 18 = 2 \cdot 13 \cdot T - 3^2 - 3^2, N = T \geq 26,$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 3, 3), n_{\min} = 26 \cdot 26 - 18 = 676 - 18 = 658$$

$$10. n = 26T - 20 = 2 \cdot 13 \cdot T - 4^2 - 2^2, N = T \geq 26,$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 4, 2)$$

$$n_{\min} = 26 \cdot 26 - 20 = 676 - 20 = 656$$

$$11. \quad n = 26T - 22 = 26(T + 2) - 74 = 2 \cdot 13 \cdot (T + 2) - 7^2 - 5^2$$

$$N = T + 2 \geq 26, \quad T \geq 24 \quad \overset{1}{F} = (M, N, a, b) = (13, T + 2, 7, 5)$$

$$n_{\min} = 26 \cdot 24 - 22 = 624 - 22 = 602$$

$$12. \quad n = 26T - 24 = 26(T + 1) - 50 = 2 \cdot 13 \cdot (T + 1) - 7^2 - 1^2$$

$$N = T + 1 \geq 26, \quad T \geq 25 \quad \overset{1}{F} = (M, N, a, b) = (13, T + 1, 7, 1)$$

$$n_{\min} = 26 \cdot 25 - 50 = 650 - 50 = 600$$

$$13. \quad n = 26T - 26 = 2 \cdot 13 \cdot T - 5^2 - 1^2 \quad N = T \geq 26,$$

$$\overset{1}{F} = (M, N, a, b) = (13, T, 5, 1)$$

$$n_{\min} = 26 \cdot 26 - 26 = 650. \quad \max(n_{\min}) = 671. \quad \text{Теорема 1 доказана.}$$

2. Построение восьмиугольников, состоящих из любых n равных, не равнобедренных треугольников

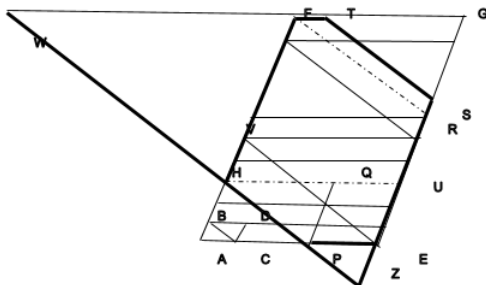
Теорема 2. Из любого числа $n \geq 671$ одинаковых не равнобедренных треугольников, можно сложить выпуклый восьмиугольник.

Доказательство. Пусть построен шестиугольник, состоящий из $n \geq 671$ произвольных равных не равнобедренных треугольников ABC . См. рис. 1. и теорему 1. Проведем прямые $EV \parallel HP$, $FR \parallel TS$. Ясно, что фигура $FTSR$ состоит из $M^2 - b^2$ треугольников ABC , а фигура $VEPH$ состоит из $M^2 - a^2$ треугольников ABC . Фигура $FTSR$ состоит из $M^2 - b^2$ треугольников ABC . Фигура $FREV$ состоит из $2M(N - M)$ треугольников ABC , Шестиугольник состоит из $n = 2MN - a^2 - b^2$ треугольников ABC , Повернем на 180° фигуру $FTSR$ относительно перпендикуляра к FR , проходящего через его середину. Аналогично, повернем на 180° фигуру $VEPH$ относительно перпендикуляра к VE , проходящего через его середину. Тогда получается выпуклый восьмиугольник $FTSRUEPH$, состоящий из такого же количества треугольников, что и шестиугольник, то есть из $n = 2MN - a^2 - b^2$ треугольников ABC (Здесь $M = 13, 0 < a, b < N$), $n \geq 671$. Теорема доказана.

3. Геометрическое доказательство формулы: $n = A^2 - B^2 - C^2 - D^2$

Теорема 3. Любое натуральное число $n \geq 671$ можно выразить через натуральные числа в виде $n = A^2 - B^2 - C^2 - D^2$.

Доказательство. По теореме 1 при $n \geq 671$ шестиугольник строится из любых равных треугольников ABC . Если достроить шестиугольник до треугольника, как на рис.3, то шестиугольник получается отрезанием от треугольника WGZ трех уголков-треугольников: WFH , TGS , PEZ . Отсюда верно тождество $n = (N + M - a)^2 - (M - a)^2 - (N - b)^2 - (N - a)^2$. Действительно, $FH = (N - a)AB$, $WH = (N - a)BC$, $PE = (M - a)AC$, $PZ = (M - a)BC$, $VE = HZ = M \cdot BC = (N - a)AB$, $WZ = (N - a)BC + M \cdot BC = (M + N - a) \cdot BC$, $TS = (M - a)BC$. Отсюда, Из подобия треугольников: ΔWGH состоит из $(M + N - a)^2$ треугольников ABC , ΔTGS состоит из b^2 треугольников ABC , ΔWFH состоит из $(N - a)^2$ треугольников ABC , ΔPEZ состоит из $(M - a)^2$ треугольников ABC . Поэтому шестиугольник $FTSEPH$ состоит из $n = (M + N - a)^2 - (N - a)^2 - b^2 - (M - a)^2$ треугольников ABC . Теорема доказана.



Заключение. В работе систематически устанавливается связь между геометрическими и алгебраическими представлениями. Результат также устанавливает эту связь. Действительно, доказано, что шестиугольник строится из $n \geq 671$ любых равных треугольников, и, что в этом случае любое натуральное число $n = A^2 - B^2 - C^2 - D^2$, где A, B, C, D - натуральные числа. В естественном смысле, все натуральные числа $n \geq 671$, можно назвать “шестиугольными” и “восьмиугольными”. Указанный подход является естественным, наглядным, и плодотворным. Аналогично, можно показать, что, если из n равных произвольных не равнобедренных треугольников строится пятиугольник (отрезанием одного уголка от параллелограмма), то строится и семиугольник, состоящий из того же числа треугольников. Естественно ожидать применение этой работы в комбинаторной геометрии и в теории чисел.

Список литературы

1. К.Чандрасекхаран. Введение в аналитическую теорию чисел. М., Мир, 1974.
2. В.А.Свентковский. Некоторые задачи и проблемы комбинаторной геометрии. Математика и ее приложения. Выпуск 1. Санкт Петербург, 2008.
3. Ю.И. Манин. Введение в современную теорию чисел. Москва, МЦНМО, 2013.

УДК 512.556

О полукольцах непрерывных частичных функций

Вечтомов Е.М., Лубягина Е.Н.

Вятский государственный университет

vecht@mail.ru

Аннотация

Работа относится к общей теории полуколец непрерывных функций. Рассматриваются идеалы и конгруэнции полуколец $CP(X, S)$ непрерывных частичных функций со значением в топологическом полукольце S . Рассмотрены вопросы двойственности и определяемости.

Введение. Работа посвящена исследованиям в области полуколец непрерывных S -значных функций над топологическими пространствами X .

Полукольца непрерывных функций возникли в рамках классической теории колец $C(X)=C(X, \mathbb{R})$ непрерывных действительных функций [1].

С теорией полуколец непрерывных неотрицательных действительных функций на топологических пространствах можно ознакомиться в монографии [2]. Рассматривались также другие полукольца значений. Полукольцам непрерывных $[0, 1]$ -значных функций посвящена статья [3]. Частичные кольца непрерывных $[-1; 1]$ -значных функций изучались в [4], а частичные полукольца непрерывных $[0, \infty]$ -значных функций – в [5]. Непрерывные $(0, \infty]$ -значные функции рассматривались в [6].

Полугруппам непрерывных частичных функций посвящена работа [7]. Исследование полуколец $CP(X, S)$ непрерывных частичных функций начато в [8]. Полукольца $CP(X)=CP(X, \mathbb{R})$ рассматривались также в [9].

В работе в терминах полуколец $CP(X, S)$ установлена определяемость T_1 -пространств пространств X , получены характеристики некоторых топологических свойств пространств X .

Полукольца частичных функций

Полукольцом называется непустое множество S с бинарными операциями сложения $+$ и умножения \cdot , для которых $\langle S, + \rangle$ – коммутативная полугруппа, $\langle S, \cdot \rangle$

– полугруппа и умножение дистрибутивно относительно сложения с обеих сторон.

Для полукольца S и произвольного множества X через $SP^X = \bigcup \{S^Y : Y \subseteq X\}$ обозначается множество всех частичных функций из X в S . Через $D(f)$ – область определения частичной функции f . На множестве SP^X определены полукольцевые операции по правилу: на $D(f+g)=D(f \cdot g)=D(f) \cap D(g)$ функции $f+g$ и fg определяются поточечно, то есть $(f \dot{+} g)(x) = f(x) \dot{+} g(x)$ для всех $x \in D(f) \cap D(g)$. В результате множество SP^X становится полукольцом с поглощающим элементом \emptyset .

Отношение $\rho_D: f \rho_D g \Leftrightarrow D(f)=D(g)$ служит конгруэнцией на полукольце SP^X , причем, фактор-полукольцо SP^X / ρ_D является моно-полукольцом, изоморфным булеану $\langle \mathbf{B}(X), \cap, \cup \rangle$ множества X , рассматриваемому с одной операцией пересечения \cap .

К любому полукольцу S можно внешним образом присоединить поглощающий элемент ∞ , в результате чего получим новое полукольцо $S \cup \{\infty\}$ с поглощающим элементом. Полукольцо SP^X изоморфно полукольцу $(S \cup \{\infty\})^X$ всюду определенных $S \cup \{\infty\}$ -значных функций на X .

Для топологического полукольца S и топологического пространства X через $C(X, S)$ обозначается полукольцо всех непрерывных S -значных функций на X , а через $CP(X, S) = \bigcup \{C(Y, S) : Y \subseteq X\}$ – полукольцо всех непрерывных частичных S -значных функций на X с поточечными операциями сложения и умножения частичных функций f и g на их общей области определения $D(f) \cap D(g)$.

Если пространство X дискретно либо полукольцо S антидискретно, то $C(X, S) = S^X$ и $CP(X, S) = SP^X$. Ясно, что полукольцо $C(X, S)$ является подполукольцом полукольца S^X , а $CP(X, S)$ есть подполукольцо в SP^X и $CP(X, S)$ также содержит поглощающий элемент \emptyset . Если полукольцо S имеет 1, то и CP будет полукольцом с единицей $\mathbf{1} \in C(X, S)$. Фактор-полукольцо $CP(X, S) / \rho_D$ снова изоморфно моно-полукольцу $\langle \mathbf{B}(X), \cap, \cup \rangle$.

Напомним, что топологическое пространство X называется T_1 -пространством, если для любых его различных точек x и y существует открытое в X множество U , содержащее x и не содержащее y . T_1 -пространство X называется тихоновским, если для любой его точки x и несодержащего ее замкнутого множества B , существует непрерывная функция $f: X \rightarrow \mathbf{R}$, равная нулю на множестве B и единице в точке x .

Для каждого топологического пространства X существует тихоновское пространство τX , такое, что $X \subseteq \tau X$ и кольцо $C(X)$ канонически изоморфно кольцу $C(\tau X)$. Тихоновость пространства X означает, что $X = \tau X$ [10].

Лемма 1. Пусть топологическое полукольцо S содержит единичный элемент 1 и $\{1\}$ есть собственное замкнутое подмножество в S , а X – произвольное T_1 -пространство. Множество $A \subseteq X$ замкнуто тогда и только тогда, когда $\forall e_x \in CP(X, S)$ ($e_A e_x = \emptyset \Rightarrow \exists f \in CP(X, S)$ ($f e_A = e_A$ и $\emptyset \neq f e_x \neq e_x$)).

Доказательство. Замкнутость множества A в X эквивалентна условию: $\forall x \in X$ ($x \notin A \Rightarrow \exists$ замкнутое в X множество $B \supseteq A$, $x \notin B$).

Пусть A – замкнутое подмножество T_1 -пространства X и $x \in X \setminus A$. Рассмотрим функцию $f \in SP^X$: $D(f) = A \cup \{x\}$, $f = 1$ на A , $f(x) \neq 1$. Частичная функция f непрерывна на своей области определения, для нее $f e_A = e_A$ и $\emptyset \neq f e_x \neq e_x$.

Обратно, пусть для произвольной точки $x \in X \setminus A$ найдется функция $f \in CP(X, S)$, для которой $f e_A = e_A$ и $\emptyset \neq f e_x \neq e_x$. Тогда множество $f^{-1}(1)$ замкнуто в $D(f)$ и $f^{-1}(1) \supseteq A$, $x \in D(f) \setminus f^{-1}(1)$. Значит, подмножество A замкнуто в X .

Имеет место следующее (см. [7])

Предложение 1. Пусть S – неоднородное топологическое полукольцо с единицей 1 с замкнутым множеством $\{1\}$ и X, Y – произвольные T_1 -пространства. Тогда эквивалентны следующие условия:

- 1) X и Y гомеоморфны;
- 2) полукольца $CP(X, S)$ и $CP(Y, S)$ изоморфны;
- 3) мультипликативные полугруппы $CP(X, S)$ и $CP(Y, S)$ изоморфны.

Идеалы и конгруэнции полуколец $CP(X, S)$

Идеалом коммутативного полукольца S называется всякое его непустое подмножество I , такое, что для любых $a, b \in I$; $s \in S$ выполняется: $as \in I$, $a + b \in I$.

Пусть S содержит единицу 1 . Важную роль в полукольцах SP^X играют унитарные идемпотенты (по умножению) e_A , $A \subseteq X$:

$$D(e_A) = A \text{ и } e_A(x) = 1 \text{ для всех } x \in A.$$

В частности получаем идемпотенты $e_x = e_{\{x\}}$ для произвольного $x \in X$, $e_x = 1$ и $e_\emptyset = \emptyset$. Элемент e_A служит единицей полуколец S^A и SP^A . При этом полукольцо $SP^A = e_A SP^X$ является главным идеалом полукольца SP^X .

Пусть полукольцо S содержит нулевой элемент 0 . Получаем идемпотенты θ_A , $A \subseteq X$:

$$D(\theta_A) = A \text{ и } \theta_A(x) = 0 \text{ для всех } x \in A.$$

$$\text{В частности, получаем } \theta_x = \theta_{\{x\}} \text{ для любого } x \in X, \theta_x = 0 \text{ и } \theta_\emptyset = e_\emptyset = \emptyset.$$

Очевидно следующие утверждения:

Лемма 2. Для любого идеала I полукольца $CP(X, S)$ и подмножества $D \subseteq X$ если множество $I \cap C(D, S)$ не пусто, то оно является идеалом в $C(D, S)$.

Предложение 2. Для любых топологического пространства X и топологического полукольца S с нулем минимальными идеалами полукольца

$CP(X, S)$ будут в точности главные идеалы $(\theta_x) = \theta_x CP(X, S) = \{\theta_x, \emptyset\}$ для $x \in X$.

Заметим, что в $CP(X, \mathbf{R})$ для любой точки $x \in X$ идеал (e_x) , как полукольцо изоморфный $\mathbf{R} \cup \{\emptyset\}$, содержит ровно два идеала

$(e_x) \supset (\theta_x) \supset \{\emptyset\}$ полукольца $CP(X, \mathbf{R})$.

Идеал I полукольца $CP(X, S)$ называется:

– биидеалом, если $I + CP(X, S) \subseteq I$;

– D -идеалом, если $f \in I, g \in CP(X, S)$ и $D(f) = D(g)$ влекут $g \in I$.

Предложение 3. Для любых топологического пространства X и топологического кольца S с единицей в полукольце $CP(X, S)$ биидеалы совпадают с D -идеалами и исчерпываются всеми объединениями главных идеалов вида $(e_A), A \subseteq X$.

Доказательство. Пусть идеал I полукольца $CP(X, S)$ является D -идеалом. Рассмотрим функции $f \in I$ и $g \in CP(X, S)$. Покажем, что их сумма $f+g$ лежит в идеале I . Возьмем функцию $h = gf$. Имеем $h \in I$ и $D(h) = D(f) \cap D(g) = D(f+g)$. Значит, $f+g \in I$ и I – биидеал.

Пусть теперь идеал I полукольца $CP(X, S)$ является биидеалом. Рассмотрим функции $f \in I$ и $g \in CP(X, S)$, такие что $D(g) = D(f) = D$. Тогда $f+g, f, g \in C(D, S)$. Так как I – биидеал, то $f+g \in I$. По лемме 2 непустое множество $J = I \cap C(D, S)$ – идеал в кольце $C(D, S)$. Имеем $f \in J$ и $f+g \in J$. Выразим функцию g через функции $f+g$ и f : $g = (f+g) - f = (f+g) + (-1)f$. Получаем, что $g \in J \subseteq I$. Значит, I – D -идеал.

Последнее утверждение очевидно.

Предложение 4. Для любых топологического пространства X и топологического полукольца S с единицей максимальные идеалы полукольца $CP(X, S)$ имеют вид $(CP(X, S) \setminus C(X, S)) \cup M$, где M – произвольный максимальный идеал, полукольца $C(X, S)$.

Доказательство. Ясно, что $(CP(X, S) \setminus C(X, S)) \cup M$ – максимальный идеал в $CP(X, S)$ для всякого максимального идеала M полукольца $C(X, S)$.

Покажем, что других максимальных идеалов в полукольце $CP(X, S)$ нет. Пусть I – максимальный идеал полукольца $CP(X, S)$. Тогда $1 \notin I$. Если $I \cap C(X, S) = \emptyset$, то $I \subseteq (CP(X, S) \setminus C(X, S)) \cup M$ для любого идеала M полукольца $C(X, S)$ – противоречие. Если $J = I \cap C(X, S) \neq \emptyset$, то по лемме 2 J – это идеал полукольца $C(X, S)$, отличный от $C(X, S)$. Получаем, что $J \subseteq M$ для некоторого максимального идеала M полукольца $C(X, S)$. Таким образом, $I \subseteq (CP(X, S) \setminus C(X, S)) \cup M$. Поскольку $1 \notin M$, то $I = (CP(X, S) \setminus C(X, S)) \cup M$. \square

Отношение эквивалентности ρ на полукольце S называется конгруэнцией, если ρ сохраняет полукольцевые операции, то есть $\forall a, b, a_1, b_1 \in S (a \rho b \text{ и } a_1 \rho b_1 \Rightarrow (a+a_1) \rho (b+b_1) \text{ и } (aa_1) \rho (bb_1))$.

Конгруэнция ρ на полукольце $CP(X, S)$ называется D-конгруэнцией, если $f, g \in CP(X, S)$ и $D(f)=D(g)$ влекут $f \rho g$.

Наименьшей D-конгруэнцией на полукольце $CP(X, S)$ служит конгруэнция ρ_D .

Предложение 5. Для любого топологического пространства X и топологического кольца S с единицей максимальные конгруэнции на полукольце $CP(X, S)$ совпадают с двухклассовыми D-конгруэнциями.

Доказательство. Достаточно показать, что если $\rho \vee \rho_D = 1$, то $\rho = 1$. Получаем, что максимальные конгруэнции на $CP(X, S)$ являются D-конгруэнциями и имеют вид $\rho \vee \rho_D$ по всем конгруэнциям ρ на $CP(X, S)$. Подрешетка D-конгруэнций на $CP(X, S)$ изоморфна решетке конгруэнций идемпотентного моно-полукольца $\langle B, \cap, \cup \rangle$. Из предложения 6.1.5 [11] максимальные D-конгруэнции на $CP(X, S)$ будут двухклассовыми.

Каждой функции $f \in C(X)$ соответствуют нуль-множество $Z(f)=f^{-1}(0)$ и его внутренность $Z_0(f)$.

Для идеалов (множеств) I и J обозначим идеалы (множества)

$$I+J = \{f+g: f \in I; g \in J\},$$

$$I \cdot J = \left\{ \sum_{i=1}^n f_i g_i : f_i \in I; g_i \in J; n \in \mathbb{N} \right\}.$$

Множество $\text{Id}CP(X, S)$ всех идеалов полукольца $CP(X, S)$ относительно теоретико-множественного включения \subseteq есть решетка с операциями $\sup(I, J) = I \cup J \cup (I+J)$ и $\inf(I, J) = I \cap J$.

Предложение 6. Для любых топологического пространства X и топологического кольца S с единицей решетка $\text{Id}CP(X, S)$ модулярна.

Доказательство. Пусть A, B, C – произвольные идеалы полукольца $CP(X, S)$ и $B \subseteq A$. Нужно показать, что $A \wedge (B \vee C) \subseteq (A \wedge B) \vee (A \wedge C)$, то есть

$$A \cap (B \cup C \cup (B+C)) \subseteq (A \cap B) \cup (A \cap C) \cup (A \cap B + A \cap C).$$

Достаточно доказать, что

$$A \cap (B+C) \subseteq A \cap B + A \cap C.$$

Пусть для функции $f \in A \cap (B+C)$ выполняется равенство $f=g+h$, где $g \in B \subseteq A$ и $h \in C$. Тогда $D(f)=D(g) \cap D(h)=D$ и функции $g \in D$, $h \in D$, $f=g \in D + h \in D$ принадлежит кольцу $C(D, S)$. По лемме 1 $A' = A \cap C(D, S)$, $B' = B \cap C(D, S)$, $C' = C \cap C(D, S)$ – идеалы в $C(D, S)$.

Так как $g \in D \subseteq B' \subseteq A'$, $h \in D = f - g \in A' \cap C'$, то $f \in A' \cap B' + A' \cap C' \subseteq A \cap B + A \cap C$. Значит, $A \cap (B+C) \subseteq A \cap B + A \cap C$. Следовательно, решетка $\text{Id}CP(X, S)$ модулярна.

Заметим, что для неодноэлементного тихоновского пространства X решетка конгруэнций полукольца $CP(X)$ не модулярна.

Наименьшим и наибольшим элементами решетки $IdCP(X, S)$ являются идеалы $\{\emptyset\}$ и $CP(X, S)$.

Для произвольного идеала $I \in IdCP(X, S)$ обозначим $D(I) = \bigcup_{f \in I} D(f)$.

Любому идеалу $J \subseteq CP(X, S)$ соответствует идеал $CP(X \setminus D(I), S) = \{f \in CP(X, S) : D(f) \subseteq X \setminus D(I)\} = \{f \in CP(X, S) : D(f) \cap D(I) = \emptyset\} = \{f \in CP(X, S) : fg = \emptyset \text{ для всех } g \in J\}$.

Псевдодополнением элемента a решетки $\langle L, \vee, \wedge, 0, 1 \rangle$ называется наибольший элемент $a^* \in L$, удовлетворяющий условию $a \wedge a^* = 0$. Дополнением элемента a решетки $\langle L, \vee, \wedge, 0, 1 \rangle$ называется элемент $a' \in L$ удовлетворяющий условиям $a \wedge a' = 0$ и $a \vee a' = 1$.

Предложение 7. Для любых топологического пространства X и топологического полукольца S с нулем и единицей $IdCP(X, S)$ – решетка с псевдодополнениями. Дополнениями в $IdCP(X, S)$ обладают только элементы $\{\emptyset\}$ и $CP(X, S)$.

Доказательство. Для произвольных идеалов A и B полукольца $CP(X, S)$ имеем

$$A \cdot B = \{\emptyset\} \Leftrightarrow A \cap B = \{\emptyset\}.$$

Поэтому для любого идеала A полукольца $CP(X, S)$ идеал $CP(X \setminus D(A), S)$ будет наибольшим идеалом среди идеалов B с условием $A \cap B = \{\emptyset\}$, то есть является псевдодополнением к A .

Возьмем произвольный идеал A полукольца $CP(X, S)$. Для его дополнения A' получаем $1 \in A \cup A' \cup (A + A')$. Если $1 \in A$, то $A = CP(X, S)$, если $1 \in A'$, то $A = \{\emptyset\}$. Пусть $1 \in A + A'$. Тогда $1 = f + g$ для некоторых функций $f \in A, g \in A'$. Значит, $0 \in A \cap A'$ – противоречие. \square

Получаем, что для произвольного подмножества $Y \subseteq X$ идеал $CP(Y, S)$ будет псевдодополнением к идеалу $CP(X \setminus Y, S)$. Итак, псевдодополнения элементов решетки $IdCP(X)$ совпадают с идеалами вида $CP(Y)$ по всем подмножествам $Y \subseteq X$.

Предложение 8. Пусть X – произвольное T_1 -пространство и S – топологическое полукольцо с нулем и единицей. Тогда эквивалентны следующие условия:

- 1) X и Y гомеоморфны;
- 2) решетки $IdCP(X, S)$ и $IdCP(Y, S)$ изоморфны.

Доказательство. Произвольное T_1 -пространство X можно сконструировать, как множество атомов $\{(\theta_x) : x \in X\}$ с системой замкнутых множеств $\{\text{псевдодополнения элементов } CP(Y) : \forall (\theta_x) \in IdCP(X, S) \text{ подрешетка}$

идеалов между $CP(Y) \vee (\theta_x)$ и наименьшим псевдодополнением $CP(Y \cup \{x\}) \supseteq CP(Y) \vee (\theta_x)$ не цепь}, соответствующих замкнутым множествам $Y \subseteq X$.

Идеалы и конгруэнции полуколец $CP(X)$

Далее в качестве полукольца значений возьмем топологическое поле R действительных чисел.

Точка $x \in X$ называется P -точкой, если $x \in Z(f)$ влечет $x \in Z_0(f)$ для любой $f \in C(X)$. Топологическое пространство X называется P -пространством, если все его точки являются P -точками. Это равносильно ([1], chapter 14) тому, что кольцо $C(X)$ регулярно по фон Нейману, то есть для любой функции $f \in C(X)$ существует функция $g \in C(X)$ такая, что $fgf=f$.

Известно ([1], chapter 14), что любое подмножество P -пространства является P -пространством. Получаем, что полукольцо $CP(X)$ регулярно тогда и только тогда, когда пространство X будет P -пространством.

Точка x пространства X называется F -точкой, если для любых функций $f, g \in C(X)$ из того, что точка $x \in Z_0(fg)$, следует что $x \in Z_0(f)$ или $x \in Z_0(g)$. Топологическое пространство X называется F -пространством, если все его точки являются F -точками.

Известно, что для любого топологического пространства X решетка идеалов $IdC(X)$ дистрибутивна тогда и только тогда, когда X – F -пространство [12].

Не всякое подмножество F -пространства само будет F -пространством ([1], chapter 14). Топологическое пространство X называется наследственным F -пространством, если любое подмножество X является F -пространством. Отметим, что

Предложение 9. Для любого топологического пространства X решетка идеалов $IdCP(X)$ дистрибутивна тогда и только тогда, когда X – наследственное F -пространство.

Доказательство. Если решетка идеалов $IdCP(X)$ дистрибутивна, то дистрибутивны решетки идеалов $IdCP(Y)$ для всех подпространств $Y \subseteq X$, что возможно лишь когда Y – F -пространство. То есть X – наследственное F -пространство.

Пусть теперь X – наследственное F -пространство и A, B, C – произвольные идеалы полукольца $CP(X)$. Покажем, что $A \wedge (B \vee C) \subseteq (A \wedge B) \vee (A \wedge C)$, то есть

$$A \cap (B \cup C \cup (B+C)) \subseteq (A \cap B) \cup (A \cap C) \cup (A \cap B + A \cap C).$$

Достаточно доказать, что

$$A \cap (B+C) \subseteq A \cap B + A \cap C.$$

Пусть для функции $f \in A \cap (B+C)$ выполняется равенство $f=g+h$, где $g \in B$ и $h \in C$. Снова обозначим $D=D(f)=D(g) \cap D(h)$. Функции geD , heD и $f=geD+heD$ принадлежат кольцу $C(D)$, для которого решетка идеалов $IdC(D)$ дистрибутивна.

Значит, для идеалов $A'=A \cap C(D)$, $B'=B \cap C(D)$, $C'=C \cap C(D)$ из $\text{Id}C(D)$ выполняется $A' \cap (B'+C') \subseteq A' \cap B'+A' \cap C'$. Получаем, что $f \in A' \cap (B'+C')$ и $f=g'+h'$ для подходящих функций $g' \in A' \cap B'=A \cap B \cap C(D)$, $h' \in A' \cap C'=A \cap C \cap C(D)$, то есть $f \in A \cap B+A \cap C$. Значит, $A \cap (B+C) \subseteq A \cap B+A \cap C$. Поэтому решетка $\text{Id}CP(X)$ дистрибутивна.

Двойственность

Отметим на полукольце $CP(X)$ естественный порядок \leq : для всех $f, g \in CP(X)$ $f \leq g$ означает, что

$$D(f) \subseteq D(g) \text{ и } f(x) \leq g(x) \text{ для всех } x \in D(f).$$

Порядок \leq порождает операцию \wedge :

$$D(f \wedge g) = D(f) \cap D(g) \text{ и}$$

$$(f \wedge g)(x) = \min(f(x), g(x)) \text{ для всех } x \in D(f \wedge g)$$

Получаем нижнюю полурешетку $\langle CP(X), \wedge \rangle$ с наименьшим элементом \emptyset .

Упорядоченное множество $\langle \{eA: A \subseteq X\}, \leq \rangle$ всех унитарных идемпотентов изоморфно булеану $B(X)$ подмножеств пространства X . Имеем $eA \wedge eB = eA \cap B$, $eA \vee eB = eA \cup B$ для любых $A, B \subseteq X$.

Существование $f \vee g$ для всех $f, g \in CP(X)$ равносильно дискретности пространства X .

Полукольцевой гомоморфизм $\alpha: CP(X) \rightarrow CP(Y)$, сохраняющий единицу 1, назовем \vee -полным, если α сохраняет точную верхнюю грань любого множества $\{eA: A \subseteq F\}$, $F \subseteq B(X)$, унитарных идемпотентов: $\alpha(\vee eA) = \vee \alpha(eA)$ по всем $A \in F$.

Любое непрерывное отображение $\varphi: Y \rightarrow X$ топологических пространств индуцирует полукольцевой гомоморфизм $\bar{\varphi}: CP(X) \rightarrow CP(Y)$ по формуле: для всех $f \in CP(X)$ на $D(\bar{\varphi}(f)) = \varphi^{-1}(D(f))$ имеем $\bar{\varphi}(f)(y) = f(\varphi(y))$.

Ясно, что индуцированные гомоморфизмы сохраняют функции-константы, и унитарные идемпотенты eA , $A \subseteq X$, переводят в унитарные идемпотенты eB , где $B = \varphi^{-1}(A) \subseteq Y$, поглощающий элемент в поглощающий.

Предложение 10. X – произвольное T_1 -пространство. Полукольцевой гомоморфизм $CP(X) \rightarrow CP(Y)$ будет индуцированным тогда и только тогда, когда он является \vee -полным.

Доказательство. Пусть полукольцевой гомоморфизм $\alpha: CP(X) \rightarrow CP(Y)$ индуцирован отображением $\varphi: Y \rightarrow X$. Тогда α сохраняют единицу 1. Для любого множества F подмножеств $A \subseteq X$ обозначим $\cup A \in FA = D$. Тогда $D(\alpha(\vee eA)) = \varphi^{-1}(D(\vee eA)) = \varphi^{-1}(D)$. Для любой точки $y \in \varphi^{-1}(D)$ имеем $\alpha(\vee_{A \in F} eA)(y) =$

$\alpha(eD)(y) = (eD)(\varphi(y)) = 1 = \bigvee_{A \in F} eA(\varphi(y)) = \bigvee_{A \in F} \alpha(eA)(y)$. Значит, α является \vee -полным.

Пусть полукольцевой гомоморфизм $\alpha : CP(X) \rightarrow CP(Y)$ является \vee -полным. Тогда для любого $y \in Y$ найдется $x \in X$ такой, что $\alpha(ex) = eB$, $B \subseteq Y$, $y \in B$. Действительно, существование точки $y \in Y$ такой, что для любого $x \in X$ $\alpha(ex) = eB$, $B \subseteq Y \setminus \{y\}$, противоречит \vee -полноте гомоморфизма: $\alpha(\bigvee_{x \in X} ex) = 1 \neq \bigvee_{x \in X} \alpha(ex) = eC$, $y \notin C$, по всем $x \in X$.

Для различных точек $x_1, x_2 \in X$ имеем $\alpha(ex_1) \cdot \alpha(ex_2) = \alpha(ex_1 \cdot ex_2) = \alpha(\emptyset) = \emptyset$. Таким образом, для любого $y \in Y$ найдется единственный $x \in X$ такой, что $\alpha(ex) = eB$, $B \subseteq Y$, $y \in B$.

Зададим отображение $\varphi : Y \rightarrow X$ по правилу $\varphi(y) = x \Leftrightarrow \exists B \subseteq Y, y \in B: \alpha(ex) = eB$.

Покажем, что отображение φ непрерывно в каждой точке $y \in Y$. То есть для любой окрестности U точки $\varphi(y) \in X$ множество $W = \varphi^{-1}(U)$ открыто.

По лемме 1 имеем: W открыто в $Y \Leftrightarrow Y \setminus W$ замкнуто $\Leftrightarrow \forall ey \in CP(Y) (eY \setminus Wey = \emptyset \Rightarrow \exists f \in CP(Y) (feY \setminus W = eY \setminus W \text{ и } \emptyset \neq fey \neq ey)) \Leftrightarrow \forall y \in Y (y \in Y \Rightarrow \exists f \in CP(Y): f(Y \setminus W) = 1 \text{ и } f(y) = 1)$ (*).

Зафиксируем произвольную точку $y' \in W$, обозначим $x' = \varphi(y') \in U$. В $CP(X)$ возьмем функцию $f: D(f) = (X \setminus U) \cup \{x'\}$, $f(X \setminus U) = 1$, $f(x') = 0$.

Имеем $e_{\varphi^{-1}(x')} = \alpha(ex') = \alpha(\theta_{x'+ex'}) = \theta_{C+e\varphi^{-1}(x')}$, $\alpha(\theta_{x'}) = \theta_C$ для некоторого $C \subseteq Y$, то есть $y' \in \varphi^{-1}(x') \subseteq C$. Для функции $\alpha(f) \in CP(Y)$ получаем

$\alpha(f)(y') = \alpha(f)e_{\varphi^{-1}(x')}(y') = \alpha(f)\alpha(ex')(y') = \alpha(fex')(y') = \alpha(\theta_{x'})(y') = 0$. Так как $W = \varphi^{-1}(U)$, то для любой точки $y'' \notin W$ получаем, что $x'' = \varphi(y'') \notin U$ и $\alpha(ex'') = eB$, $y \in B$. Для любой $y'' \in Y \setminus W$ получаем $\alpha(f)(y'') = \alpha(fex'')(y'') = \alpha(ex'')(y'') = 1$. Итак, W удовлетворяет условиям (*), то есть открыто. \square

Обозначим через K категорию всех T1-пространств X и их непрерывных отображений φ , а через C – категорию всех полуколец $CP(X)$ и их \vee -полных гомоморфизмов α . Для любых непрерывных отображений $\psi : Z \rightarrow Y$ и $\varphi : Y \rightarrow X$ имеем $\varphi \circ \psi = \psi \circ \varphi$. Поэтому соответствие $F, F(X) = CP(X)$ и $F(\varphi) = \varphi$, является контрвариантным функтором из категории K в категорию C .

Предложение 10. Категория K антиэквивалентна (двойственна) категории C .

Список литературы

1. Gillman L., Jerison M. Rings of continuous functions. N. Y.: Springer-Verlang, 1976. 300 p.
2. Вечтомов Е. М., Сидоров В. В., Чупраков Д. В. Полукольца непрерывных функций. Киров: Изд-во ВятГУ, 2011. 312 с.
3. Вечтомов Е. М., Лубягина Е. Н. Полукольца непрерывных $[0, 1]$ -значных функций // Фундаментальная и прикладная математика. 2012. Т. 17. Вып. 1. С. 53–82.
4. Лубягина Е. Н. Частичные кольца непрерывных функций // Научно-технический вестник Поволжья: физико-математические науки, химические науки, технические науки. 2015. № 3. С. 44–46.
5. Вечтомов Е. М., Шалагинова Н. В. Простые идеалы в частичных полукольцах непрерывных $[0, \infty]$ -значных функций // Вестник Пермского университета. Сер.: Математика. Механика. Информатика. 2014. Вып. 1 (24). С. 5–12.
6. Вечтомов Е. М., Шалагинова Н. В. Об идеалах в полукольцах непрерывных $(0, \infty]$ -значных функций // Алгебра, теория чисел и дискретная геометрия: современные проблемы и приложения: материалы XIII Международной конференции. Тула: ТГПУ, 2015. С. 153–155.
7. Вечтомов Е. М. О полугруппах непрерывных частичных функций на топологических пространствах // УМН, 1990. Т. 46. Вып. 4. С. 143–144.
8. Вечтомов Е. М., Лубягина Е. Н. Полукольца частичных функций // Алгебра, теория чисел и дискретная геометрия: современные проблемы и приложения: материалы XIII Международной конференции, посв. 85-летию со дня рождения проф. С. С. Рышкова. Тула, 2015. С. 148–150.
9. Лубягина Е. Н., Христолюбова И. А. Об идеалах и конгруэнциях в полукольцах частичных функций // Математический вестник педвузов и университетов Волго-Вятского региона: периодический межвузовский сборник научно-методических работ. Выпуск 18. Киров: Науч. изд-во ВятГУ, 2016. С. 82–88.
10. Вечтомов Е. М., Петров А. А. Полукольца с идемпотентным умножением: монография. Киров: Изд-во ООО «Радуга-ПРЕСС», 2015. 144 с.
11. Вечтомов Е. М. Дистрибутивные кольца непрерывных функций и F -пространства // Математические заметки. 1983. Т. 34. № 3. С. 321–332.
12. Энгелькинг Р. Общая топология. – М.: Мир, 1986. 752 с.

УДК 512.558

О конгруэнциях на мультипликативно идемпотентных полукольцах

Вечтомов Е.М., Петров А.А.

Вятский государственный университет

vecht@mail.ru

Аннотация

Вечтомов Е.М., Петров А.А. О конгруэнциях на мультипликативно идемпотентных полукольцах. Рассматриваются результаты о конгруэнциях на полукольцах с идемпотентным умножением, в том числе – о максимальных конгруэнциях. Основное внимание уделяется коммутативным мультипликативно идемпотентным полукольцам, в частности, полукольцам, мультипликативная полугруппа которых является цепью.

Полукольцом называется алгебраическая структура $\langle S, +, \cdot \rangle$ с двумя бинарными операциями сложения $+$ и умножения \cdot , такая, что: $\langle S, + \rangle$ – коммутативная полугруппа, $\langle S, \cdot \rangle$ – полугруппа, умножение дистрибутивно относительно сложения с обеих сторон.

Полукольцо называется коммутативным, если на нем тождественно $xu=ux$.

Полукольцо с тождеством $xx=x$ (с тождеством $x+x=x$) называется мультипликативно идемпотентным (соответственно, аддитивно идемпотентным). Полукольцо, одновременно мультипликативно идемпотентное и аддитивно идемпотентное, называется *идемпотентным*. Теория мультипликативно идемпотентных полуколец развита в работах [1, 2].

Полукольцо с тождеством $x+y=xy$ называется *моно-полукольцом*. Будем говорить, что полукольцо S обладает *константным сложением*, если оно удовлетворяет тождеству $x+y=u+v$.

Элемент θ произвольного полукольца S назовем *поглощающим по умножению (поглощающим по сложению)*, если для всех $x \in S$ выполняется $\theta x = x \cdot \theta = \theta$ (соответственно, $x + \theta = \theta$). Элемент $\infty \in S$, поглощающий по сложению и по умножению, называется *поглощающим*.

Если в полукольце S существует элемент 0 , нейтральный по сложению и поглощающий по умножению, то S называется *полукольцом с нулем* 0 . Наконец, если полукольцо S обладает элементом 1 , нейтральным по умножению, то S называется *полукольцом с единицей* 1 .

Отметим, что к любому полукольцу S можно естественным образом присоединить нулевой элемент 0 или поглощающий элемент ∞ . Обозначим полученные полукольца $S \cup \{0\}$ и $S \cup \{\infty\}$, соответственно.

Мультипликативная полугруппа любого коммутативного мультипликативно идемпотентного полукольца S является нижней полурешеткой, ее можно упорядочить следующим образом:

$$x < y \Leftrightarrow xy = x.$$

Аналогично, любое аддитивно идемпотентное полукольцо является верхней полурешеткой по сложению, в ней вводится отношение порядка \leq :

$$x \leq y \Leftrightarrow x + y = y.$$

Конгруэнцией на полукольце S называется отношение эквивалентности ρ на S , стабильное относительно операций:

$$arb \text{ и } cpd \text{ влекут } (a+c)\rho(b+d) \text{ и } (ac)\rho(bd) \text{ для любых } a, b, c, d \in S.$$

Множество $\text{Con } S$ всех конгруэнций на полукольце S является ограниченной решеткой относительно включения конгруэнций:

$$\rho \subseteq \tau \text{ означает, что } arb \Rightarrow atb \text{ для любых } a, b \in S.$$

Наименьшим элементом в $\text{Con } S$ служит нулевая конгруэнция 0_S – отношение равенства, наибольшим – единичная конгруэнция 1_S – одноклассовая. Полукольцо S называется подпрямо неразложимым, если на нем существует наименьшая ненулевая конгруэнция; конгруэнция-простым, если оно обладает ровно двумя конгруэнциями: отношением равенства и одноклассовой.

На произвольном полукольце S для любого фиксированного натурального числа $n \geq 2$ определим конгруэнцию $\approx_{(n)}$:

$$x \approx_{(n)} y \Leftrightarrow nx = ny.$$

В произвольном полукольце S вводится «разностное» отношение \leq :

$$x \leq y \Leftrightarrow x = y \text{ или } (\exists z \in S)(x + z = y).$$

Оно рефлексивно и транзитивно, но не обязательно антисимметрично. Если отношение \leq антисимметрично, то есть является отношением порядка, то такое полукольцо назовем упорядочиваемым.

Бинарное отношение \approx :

$$x \approx y \Leftrightarrow x \leq y, y \leq x$$

является конгруэнцией на произвольном полукольце.

Лемма 1. Для произвольного мультипликативно идемпотентного полукольца S справедливы следующие утверждения:

1. факторполукольцо $S / \approx_{(2)}$ является идемпотентным полукольцом;
2. факторполукольцо $S / \approx_{(3)}$ удовлетворяет тождеству $3x = x$;
3. факторполукольцо S / \approx удовлетворяет тождеству $3x = 2x$ и является упорядочиваемым полукольцом;
4. пересечение конгруэнций $\approx_{(3)}$ и \approx есть отношение равенства на S .

Следствие 1. Любое мультипликативно идемпотентное полукольцо является подпрямым произведением полуколец, одно из которых удовлетворяет тождеству $3x = x$, а другое удовлетворяет тождеству $3x = 2x$.

С точностью до изоморфизма существует ровно четыре двухэлементных коммутативных мультипликативно идемпотентных полуколец:

- двухэлементная цепь $\mathbf{B}=\{0,1\}$;
- двухэлементное поле $\mathbf{Z}_2=\{0,1\}$;
- двухэлементное идемпотентное моно-полукольцо $\mathbf{D}=\{1,\infty\}$ с единицей 1;
- двухэлементное полукольцо $\mathbf{T}=\{1,\infty\}$ с единицей 1 и константным сложением.

Предложение 1. Полукольца \mathbf{B} , \mathbf{Z}_2 , \mathbf{D} , \mathbf{T} – это в точности все конгруэнц-простые коммутативные мультипликативно идемпотентные полукольца.

Замечание 1. В работе [3] указан пример трехэлементного конгруэнц-простого полукольца T с некоммутативным идемпотентным умножением. Приведем таблицы Кэли его редуктов:

+	a	1	b
a	a	1	b
1	1	1	b
b	b	b	b

·	a	1	b
a	a	a	b
1	a	1	b
b	a	b	b

Построенное полукольцо с единицей 1 обладает следующими свойствами: по сложению имеем трехэлементную цепь $a < 1 < b$, мультипликативная же полугруппа является полугруппой правых нулей, то есть T удовлетворяет тождеству $xу=y$.

Предложение 2. Для любого полукольца S справедливы следующие утверждения:

1. S мультипликативно идемпотентно тогда и только тогда, когда классы любой конгруэнции ρ на S являются (выпуклыми) подполугруппами полугруппы $\langle S, \cdot \rangle$;

2. S идемпотентно тогда и только тогда, когда классы любой конгруэнции ρ на S являются подполукольцами в S .

Теорема 1. Для всякого подпрямо неразложимого коммутативного мультипликативно идемпотентного полукольца S справедливы следующие утверждения:

1. S имеет единицу 1 ;
2. множество $S \setminus \{1\}$ – наибольший идеал в S , обладающий единицей e ;
3. наименьшей ненулевой конгруэнцией на S служит конгруэнция, склеивающая только элементы 1 и e .

Пример 1. Если подпрямо неразложимое коммутативное мультипликативно идемпотентное полукольцо S обладает нулем 0 , то полукольцо $S \cup \{\infty\}$ также будет подпрямо неразложимым. В самом деле, для любых $a, b \in S$ и конгруэнции ρ на $S \cup \{\infty\}$ из $a\rho\infty$ следует $0\rho\infty$, так как $(a \cdot 0)\rho(\infty \cdot 0)$.

Тогда $b\rho\infty$, поскольку $(b+0)\rho(b+\infty)$. Таким образом, $\rho = \mathbf{1}_{S \cup \{\infty\}}$, откуда для любой нетривиальной конгруэнции ρ' на $S \cup \{\infty\}$ класс $[\infty]_{\rho'}$ одноэлементен. Значит, полукольцо $S \cup \{\infty\}$ подпрямо неразложимо вместе с S .

Аналогично проверяется, что для всякого подпрямо неразложимого коммутативного мультипликативно идемпотентного полукольца S с поглощающим элементом ∞ полукольцо $S \cup \{0\}$ является подпрямо неразложимым.

Коммутативное мультипликативно идемпотентное полукольцо назовем *цепным*, если полурешетка $\langle S, \cdot \rangle$ является цепью.

Пример 2. Обозначим $S_0 = \mathbf{B}$, Z_2 – одно из двухэлементных коммутативных мультипликативно идемпотентных полуколец с нулевым элементом, а $T_0 = \mathbf{D}$, \mathbf{T} – одно из двухэлементных полуколец с поглощающим элементом. Тогда из примера 1 полукольца $S_1 = S_0 \cup \{\infty\}$ и $T_1 = T_0 \cup \{\infty\}$ подпрямо неразложимы. Снова имеем подпрямо неразложимые полукольца $S_2 = S_1 \cup \{0\}$ и $T_2 = T_1 \cup \{\infty\}$.

Продолжая процесс далее, для любого натурального n получим четыре подпрямо неразложимых полукольца: $S_n = S_{n-1} \cup \{\infty\}$ и $T_n = T_{n-1} \cup \{0\}$, если n нечетно, и $S_n = S_{n-1} \cup \{0\}$ и $T_n = T_{n-1} \cup \{\infty\}$ при четном n .

Теорема 2. Для любого кардинала $m \geq 2$ существуют цепные подпрямо неразложимые полукольца мощности m как с 0 , так и с ∞ .

В самом деле, полученные цепные полукольца из примера 2 образуют цепочки вложенных полуколец

$$\begin{aligned} S_0 &\subset S_1 \subset \dots \subset S_n \subset \dots, \\ T_0 &\subset T_1 \subset \dots \subset T_n \subset \dots \end{aligned}$$

Их объединения $\bigcup_{k=0}^{\infty} S_k$ и $\bigcup_{k=0}^{\infty} T_k$ также будут цепными подпрямо неразложимыми полукольцами. Присоединяя к ним 0 или ∞ , снова получаем цепные подпрямо неразложимые полукольца.

Для любого ординала α по трансфинитной индукции построим цепные подпрямо неразложимые полукольца S_α и T_α .

В случае непердельного ординала α положим

$$S_\alpha = S_{\alpha-1} \cup \{\theta\},$$

где $\theta = \infty$, если $S_{\alpha-1}$ – полукольцо с нулем 0, и $\theta = 0$, если $S_{\alpha-1}$ содержит ∞ .

Если α – предельный ординал, то положим

$$S_\alpha = \left(\bigcup_{\beta < \alpha} S_\beta \right) \cup \{\theta\},$$

где $\theta = 0$ или $\theta = \infty$.

Аналогично строятся полукольца T_α .

Заметим, что для каждого бесконечного ординала α мы построили цепные подпрямо неразложимые полукольца, имеющие мощность $|\alpha|$.

В теореме 2 для каждого кардинала $m \geq 2$ получаются два подпрямо неразложимых идемпотентных цепных полукольца, что дает результат А. Романовской [4, corollary 2.9].

Предложение 3. Конечные цепные подпрямо неразложимые полукольца – с точностью до изоморфизма – это полукольца S_n и T_n из примера 2. Значит, для любого натурального числа $m \geq 2$ существует ровно 4 m -элементных цепных подпрямо неразложимых полукольца.

Замечание 2. Если подпрямо неразложимое коммутативное мультипликативно идемпотентное полукольцо S имеет поглощающий элемент ∞' , то, присоединяя к нему поглощающий элемент ∞ , по пункту 3 теоремы 1 получаем, что полукольцо $S \cup \{\infty\}$ подпрямо разложимо, так как задает конгруэнцию следующее разбиение $\tau_{\{\infty', \infty\}}$: одним ее классом будет $\{\infty', \infty\}$, остальные классы одноэлементны. При этом $S \cup \{\infty\} / \tau_{\{\infty', \infty\}} \cong S$.

Аналогично, для подпрямо неразложимого коммутативного мультипликативно идемпотентного полукольца S с нулем $0'$, полукольцо $S \cup \{0\}$ с присоединенным нулем 0 будет по теореме 1 подпрямо разложимым (разбиение $\tau_{\{0', 0\}}$ задает конгруэнцию, причем $S \cup \{0\} / \tau_{\{0', 0\}} \cong S$).

Из предложения 1 вытекает

Предложение 4. Для произвольной конгруэнции ρ на коммутативном мультипликативно идемпотентном полукольце S справедливы следующие утверждения:

1. ρ максимальная тогда и только тогда, когда она двухклассовая, то есть факторполукольцо S/ρ двухэлементно;
2. если $\rho \neq 1_S$, то ρ содержится в некоторой максимальной конгруэнции на S .

Замечание 3. Максимальные конгруэнции на некоммутативном мультипликативно идемпотентном полукольце могут иметь более двух классов. В самом деле, рассмотрим трехэлементное некоммутативное идемпотентное конгруэнц-простое полукольцо T из замечания 1.

Обозначим $T_1 = \{a, 1, b, c\}$ – четырехэлементное идемпотентное полукольцо с тождеством $xu = y$, такое, что $a \leq 1 \leq b \leq c$. Единственную нетривиальную конгруэнцию μ на T_1 задает трехклассовое разбиение $\{a\}, \{1\}, \{b, c\}$. Конгруэнция μ максимальна и $T_1/\mu \cong T$.

Пример 3. Рассмотрим конгруэнции на конечных цепных подпрямо неразложимых полукольцах S из примера 2. Возьмем произвольную ненулевую конгруэнцию ρ на S . Тогда $1\rho e$ по теореме 1. Пусть для различных элементов $a, b \in S \setminus \{1, e\}$, $a < b$ и $a\rho b$. Тогда $[a; b]_{\rho} \subseteq [a]_{\rho}$ по предложению 2. Найдутся $0', \infty' \in [a; b]_{\rho}$, такие, что $0'$ будет нулевым элементом полукольца $[0'; 1]_{\rho}$, а ∞' – поглощающий элемент в полукольце $[\infty'; 1]_{\rho}$. Поэтому имеем $0'\rho\infty'$, $(1+0')\rho(1+\infty')$, $(1+0')\rho(1+\infty')$, $1\rho\infty'$. По предложению 2 получаем $[a; 1]_{\rho} \subseteq [a]_{\rho}$. Таким образом, все конгруэнции на S имеют следующий вид: одним классом служит отрезок $[c; 1]_{\rho}$, где $c \in S$, остальные классы одноэлементны. Значит, $\text{Con } S \cong \langle S, < \rangle$ – $|S|$ -элементная цепь. Произвольная максимальная конгруэнция μ в этом случае имеет вид: одним ее классом будет промежуток $(m; 1]_{\rho}$, другим – m , где m – наименьший элемент цепи $\langle S, < \rangle$. Отметим, что двухэлементное факторполукольцо S/μ при этом идемпотентно, то есть изоморфно **B** или **D**.

Даже цепные подпрямо неразложимые полукольца не обязаны обладать максимальными конгруэнциями.

Замечание 4. Рассмотрим бесконечные цепные полукольца S_{α} , описанные в теореме 2. Конгруэнции на таких полукольцах устроены так же, как в примере 3: одним классом будет отрезок $[a; 1]_{\rho}$, где $a \in S$, остальные классы одноэлементны. Ясно, что полукольцо S_{α} не имеет максимальных конгруэнций тогда и только тогда α – предельный ординал. Кроме того $\text{Con } S_{\alpha} \cong \langle S_{\alpha}, > \rangle$.

Замечание 5. Отметим, что решетка конгруэнций любой дистрибутивной решетки является дистрибутивной. Однако решетка конгруэнций $\text{Con } S$ произвольного мультипликативно идемпотентного полукольца S не обязана быть модулярной. Например, рассмотрим булеан $B\{a, b\} = \{\emptyset, a, b, \{a, b\}\}$ как идемпотентное моно-полукольцо с одной операцией \cup . Легко проверить, что решетка $\text{Con } \langle B\{a, b\}, \cup, \cup \rangle$ содержит 7 элементов и не модулярна.

Идеалом полукольца S называется всякое его непустое подмножество I , такое, что для любых $a, b \in I, s \in S$ выполняется: $a + b, as, sa \in I$.

Для идеалов I, J обозначим идеал:

$$I + J = \{a + b : a \in I, b \in J\}.$$

Множество $\text{Id } S$ всех идеалов полукольца S относительно теоретико-множественного включения есть решетка с операциями

$$I \vee J = \sup(I, J) = I \cup J \cup (I + J), \quad I \wedge J = \inf(I, J) = I \cap J.$$

Предложение 5. Для любого мультипликативно идемпотентного полукольца S решетка $\text{Id } S$ всех его идеалов дистрибутивна.

Действительно, пусть $A, B, C \in \text{Id } S$. Положим $A + B = \{a + b : a \in A, b \in B\}$ и $AB = \{ab : a \in A, b \in B\}$. В решетке $\text{Id } S$ имеем $A \cap B = AB$ и $A \vee B = A + B$. Поэтому дистрибутивный закон принимает вид:

$$((A + B) \cup A \cup B)C = (AC + BC) \cup AC \cup BC.$$

Очевидно, что $AC \cup BC = (A \cup B)C$ и $(A + B)C = AC + BC$.

Обратно, возьмем $x \in AC + BC$. Тогда $x = ac + bd$ для подходящих элементов $a \in A, b \in B, c, d \in C$, откуда $ac \in A, bd \in B, x = ac + bd \in C$. Поэтому

$$x = x^2 = (ac + bd)x \in (A + B)C.$$

Работа выполнена в рамках государственного задания Минобрнауки РФ, № 1.1375.2014/ К.

Список литературы

1. Вечтомов Е. М., Петров А. А. Мультипликативно идемпотентные полукольца // *Фунд. и прикл. матем.* 2013. Т. 18. № 4. С. 41–70.
2. Вечтомов Е. М., Петров А. А. Полукольца с идемпотентным умножением. Киров: Изд-во ООО «Радуга-ПРЕСС», 2015. 144 с.
3. Monico C. On finite congruence-simple semirings // *Journal of Algebra*. 2004. V. 271. P. 846–854.
4. Romanowska A. On bisemilattices with one distributive law // *Algebra Universalis*. 1980. V. 10. P. 36–47.

УДК 512.55

Криптографические алгоритмы над абстрактными полукольцами

Чупраков Д.В.

Вятский государственный университет

chupdiv@yandex.ru

Аннотация

Чупраков Д.В. *Криптографические алгоритмы над абстрактными полукольцами.* В работе описаны криптоалгоритмы над абстрактными полукольцами, в основу которых положена сложность полиномиальной задачи разложения (PSD).

Введение. Работа носит обзорный характер и посвящена полукольцевым криптографическим алгоритмам опирающимся на сложность полиномиальной задачи разложения элемента b полукольца R в произведение $x^m a x^n$, где $a \in R$, x — элемент однопорождённой подалгебры, n, m — неотрицательные целые числа.

Под полукольцом $\langle R, +, \cdot \rangle$ понимается непустое множество с двумя ассоциативными бинарными операциями сложением (+) и умножением (\cdot) связанными дистрибутивными законами: $a(b+c) = ab+ac$ и $(a+b)c = ac+bc$ для любых элементов $a, b, c \in R$. Дополнительно будем полагать, что операция сложения коммутативна. Наличие нуля допускается, но не предполагается. Полукольцо называется коммутативным, если в нём коммутативна операция умножения. Если множество всех обратимых элементов полукольца R совпадает с $R \setminus \{0\}$, то R называется полукольцом с делением. Полукольцо с делением, не являющееся кольцом, называется полутелом. Коммутативное полутело называется полуполем.

Пионерскими трудами в области алгебраической криптографии являются статья [1] У. Диффи и М. Хеллмана, положившая начало криптографии с открытым ключом и работа [2] Р. Ривеста, А. Шамира и Л. Адлемана, в которой предложена практическая реализация криптосистемы с открытым ключом (RSA). Идея использования абстрактных алгебраических систем как источника криптографических преобразований связана с работой Т. ЭльГамала [3]. Предложенные им алгоритмы, описанные в терминах мультипликативной группы \mathbf{Z}_p^* классов вычетов по простому модулю p , не используют специфических свойств поля \mathbf{Z}_p и, следовательно, могут быть перенесены на другие группы. Так идеи алгоритмов Эль Гамала переносятся в работах Н. Коблица [4] и В. Миллера [5] на группы точек эллиптических кривых над конечным полем. Л. Аншель, М. Аншель и Г. Голдфилд в статье [6] предлагают использовать

некоммутативные группы кос Артина в качестве носителя криптопреобразований. Таким образом, криптография на группах оформилась в самостоятельное направление. Её естественным развитием стала кольцевая и полукольцевая криптография.

Первые эпизодические приложения полуколец в криптографии описаны в качестве примеров в монографии известной монографии Голана «Полукольца и их применения» [7], однако систематическое изучение криптопреобразований на полукольцах начато в статье [8] как развитие групповой криптографии. В указанных работах рассматриваются полугрупповые криптоалгоритмы на полумодулях над полукольцом.

Другим веткой развития алгебраической криптографии можно считать обобщение классических групповых алгоритмов на случай абстрактных колец подробно изложенных в монографии Ж. Као «Новые направления современной криптографии» [9]. Криптографические преобразования при этом выполняются в мультипликативной полугруппе кольца. Интерес представляет применение некоммутативных полуколец с делением в статьях [10], [11] и [12] в первой и второй работах предпринята попытка построения схемы электронной подписи. В третьей описан протокол распределения ключей Диффи–Хеллмана на полукольце с делением.

Развитием этого направления являются работы [13] и [14], выполненные под руководством Д.В. Чупракова.

Односторонние функции над некоммутативным полукольцом

Обозначим через \mathbf{N}_0 — неотрицательные целые числа. Степенью элемента a группы G называется отображение $\wedge: G \times \mathbf{N}_0 \rightarrow G$ по правилу $a^x = a \wedge x = a * \dots * a$, (x операндов). Задачей дискретного логарифмирования (DL) в группе G называется решение уравнения $a^x = b$ относительно $x \in \mathbf{N}_0$ при заданных $a, b \in G$.

Возьмём некоммутативное полукольцо $\langle R, +, \cdot \rangle$. Полагая $k \in \mathbf{N}_0$ и $r \in R$, введём обозначение $kr = r + \dots + r$ (k слагаемых).

При $k = 0$ будем считать $kr = 0$. Рассмотрим в полукольце R подалгебру $\langle r \rangle$, порождённую элементом $r \in R$. Элементы её можно представить в виде всевозможных многочленов $f(r) = a_n r^n + \dots + a_1 r + a_0$ с неотрицательными целыми коэффициентами a_0, \dots, a_n .

Теорема 1. *Подалгебра $\langle r \rangle$, порождённая элементом $r \in R$, коммутативна относительно умножения.*

Доказательство. Пусть $r \in R$ заметим, что для любых $a, b, n, k \in \mathbf{N}_0$

$$ar^n \cdot br^m = (r^n + \dots + r^n) \cdot (r^m + \dots + r^m) = r^{n+m} + \dots + r^{n+m} = (ab)r^{n+m}.$$

Значит, справедливо свойство коммутативности одночленов
 $ar^n \cdot br^m = (ab)r^{n+m} = br^m \cdot ar^n$.

Рассмотрим произвольные элементы $r_1, r_2 \in \langle r \rangle$. Найдутся такие многочлены

$$f_1(x) = a_n x^n + \dots + a_1 x + a_0 \in \mathbf{N}_0[x] \quad f_2(x) = b_m x^m + \dots + b_1 x + b_0 \in \mathbf{N}_0[x],$$

что $r_1 = f_1(r), r_2 = f_2(r)$. Применяя доказанное свойство коммутативности одночленов, получаем

$$r_1 r_2 = f_1(r) f_2(r) = \sum_{i=0}^n \sum_{j=0}^m (a_i b_j) x^{i+j} = r_2 r_1.$$

Итак, подалгебра $\langle r \rangle$ коммутативна. Теорема доказана.

Отметим, что в общем случае $f_1(r) \cdot f_2(s) \neq f_2(s) \cdot f_1(r)$ при $r \neq s$.

Возьмём теперь мультипликативную полугруппу $\langle R, \cdot \rangle$ полукольца R . Над ней, имеют место следующие задачи:

Задачей симметричного разложения (*SDP*) называется задача нахождения какого-либо элемента $x \in R$, для которого выполняется равенство $b = x^m a x^n$ при заданных $a, b \in R$ и $m, n \in \mathbf{N}_0$.

Обобщенной задачей симметричного разложения (*GSDP*) называется *SDP*-задача в которой известно, что элемент x принадлежит подмножеству $S \subseteq R$. При $S = \langle p \rangle$ частная *GSDP*-задача называется полиномиальной задачи разложения (*PSD*).

Отметим, что в монографии [3] описанные задачи, определённые в абстрактных группах, использованы для построения криптосистем над абстрактным кольцом, а в статьях [4, 5] применяются в криптосистемах над полукольцами делением.

Для каждого фиксированных неотрицательных чисел m, n и каждого $p \in R$ определим отображение $e_{n,m,p} : R \times \langle p \rangle \rightarrow R$ правилом

$$e_{n,m,p}(a, x) = x^m \cdot a \cdot x^n.$$

Его можно воспринимать как операцию возведения в степень $a^x = e_{n,m,p}(a, x)$ в полукольце $\langle R, +, \cdot \rangle$. При этом соответствующая *PSD*-задача становится аналогом задачи дискретного логарифмирования в группе классов вычетов по простому модулю.

Криптографические алгоритмы над абстрактным полукольцом

I. Протокол выработки секретного ключа Диффи—Хеллмана над произвольным полукольцом, предложенный для полукольца с делением в работе [11].

Стороны открытым образом выбирают ненулевые элементы p, q из полукольца R и натуральные числа n и m .

Сторона A определяет ненулевой секретный элемент $x_A = f_A(p) \in \langle p \rangle$, где $f_A(x) \in \mathbf{N}_0[x]$ — некоторый (секретный) многочлен, и передаёт стороне B значение

$$y_A = e_{n,m,p}(q, x_A) = x_A^n q x_A^m.$$

Аналогично сторона B выбирает элемент $x_B = f_B(p) \in \langle p \rangle$, где $f_B(x) \in \mathbf{N}_0[x]$, и сообщает стороне A значение $y_B = e_{n,m,p}(q, x_B) = x_B^n q x_B^m$.

Теперь стороны A и B вычисляют ключи по формулам

$$k_A = e_{n,m,p}(y_B, x_A) = x_A^n y_B x_A^m,$$

$$k_B = e_{n,m,p}(y_A, x_B) = x_B^n y_A x_B^m.$$

По теореме 1 $x_A x_B = x_B x_A$ и, следовательно, $k = k_A = k_B$ общий секретный ключ.

Данный протокол можно использовать для n участников шифрованного обмена информацией [11].

Заметим, что на полукольце R определяется полиномиальная задача Диффи—Хеллмана (PDH) как задача вычисления $a^{uv} = e_{n,m,p}(a, uv)$ по известным $a^u, a^v, p \in R, u, v \in \langle p \rangle$.

II. Алгоритм Эль-Гамала на абстрактном полукольце, описанный в статье [12]. Для колец аналогичный алгоритм приведён в [9].

Существенным препятствием к построению криптографических функций над полукольцами служит необратимость операций, что приводит к невозможности построения обратимой функции. Указанное затруднение может быть преодолено различными способами. Так при построении алгоритма Эль-Гамала рассматривается источник открытых сообщений M , порождающий двоичные последовательности длины s и шифрование выполняется в аддитивной группе $\langle \mathbf{Z}_2^s, \oplus \rangle$ с покомпонентным сложением \oplus . Полукольцо R , в свою очередь, служит источником псевдослучайных аргументов хеш-функции $H : R \rightarrow \mathbf{Z}_2^s$.

Итак, алгоритмом Эль-Гамала на полукольце R с параметрами $m, n \in \mathbf{N}_0$ и криптографической хеш-функцией $H : R \rightarrow \mathbf{Z}_2^s$ состоит из следующих

преобразований:

1. Формирование ключей. Пусть p, q — случайно выбранные пользователем элементы полукольца $\langle R, +, \cdot \rangle$, а $f(x) \in \mathbf{N}_0[x]$ — многочлен, выбираемый пользователем, так, что $f(p) \neq 0$. Обозначим $f_p = f(p) \in \langle p \rangle$. Секретным ключом криптоалгоритма будет упорядоченная тройка $k_{priv} = (p, q, f_p)$, а открытым ключом — $k_{pub} = (p, q, y)$, где $y = f_p^m q f_p^n$. Многочлен f после формирования ключей уничтожается.

2. Шифрование сообщения $M \in \mathbf{Z}_2^s$ выполняется функцией $E_{k_{pub,h}} : \mathbf{Z}_2^s \rightarrow G \times \mathbf{Z}_2^s$, заданной правилом $E_{k_{pub,h}}(M) = (a, b)$, где

$$a = h^m q h^n \in R \text{ и } b = H(h^m y h^n) \oplus M \in \mathbf{Z}_2^s$$

зависят от сессионного ключа $h = h(p) \in \langle p \rangle$, заданного случайно сгенерированным многочленом $h(x) \in \mathbf{N}_0[x]$, таким, что $h(p) \neq 0$.

3. Расшифрование осуществляется функцией $D_{k_{priv}} : G \times \mathbf{Z}_2^s \rightarrow \mathbf{Z}_2^s$ по правилу

$$D_{k_{priv}}(a, b) = H(f_p^m a f_p^n) \oplus b.$$

Теорема 2. Для исходного сообщения $M \in \mathbf{Z}_2^s$, равномерного распределённого по всему пространству \mathbf{Z}_2^s , алгоритм Эль-Гамала на полукольце R является стойким к атаке на основе подобранным открытого текста по принципу «всё или ничего» в том и только том случае, когда задача PDH является вычислительно сложной.

Принцип «всё или ничего» означает, что в результате атаки злоумышленник либо не получает никакой информации, либо находит всю интересующую его информацию. Принцип «всё или ничего» устанавливает минимальный уровень криптостойкости алгоритма.

Доказательство. Необходимость. Пусть известно, что пара $(a, b) \in R \times \mathbf{Z}_2^s$ получена с помощью рассмотренного алгоритма с параметрами $m, n \in \mathbf{N}_0$, криптографической хеш-функцией $H : R \rightarrow \mathbf{Z}_2^s$ и открытым ключом $k_{pub} = (p, q, y)$. Допустим существует оракул, который вычисляет $e_{n,m,p}(a, uv)$ по известным $e_{n,m,p}(a, u)$, $a^v = e_{n,m,p}(a, v)$, $p \in R$ и $u, v \in \langle p \rangle$. Иными словами PDH-задача в полукольце R имеет эффективный алгоритм решения. По известным значениям $f_p, h \in \langle p \rangle$, $m, n \in \mathbf{N}_0$, $a = e_{n,m,p}(q, h)$ и $y = e_{n,m,p}(q, f_p)$ оракул, в силу теоремы 1, вернёт

$$e = e_{n,m,p}(q, f_p h) = h^m f_p^m \cdot q \cdot f_p^n h^n = h^m y h^n.$$

Теперь исходный текст M может быть восстановлен следующим образом:

$$H(e) \oplus b = H(h^m y h^n) \oplus H(h^m y h^n) \oplus M = M.$$

Достаточность. Допустим, что существует оракул, позволяющий с достаточно большой вероятностью ε восстанавливать сообщение M по его криптотексту $(a, b) \in R \times \mathbf{Z}_2^s$, зная только открытый ключ $k_{pub} = (p, q, y)$ и параметры алгоритма $m, n \in \mathbf{N}_0$, $H: R \rightarrow \mathbf{Z}_2^s$.

Рассмотрим PDH-задачу задача вычисления a^{uv} по известным $a^u, a^v, p \in R$, $u, v \in \langle p \rangle$ с операцией возведения в степень, заданной равенством $a^x = e_{n,m,p}(a, x)$. Примем в качестве открытого ключа $k_{pub} = (p, a, a^u)$ и в качестве шифротекста $E(M) = (a^v, b)$, где b — случайный элемент \mathbf{Z}_2^s .

Оракул вернёт значение M , соответствующее паре (a^v, b) . Тогда, учитывая коммутативность подалгебры $\langle p \rangle$ (теорема 1), получим хэш-значение

$$h = H(a^{uv}) = H(u^m v^n a v^n u^m) = b \oplus M.$$

Предположим, оракул угадывает значение $h = H(a^{uv})$ с указанной достаточно большой вероятностью ε .

Рассмотрим следующую модель указанного оракула. Пусть есть список L пар $(r_i, h_i) \in R \times \mathbf{Z}_2^s$, изначально пустой. Когда оракул возвращает значение h происходит проверка наличия в L пары (r, h) . Если она найдена, то возвращается значение оракул возвращает $M = h \oplus b$, иначе в список добавляется случайная пара $(r, h) \in R \times \mathbf{Z}_2^s$, где r выбирается случайным образом из полукольца R , а h вычисляется как $h = H(r)$. Оракул при этом возвращает $M = h \oplus b$. Так как по условию с достаточно большой вероятностью ε оракул угадывает значение M , то с той же вероятностью он будет правильно выбирать прообраз r значения хэш-функции H .

Итак, для $H(a^{uv}) = h_i$ взяв $a^{uv} = r_i$ получим решение PDH-задача с вероятностью ε . Что завершает доказательство теоремы.

III. Симметричный поточный алгоритм шифрования над полукольцом с делением R , описанный в работе [13].

Рассмотрим иной подход к решению проблемы построения обратимых функций. Положим, что R является полукольцом с делением. Тогда функция $e_{n,m,p}(x, k) = k^m \cdot a \cdot k^n$, где n, m — натуральные числа, инъективна при любом ненулевом значении x из полукольца R и, значит, является криптофункцией.

Представим исходный текст в виде вектора $X = (x_1, \dots, x_s)$ и рассмотрим криптофункцию

$$E_k(x_i) = e_{ni,mi,p}(H(x_i), k) = k^{mi} H(x_i) k^{-ni},$$

где x_i – i -й блок исходного текста, $H: \mathbf{N} \rightarrow R$ инъективная функция, сопоставляющая блоку исходного текста элемент полукольца R такая, что прообразом 1 полукольца R служит пустое множество, $k \in R$ – ключ шифрования, n и m – натуральные числа, известные обеим сторонам шифрованного обмена информацией и являющиеся несекретными параметрами.

В полукольце с делением любой ненулевой элемент имеет обратный, следовательно функция расшифрования может быть записана в виде

$$D_k(y_i) = H^{-1}(k^{-ni} y_i k^{-mi}),$$

где $H^{-1}: R \rightarrow \mathbf{N}$ – функция обратная к функции H .

Отметим, что восстановление ключа k по известному набору пар (x_i, y_i) является SDP-задачей.

Об электронной подписи. Естественно поставить задачу о реализации электронной подписи над абстрактным полукольцом или полукольцом с делением. Попытки её решения предприняты в статьях [10, 11]. Однако, в первой статье подпись может быть скомпрометирована простой заменой подписываемого текста, т.к. верифицирующая функция зависит только от элементов подписи и открытого ключа и не зависит от подписываемого текста. Во второй статье, подпись сообщения зависит от секретного корреспондента и открытого ключа получателя, причём указанные ключи выбираются из общей однопорождённой подалгебры. Поэтому каждому корреспонденту должна соответствовать своя пара открытого и секретного ключей у получателя сообщения. Что ограничивает применение предлагаемого алгоритма. Таким образом, задача построения алгоритма электронной подписи остаётся открытой.

Список литературы

1. Diffie W., Hellman M.E. New Directions in Cryptography, IEEE Transactions on Information Theory, vol. IT-22, 1976. — p: 644-654.
2. Rivest R. L., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Communications of the ACM, T. 21, № 2, 1978. — p: 120-126.
3. ElGamal T. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms // IEEE Transactions on Information Theory. 31 (4). 1985. — p. 469-472.
4. Koblitz N. Elliptic curve cryptosystems // Mathematics of Computation, 1987, Vol. 48 (177). — p. 203-209
5. Miller V. Use of elliptic curves in cryptography // CRYPTO. Lecture Notes

in Computer Science, 1985, Vol. 85. — p. 417-426.

6. Anshel I., Anshel M., Goldfeld D. An algebraic method for public-key cryptography // Math. Res. Lett, vol. 6, 1999 —p. 287-291.

7. Golan J.S. Semirings and their applications, 1999. 381 p.

8. Sakalauskas E. One Digital Signature Scheme in Semimodule over Semiring // INFORMATICA, Vol. 16, No. 3, 2005. — p. 383-394.

9. Cao Z. New Directions of Modern Cryptography. London: CRC Press, 2012.

10. Anjaneyulu G.S. G.N., Vasudeva Reddy P., Reddy U.M. Secured Digital Signature Scheme using Polynomials over Non-Commutative Division Semirings // International Journal of Computer Science and Network Solutions (IJCSNS). Vol. 8. No. 8. 2008. — p. 278-283.

11. Anjaneyulu G.S. G.N., Reddy U.M Secured Directed Digital Signature over Non-Commutative Division Semirings and Allocation of Experimental Registration Number // IJCSI International Journal of Computer Science Issues, Vol. 9. Issue 5. No. 3. 2012. — p. 376-386.

12. Anjaneyulu G.S. G.N. Sanyasirao A. Distributed Group Key Management Protocol over Non-commutative Division Semirings // Indian Journal of Science and Technology. Vol 7. 2014. — p. 871-876.

13. Погудина А.А. Симметричная криптосистема над полукольцом с делением. // Информационные технологии и прикладная математика. Межвузовский сборник аспирантских и студенческих работ, Вып. 5. 2015. — С. 130-133.

14. Погудина Е.Н. Чупраков Д.В. Криптоалгоритм Эль Гамала над абстрактным полукольцом. // Математический вестник педвузов и университетов Волго-Вятского региона, Вып. 18, 2016. — С. 375-382.

Секция № 2
Программная инженерия



УДК 004.056.5

Программная система встраивания информации в изображения с использованием стеганографических и криптографических алгоритмов

Егоров А.А., Чернышова А.В.

Донецкий национальный технический университет
yegorov0725@yandex.ru, alla@pmi.dgtu.donetsk.ua

Аннотация

Егоров А.А., Чернышова А.В. Программная система встраивания информации в изображения с использованием стеганографических и криптографических алгоритмов. В статье описаны основные структуры и алгоритмы разработанной программной системы защиты информации с использованием стеганографических и криптографических алгоритмов. Приведен авторский алгоритм внедрения и извлечения данных при работе с изображением. Проанализированы характеристики популярных существующих программных средств.

Введение. Цель работы: исследование стеганографических и криптографических алгоритмов защиты информации, последующее проектирование и разработка программной системы встраивания информации в изображения с использованием стеганографических и криптографических алгоритмов.

Актуальность работы: защита авторского права в настоящий момент играет большую роль при рассмотрении вопросов информационной безопасности. Механизм цифровых водяных знаков популярен как для защиты различных изображений (фото, логотипы), так и для защиты видео-файлов и аудио-файлов. Предлагаемая программная система может использоваться как средство для защиты авторского права пользователя при работе с файлами в формате .bmp и .png. Предложенный программный продукт можно адаптировать для защиты авторского права в видео-файлах после создания дополнительного модуля работы с форматом .avi [1].

При использовании стеганографических алгоритмов передаваемая информация скрывается в файле. Данные, внутри которых размещаются скрытые сообщения, называются стегоконтейнером, а сами скрытые сообщения – стегосообщениями. Канал передачи стегоконтейнера называется стеганографическим каналом или стегоканалом. К стеганографии относятся множества различных методов: условные расположения знаков, микрофотоснимки, средства связи на плавающих частотах и тайные каналы. Стеганография является отличным дополнением, но никак не заменой криптографии [2].

Использование авторского алгоритма вместе с существующими криптографическими алгоритмами шифрования позволит более надежно защитить встраиваемую в графические изображения информацию и дальнейшую ее передачу по каналам связи.

Анализ существующих систем. На сегодняшний день существует огромное количество открыто распространяемых программных продуктов, использующих стеганографическую технику. Наиболее популярны инструменты размещаются в Github-репозиториях.

В рамках проекта DarkJPEG разработан стеганографический веб-сервис нового поколения, позволяющий скрывать конфиденциальную информацию в виде незаметного шума в JPEG-изображениях, при этом выделить данную информацию можно только зная заданный при кодировании секретный ключ-пароль. Сервис использует стойкие методы стеганографии для сокрытия самого факта сокрытия информации вместе со стойкими методами криптографии для защиты данных, передаваемых по открытым каналам [3].

Основные особенности:

- использование SHA-3 для генерации ключей;
- симметричное шифрование AES-256;
- JPEG (DCT LSB) стеганография;
- поддержка RarJPEG и двойного сокрытия;
- подбор случайного контейнера;
- вычисления без участия сервера (на клиентской стороне);
- гарантия полной конфиденциальности.[3]

OpenStego – это стеганографическое приложение, которое предоставляет две возможности:

1. Сокрытие данных (можно скрыть любые данные внутри файла-контейнера).

2. Водяные знаки (встраивание невидимой подписи, например в изображениях, с целью обнаружения несанкционированного копирования).

Данная программа предоставляет возможность надежного встраивания водяных знаков при незначительном изменении размера или кадрирования изображения [4]. Отличительной особенностью OpenStego является то, что можно расширить ее возможности с помощью своих плагинов, написанных на Java.

LSB-Steganography – стеганографическая программа для сокрытия данных в изображениях, реализована в виде скрипта на языке Python, использующая методы наименьшего значащего бита. Модуль LSBSteg основан на OpenCV (библиотека для работы с изображениями). Он использует незначительный бит каждого пикселя и каждого цвета изображения для сокрытия данных [5]. Программа не выполняет никаких дополнительных криптографических преобразований для защиты внедренных данных.

Описание структуры разрабатываемой системы. При разработке системы прежде всего необходимо определить требования к программе. На рисунке 1 показана диаграмма использования разрабатываемой системы.

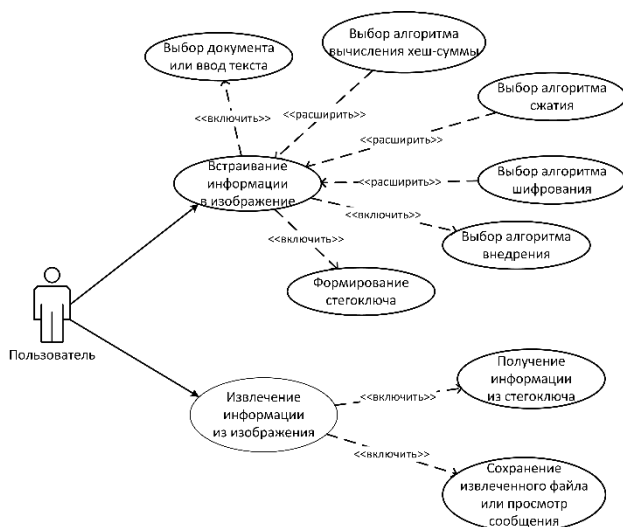


Рис. 1 – Диаграмма вариантов использования (Uses Case) разрабатываемой системы

При проектировании авторской стегосистемы можно выделить три этапа:

- получение из цифрового документа байтов;
- преобразование байтов цифрового документа (подготовка данных для внедрения в контейнер);
- внедрение данных в изображение, используя метод LSB [6].

На первом этапе некоторый цифровой объект должен быть представлен в виде потока байт. Если это файл, то его можно открыть в бинарном режиме. Если текстовое сообщение, то представить его в виде байтов, заданной кодировкой [7].

На втором этапе, на вход подаются байты, полученные на первом этапе. Прежде всего необходимо вычислить контрольную хеш-сумму [8] исходных байт. После этого сжать исходные байты, выбранным пользователем алгоритмом сжатия [9]. К сжатым байтам в начало добавляется заголовок (метаданные). После чего эти байты шифруются симметричным алгоритмом шифрования [10].

Последний этап состоит в том, что зашифрованные байты внедряются в изображение.

На рисунке 2 схематически показаны три этапа работы программной системы.

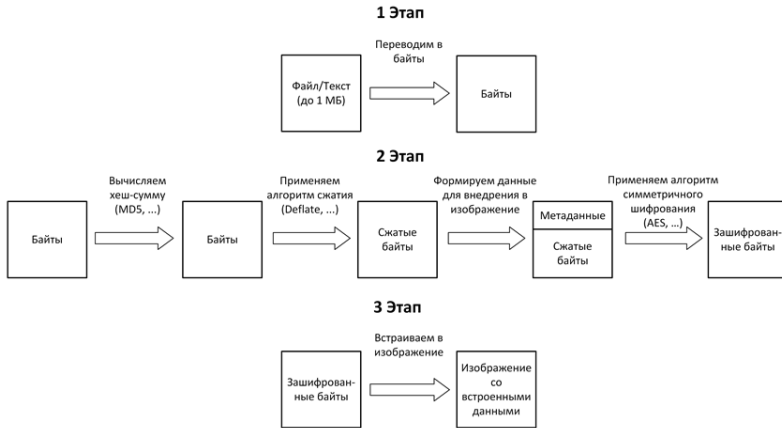


Рис. 2 – Этапы работы программной системы

Для представления пользовательских данных в виде байтов был разработан интерфейс `IData`, реализующие этот интерфейс классы должны иметь следующие методы:

- `byte[] GetBytes();`
- `int GetSize();`
- `Stream GetStream();`
- `IHeader GetHeader();`

В данной системе уже есть два класса, реализующих данный интерфейс: `File`, `Text`.

Метаданные – это данные описывающие пользовательские данные. Для представления метаданных был создан интерфейс `IHeader`. Аналогично реализованы два класса: `FileHeader` и `TextHeader`.

UML-диаграмма классов представлена на рисунке 3. Размеры структур и описание полей `FileHeader` указаны в таблице 1.

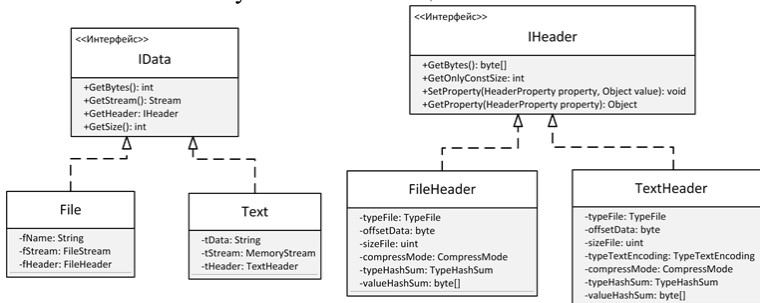


Рис. 3 – Диаграмма классов, представляющая первый уровень системы

Таблица 1. Структура класса FileHeader

Имя поля	Кол-во байт	Описание
typeFile	1	Тип внедряемого документа
offsetData	1	Смещение начала данных (размер заголовка)
sizeFile	4	Размер файла (кол-во байт)
compressMode	1	Режим сжатия (GZip, Deflate)
typeHashSum	1	Тип хеш-суммы (MD5, SHA-1, ...)
valueHashSum	n	Значение хеш-суммы (размер определяется по типу хеш-суммы)

Класс TextHeader имеет такие же поля как FileHeader и дополнительное поле typeTextEncoding, представляющее кодировку внедряемого текста. Данное поле занимает 1 байт (т. е. в системе может быть до 256 различных кодировок текста). В данной системе текст можно представить такими кодировками: ASCII, UTF-8, WINDOWS-1251, CP866.

После представления данных в виде байтов, они проходят ряд преобразований (конвейер трансформации – pipeline transform). Сначала вычисляется контрольная хеш-сумма пользовательских данных, после этого байты сжимаются и шифруются. UML-диаграмма классов второго этапа показана на рисунке 4.

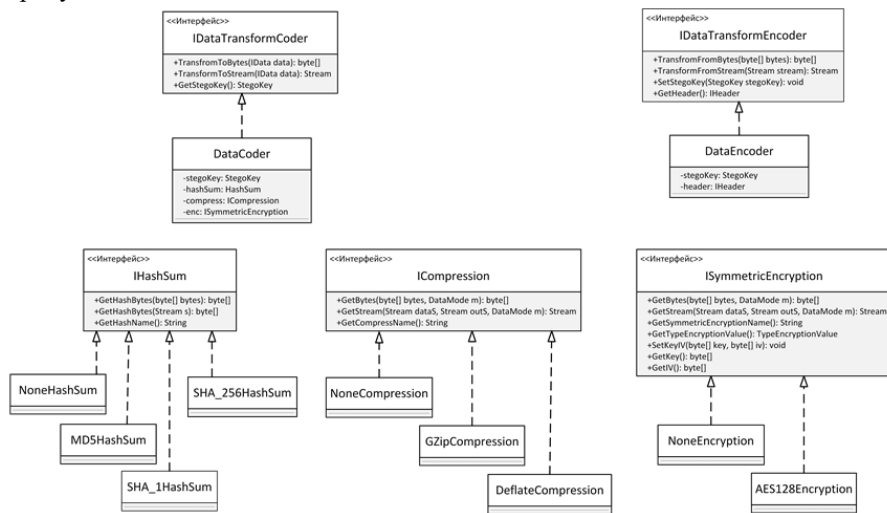


Рис. 4 – Диаграмма классов второго этапа

Последний этап системы – это внедрение зашифрованных байтов в изображение.

В данной системе создано два интерфейса `IStegoTransformEmbed` (для внедрения данных в изображение) и `IStegoTransformExtract` (для извлечения данных из изображения).

Интерфейс `IStegoTransformEmbed` имеет два метода:

- `EmbedFromBytes` (внедрение из байтов);
- `EmbedFromStream` (внедрение из потока);

Метод `EmbedFromBytes` принимает следующие аргументы:

- `int offset` – первое смещение начала чтения первого пикселя;
- `int maxDiffusion` – максимальное значение рассеивания (до 250);
- `byte[] bytes` – данные для внедрения;
- `Bitmap container` – изображение, в которое будут внедряться данные;
- `StegoKey key` – стегоключ (см таб. 2).

UML-диаграмма классов заключительного этапа показана на рисунке 5.

На рисунке 6 показана схема внедрения данных в изображение. На рисунках 7-8 показаны алгоритмы внедрения/извлечения данных.

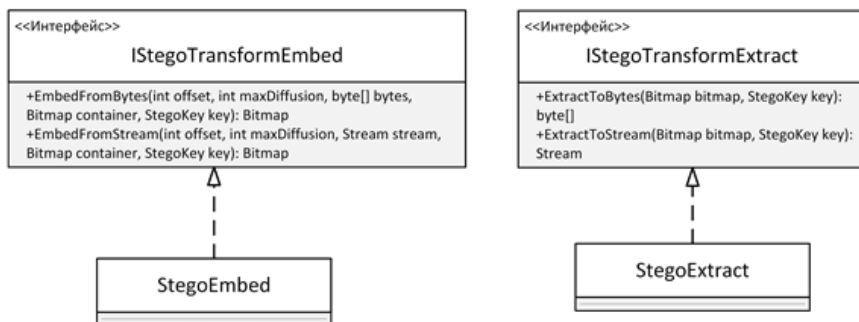


Рис. 5 – Диаграмма классов заключительного этапа

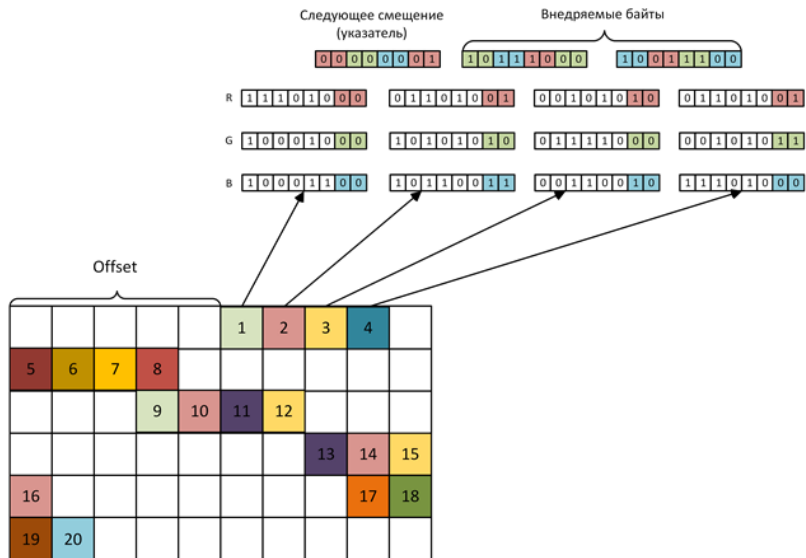


Рис. 6 – Схема внедрения данных (9-10 байт) в растровое изображения используя метод LSB.

Алгоритм Встраивания информации

Вход: offset – первое смещение начала чтения первого пикселя
 maxDiffusion – максимальное значение рассеивания
 bytes – данные для внедрения
 container – растровое изображение для внедрения данных
 key – стегоключ

Выход: result – изображение со встроенными данными

Перевести изображение в поток пикселей (по строкам).
 Заполнить стегоключ.

Пропустить заданное количество offset пикселей.

Если кол-во внедряемых байт (реальныйРазмер) – четное

То размерБайтов = реальныйРазмер – 2

Иначе размерБайтов = реальныйРазмер – 1

Пока (текущаяПара < размерБайтов)

 Считать из потока 4 пикселя.

 Сгенерировать случайное число – указатель на

 следующие 4 пикселя.

 Заменить последние два бита 4-х пикселей на указатель и

 два байта пользовательских данных.

 Увеличить текущаяПара на два.

Если кол-во внедряемых байт (реальныйРазмер) – четное

То Заменить последние два бита 4-х пикселей на 254

и два байта заключающих данных.

Иначе (кол-во внедряемых байт - нечетное)

 Заменить последние два бита 4-х пикселей на 255

 и один байт заключающих данных.

Сформировать изображение из потока пикселей.

Вернуть сформированное изображение.

Рис. 7 – Алгоритм встраивания информации

Алгоритм Извлечения информации

Вход: container – изображение с вложенными данными
key – стегоключ
Выход: байты пользовательских данных

Перевести изображение в поток пикселей (по строкам).
Считать информацию из стегоключа.
Пропустить заданное количество offset пикселей.
пока (текущаяПозицияПотока < длинаПотока - 4)
Считать из потока 4 пикселя.
Преобразовать последние 2 бита считанных пикселей
в первый байт – указатель и два пользовательских данных.
Если указатель равен 255 или 254,
То Выйти из цикла
Записать два байта в поток.
Если указатель равен 255
То Записать один байт в поток.
Иначе Если указатель равен 254
То Записать два байта в поток.
Вернуть байты из потока.

Рис. 8 – Алгоритм извлечения информации

Таблица 2. Структура класса StegoKey

Имя поля	Кол-во байт	Описание
Offset	4	Первое смещение начала чтения (кол-во пикселей от начала)
EmbeddedMode	1	Режим встраивания данных в изображение
TypeEncryption	1	Тип симметричного алгоритма шифрования
SizeEncryption-IV	2	Размер вектора инициализации
SizeEncryption-Key	2	Размер ключа шифрования
IV	Указано в SizeEncryptionIV	Вектор инициализации
Key	Указано в SizeEncryptionKey	Ключ шифрования

Из преимуществ стоит отметить, что данная система является расширяемой, например, можно добавить свой алгоритм сжатия или шифрования. Из недостатков – система не является быстродействующей.

Выводы. В статье проведен анализ существующих стегосистем, представлена структура разработанной программной системы, диаграммы классов, некоторые алгоритмы работы программной системы.

Система предоставляет возможность выбирать доступные алгоритмы обработки, которые затем будут записаны в стегоключ. Программную систему можно использовать для различных пользователей. При этом для каждого пользователя настройки программы будут собственными (используемый алгоритм сжатия, используемая кодировка, симметричный алгоритм

шифрования, однонаправленная хеш-функция), каждый будет работать со своим стегоключем.

Список литературы

1. Использование стеганографических и криптографических средств для защиты видеофайлов. А.В. Чернышова, Б.С. Маркин, Наукові праці ДонНТУ Серія “Інформатика, кібернетика та обчислювальна техніка”, №1(19), 2014, стр.46-49
2. Стеганография – актуальность в XXI веке // SeoCyber [Электронный ресурс]. – Режим доступа: <http://seocyber.net/steganografiya-aktualnost-v-xxi-veke>
3. DarkJPEG: стеганография для всех // Хабрахабр. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/187402/>
4. OpenStego // Github [Электронный ресурс]. – Режим доступа: <https://github.com/syvoidya/openstego>
5. LSB-Steganography // Github [Электронный ресурс]. – Режим доступа: <https://github.com/RobinDavid/LSB-Steganography>
6. LSB стеганография // Хабрахабр. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/112976/>
7. Краткая история кодировок от ASCII до UTF-8 // Popel-studio [Электронный ресурс]. – Режим доступа: <http://popel-studio.com/blog/article/kratkaya-istoriya-kodirovok-ot-ascii-do-utf-8.html>
8. Контрольные суммы MD5 и SHA. // Desktoplinux [Электронный ресурс]. – Режим доступа: http://desktoplinux.ru/unix_guide/proverka_khesh_summy_md5_v_linux_md5sum
9. Алгоритмы сжатия данных без потерь // Хабрахабр [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/231177/>
10. Обзор алгоритмов Шифрования // Rohos [Электронный ресурс]. – Режим доступа: http://www.rohos.com/help/crypto_algorithms.htm

УДК 004.056.5

Подсистема защиты авторского права в программном обеспечении

Чернышова А.В., Кубашевский Д.В.

Донецкий национальный технический университет
alla@pmi.dgtu.donetsk.ua, dehax12@gmail.com

Аннотация

Чернышова А.В., Кубашевский Д.В. Подсистема защиты авторского права в программном обеспечении. В данной работе выполнен анализ существующих механизмов защиты авторского права на программное обеспечение и предложено описание работы программной подсистемы защиты ПО с использованием криптографических алгоритмов шифрования.

Введение. Ежедневно продаётся огромное количество копий программного обеспечения. Каждая такая копия обычно включает в себя систему защиты от копирования и проверки лицензии на приобретённое ПО. Однако зачастую подобные системы защиты ненадёжны, так как их разработке уделяется значительно меньше времени и внимания, чем оригинальному продукту — объекту защиты. Поэтому программное обеспечение, которое имеет такую слабую защиту, обычно подвергается "взлому" "крэкерами". Далее взломанные копии попадают в открытый доступ, где обычные пользователи бесплатно скачивают полностью работоспособное ПО. Таким образом, разработчики ПО несут огромные финансовые потери.

Лицензия на программное обеспечение — это правовой инструмент, определяющий использование и распространение программного обеспечения, защищённого авторским правом. Обычно лицензия на программное обеспечение разрешает получателю использовать одну или несколько копий программы, причём без лицензии такое использование рассматривалось бы в рамках закона как нарушение авторских прав издателя.[1]

На рынке программного обеспечения в настоящий момент активно используется программное обеспечение с открытым кодом, но также большое количество коммерческих программных продуктов.

Сегодня используются такие виды лицензий на программное обеспечение: открытая лицензия (Open Source), бесплатная лицензия (Freeware, GPL, Adware, Postcardware, Donationware, Nagware, begware) условно-бесплатное программное обеспечение (ShareWare, Trial, trialware, Demo, demoware), коммерческое программное обеспечение (Commercial). [1]

Цель работы — описать подсистему защиты авторского права в программном обеспечении с использованием криптографических алгоритмов шифрования для возможного использования при лицензировании условно-бесплатного или коммерческого программного обеспечения.

Обзор существующих механизмов защиты авторского права на программное обеспечение

Для защиты ПО используется ряд методов, таких как:

- алгоритмы запутывания - используются хаотические переходы в разные части кода, внедрение ложных процедур - "пустышек", холостые циклы, искажение количества реальных параметров процедур ПО, разброс участков кода по разным областям ОЗУ и т.п.;
- алгоритмы мутации - создаются таблицы соответствия операндов - синонимов и замена их друг на друга при каждом запуске программы по определённой схеме или случайным образом, случайные изменения структуры программы;

- алгоритмы компрессии данных - программа упаковывается, а затем распаковывается по мере выполнения;

- алгоритмы шифрования данных - программа шифруется, а затем расшифровывается по мере выполнения;
- вычисление сложных математических выражений в процессе отработки механизма защиты - элементы логики защиты зависят от результата вычисления значения какой-либо формулы или группы формул;
- методы затруднения дизассемблирования - используются различные приёмы, направленные на предотвращение дизассемблирования в пакетном режиме;
- методы затруднения отладки - используются различные приёмы, направленные на усложнение отладки программы;
- эмуляция процессоров и операционных систем - создаётся виртуальный процессор и/или операционная система;
- нестандартные методы работы с аппаратным обеспечением - модули системы защиты обращаются к аппаратуре. [2]

Практически для каждого популярного ПО можно найти в интернете взломанную рабочую копию. В основном, защитные механизмы делятся на две основные категории: основанные на знании пароля, серийного номера и основанные на владении носителем информации с ключом, файлов программы.

Защита, основанная на знании, является наиболее уязвимой, так как легальные пользователи могут сообщить всем серийный номер, с помощью которого другие пользователи могут запустить программу нелегально. В случае если серийный номер уникально генерируется для каждого пользователя и является функцией его имени, активировать программу одинаковым серийным номером не получится. В этом случае придётся проанализировать алгоритм генерации серийного номера в зависимости от имени пользователя либо избавиться от алгоритма защиты, "пропатчив" исполнимый код программы, то есть, изменив в бинарном коде программы условие, при котором система защиты посчитает ПО активированным.

Более стойкими являются системы защиты второй категории, основанные на владении некоторым носителем, который практически невозможно воссоздать самостоятельно. Но огромным минусом такой защиты является тот факт, что ключевой носитель может быть потерян или испорчен. Тогда пользователю придётся пройти процедуру восстановления такого носителя при условии, если разработчик её вообще поддерживает. [3]

Алгоритм работы подсистемы защиты авторского права программного обеспечения

Предположим, имеется разработанное программное обеспечение, которое нужно защитить от незаконного копирования и дальнейшего использования.

При выборе алгоритма работы подсистемы защиты авторского права программного обеспечения будем считать, что программное обеспечение условно-бесплатное или коммерческое.

Основная идея подсистемы защиты авторского права ПО состоит в том, что существует некоторый файл, без которого программа не способна выполнить какие-либо важные для пользователя функции (набор функций программы ограничен). Для доступа ко всему набору функций ПО, необходимо получить доступ к дополнительному файлу. Получить расширенные возможности по работе с программой пользователь может лишь предварительно подтвердив свои лицензионные данные — права на использование данного ПО. Таким образом, невозможно будет использовать программу без покупки лицензии на неё. [4]

Пусть защищаемое ПО состоит из запускаемого исполнимого модуля *.exe и вспомогательного модуля *.dll [5], благодаря которому будет возможно работать с полной версией программы, доступной в платной версии. Покупая программу, пользователь получает установочный дистрибутив, который содержит только лишь запускаемый модуль *.exe.

После установки программы, система защиты запрашивает данные о лицензии, например, персональные данные пользователя и соответствующий ключ, которые данный пользователь получил при покупке ПО.

На рисунке 1 изображён процесс получения вспомогательного модуля программы.

Алгоритм получения пользователем вспомогательного модуля программы будет следующим. При установке программного обеспечения и вводе пользователем данных (ФИО, электронный адрес, номер телефона), персональные данные пользователя и некоторая информация об используемом аппаратном обеспечении компьютера (серийный номер процессора, жёсткого диска, MAC-адрес сетевого адаптера), а также ключ, который пользователь получил при покупке, передаётся на сервер фирмы-разработчика устанавливаемого программного обеспечения с использованием защищённого канала связи (использование защищённого протокола передачи данных) [6]. Полученная информация на сервере записывается в базу данных фирмы-разработчика ПО. Таким образом, на сервере хранится вся информация, идентифицирующая пользователя, установившего программный продукт.

Следующим этапом является использование криптографических алгоритмов [7] для шифрования вспомогательного модуля программы *.dll.

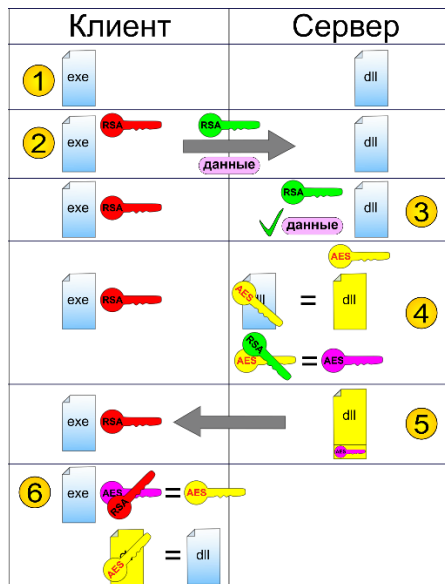


Рис. 1 — Получение вспомогательного модуля программы с использованием криптографических алгоритмов шифрования

Файлы *.dll можно защитить, используя симметричный алгоритм шифрования (например, AES) [8]. Для генерации уникального ключа симметричного алгоритма, которым будет зашифрован вспомогательный модуль, можно использовать полученную информацию из регистрационной формы пользователя.

Таким образом, вспомогательный модуль программы будет зашифрованным передаваться по сети, храниться на машине пользователя.

При использовании симметричных алгоритмов шифрования для защиты файлов, хранящихся на диске или передаваемых по сети, рекомендуется ключи, которыми шифруются файлы, также защищать. Для защиты ключей симметричных криптографических алгоритмов используют криптографические алгоритмы с открытым ключом, например алгоритм RSA.[7]

Предположим, что на сервере фирмы- разработчика ПО уже сгенерирован ключ симметричного алгоритма шифрования для пользователя, подтверждающего свою лицензию и ожидающего вспомогательного модуля программы.

Для шифрования самого ключа симметричного алгоритма используем схему шифрования с открытым ключом [7].

При регистрации пользователя в программе должна сгенерироваться пара

ключей (открытый и закрытый).[8] В дальнейшем закрытый ключ будет храниться на машине пользователя, открытый ключ вместе с персональными данными пользователя передаётся на сервер фирмы разработчика и также хранится в базе данных. С помощью открытого ключа сервер зашифрует ключ симметричного алгоритма, которым был зашифрован вспомогательный модуль программы, допишет его в зашифрованный вспомогательный модуль (например, в конец файла) и передаст программе пользователя.

При необходимости использования вспомогательного модуля программой сначала будет извлечён зашифрованный открытым ключом ключ симметричного алгоритма, с помощью закрытого ключа ассиметричного алгоритма будет расшифрован ключ симметричного алгоритма, затем с помощью ключа симметричного алгоритма будет расшифрован сам файл вспомогательного модуля. Программа сообщает об успешной активации лицензии. Теперь, при необходимости обращения к вспомогательному модулю, программа расшифровывает его "на лету", а при завершении работы удаляет расшифрованный файл из памяти.

Достоинства и недостатки подсистемы защиты авторского права программного обеспечения

Описанная подсистема защиты ПО обладает рядом достоинств и недостатков.

Достоинства:

- обычный опытный пользователь не сможет обойти защиту;
- система защиты способна справиться с крэкерами-новичками, а также запутать крэкеров среднего уровня;
- в случае использования эффективного средства обфускации для сокрытия алгоритма подсистемы защиты, а также использования методов виртуализации функций системы защиты, даже опытным крэкерам понадобится немало времени, чтобы исследовать и взломать алгоритм проверки лицензии;
- для получения вспомогательного модуля, для защиты которого используются криптографические алгоритмы, нужно знать ключи и используемые криптографические алгоритмы;
- взлом базы данных сервера фирмы- разработчика ПО, на котором хранится персональная информация о пользователях и их серийных номерах программы, не даст информации обо всех используемых ключах криптосистемы, а значит, не позволит расшифровать вспомогательный модуль программы и использовать незаконно программный продукт;
- для предотвращения использования метода перебора ключей, рекомендуется использовать значительную длину ключей в симметричных и ассиметричных алгоритмах шифрования.

Недостатки:

- без обфускации подсистема защиты не защищена от декомпиляции и

дизассемблирования с целью внесения изменений в алгоритм работы функций защиты (патчинг);

- при активации программы необходимо подключение к интернету;
- в случае использования информации об аппаратном обеспечении конечного пользователя для генерации уникального ключа, которым будет зашифрован вспомогательный модуль, при замене какой-либо аппаратной части необходимо будет сообщать разработчику об этом и проходить снова процедуру активации программы.

Выводы. В ходе выполнения данной работы были рассмотрены и проанализированы существующие механизмы защиты авторского права на программное обеспечение, предложен общий алгоритм работы подсистемы защиты авторского права программного обеспечения, рассмотрена возможность использования криптографических алгоритмов шифрования для защиты вспомогательного модуля программного обеспечения, представлены достоинства и недостатки предлагаемой подсистемы защиты. На данный момент не существует абсолютной программной защиты ПО, которое невозможно было бы взломать, если не учитывать время, затраченное на исследование этой защиты. Однако наиболее эффективными средствами при создании подобного рода систем проверки лицензии являются использование криптографических алгоритмов и обфускация функций подсистемы защиты.

Список литературы

1. Виды лицензий на программное обеспечение // Information Technology Engineering Projects. [Электронный ресурс]. – Режим доступа: http://www.it-epru.ru/knowledge_base/software_licensing/type_of_software_license/
2. Серада С.А. Оценка эффективности систем защиты программного обеспечения // CITForum. [Электронный ресурс]. – Режим доступа: <http://citforum.ru/security/software/seredal/>
3. Касперски К., Рокко Е. Искусство дизассемблирования. – СПб.: БХВ-Петербург, 2008. – 896 с.
4. Сляров Д. Искусство защиты и взлома информации. – СПб.: БХВ-Петербург, 2004. – 288 с.
5. Работа с библиотеками динамической компоновки (DLL) // Visual C++. [Электронный ресурс]. - Режим доступа: <http://www.softzenware.com/visual/visual12.html>
6. Каналы защищенной передачи данных //Энциклопедия теоретической и прикладной криптографии. [Электронный ресурс] - Режим доступа: <http://cryptowiki.net/>
7. Столлингс В. Криптография и защита сетей. - М.: Издательский дом "Вильямс", 2001 г., - 669 стр.
8. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. - М.: ДМК Пресс, 2002 г. – 655 стр.

УДК 519.254

**Описание и программная реализация методов обработки данных для
повышения точности прогнозирования
Рычка О.В.**

Донецкий национальный технический университет
olga_rychka@mail.ru

Аннотация

Рычка О.В. Описание и программная реализации методов обработки данных для повышения точности прогнозирования. В работе описаны методы, позволяющие повысить точность прогнозирования при использовании линейных регрессионных моделей и их основные преимущества. Рассмотрены этапы работы программного приложения, разработанного автором на языке программирования Visual Basic for Applications для реализации предложенных методов.

Введение. В настоящее время прогнозирование используется во всех сферах человеческой деятельности. На основании прогноза принимаются обоснованные решения. Таким образом, большое значение имеет точность измерений и прогнозов, сделанных на основе имеющихся данных.

Для прогнозирования значений переменной, часто используется регрессионный анализ. Регрессионные модели позволяют прогнозировать значения по данным объясняющих переменных. Однако в исследуемую выборку могут попасть отдельные результаты, значения которых значительно отличаются от остальных. Эти результаты представляют собой аномальные измерения. Для обнаружения таких измерений, в настоящее время, существуют различные критерии [1-5]. К данным методам относятся – метод Титьена-Мура-Бекмана, Эктона и Прескотта-Лунда. Их основным преимуществом является простота понимания и использования. Однако, в ходе исследований, был выявлен ряд существенных недостатков [6]. Основным из них является то, что при использовании данных критериев, нахождение аномальных данных происходит методом перебора. С увеличением объема выборки повышается и трудоемкость, т.к. расчеты необходимо проводить для каждого подозрительного значения в отдельности. Также в рассматриваемой выборке может оказаться большее число аномальных измерений, чем исследуется на выбросы. В связи с описанными недостатками существующих на данный момент критериев в [6] и [7] были предложены новые методы, позволяющие повысить качество регрессионных прогнозных моделей. Для упрощения использования предложенных методов и их модификаций автором было разработано программное приложение с использованием языка программирования Visual Basic for Applications (VBA).

Цель исследований. Целью данной работы является определение основных условий выбора одного из двух методов повышения точности прогнозирования или их модификаций, а также рассмотрение этапов работы программы, разработанной с помощью языка программирования VBA.

Описание методов. Задача предлагаемых методов заключается в том, что среди всех исходных статистических данных определяются измерения, которые выходят за пределы прямоугольной области со сторонами $2k \cdot \sigma_e$ и $2k \sigma'_e$. Здесь k – коэффициент (обычно $0,6 \leq k < 3$), который соответствует вероятности попадания в заданную область. Вероятность попадания в область рассчитывается по формуле (1) [8]:

$$P_0 = 2 \cdot \frac{1}{\sqrt{2\pi}} \int_0^k e^{-t^2} dt \quad (1)$$

В таблице 1 представлены значения коэффициента k , соответствующие различным вероятностям P .

Таблица 1. Значения коэффициента k при различных вероятностях P

P	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.5
k	1.9	1.75	1.6	1.5	1.4	1.3	1.2	1.05

Среднеквадратические отклонения невязок σ_e и σ'_e , определяются по формулам (2) и (3) соответственно:

$$\sigma_e = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n-2}} \quad (2)$$

$$\sigma'_e = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}'_i)^2}{n-2}} \quad (3)$$

где Y_i – фактические значения;

\hat{Y}_i – рассчитанные значения по исходному уравнению;

\hat{Y}_i' – рассчитанные значения по уравнению, график которого будет перпендикулярен исходному.

Сравнение методов. Отличие реализации методов состоит в том, что в одном методе статистические данные, выходящие за пределы прямоугольной области отбрасываются, а в другом, такие данные не исключаются, а переносятся на границы области.

Метод повышения качества прогнозной модели, основанный на отбрасывании исходных статистических данных следует применять в случае, если прогнозное значение $Y_{\text{прогн}}$ близко к среднему \bar{Y} , т.к. чем дальше от среднего значения \bar{Y} отстоит $Y_{\text{прогн}}$, тем большей величины достигает смещение при использовании данного метода и наоборот. Метод, основанный на перемещении данных, применяется в обратной ситуации.

Схематически метод, основанный на отбрасывании "аномальных" и ненадежных измерений, а также метод, основанный на переносе данных представлены на рисунке 1

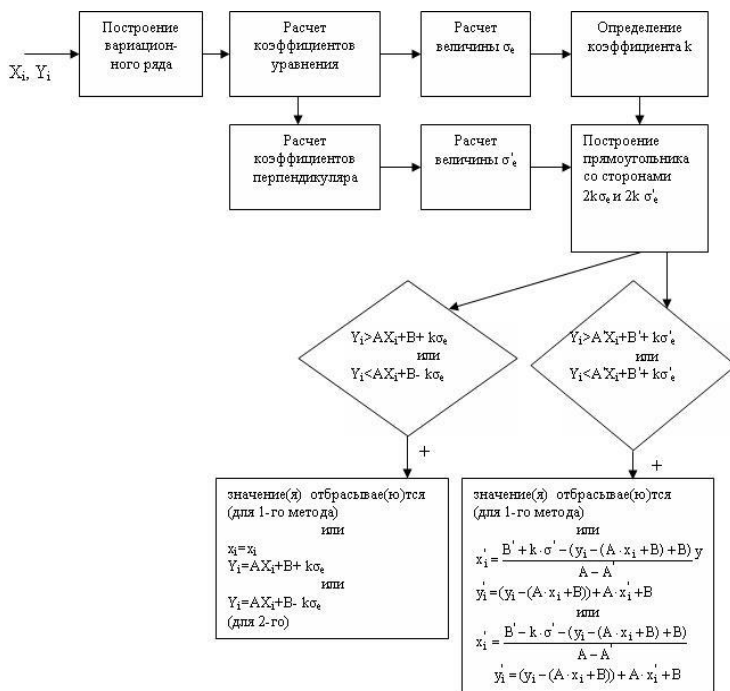


Рис. 1 – Схематическое представление методов

Основными критериями эффективности применения предложенных методов являются:

– коэффициент детерминации R^2 , который определяется соотношением (4) [9]:

$$R^2 = \frac{\sum_{i=1}^k (\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^k (Y_i - \bar{Y})^2}, \quad (4)$$

– модуль величины смещения результата прогноза $\Delta_{\text{прогн}}$ (5):

$$\Delta_{\text{прогн}} = |(A \cdot X_{\text{прогн}} + B) - (A \cdot X_{\text{прогн}} + B_n)|, \quad (5)$$

где первое и второе слагаемое, соответственно, линейное регрессионное уравнение до отбрасывания части статистики и линейное регрессионное уравнение после отбрасывания части статистики;

– доверительный интервал прогнозных значений $Y_{\text{прогн}}$ (представляет собой геометрическое место расположения прогнозных значений $Y_{\text{прогн}}$ при заданном значении $X_{\text{прогн}}$ и заданной доверительной вероятности $P_{\text{дов}}$);

– количество элементарных операций ЭВМ, необходимое для реализации методов;

– точность, которая рассчитывается по формуле 6:

$$T = R^2 \cdot \frac{m}{n} \quad (6)$$

где n – исходное количество данных;

m – количество данных, оставшихся после отбрасывания или данных, которые не подверглись преобразованию.

Наилучший вариант при этом – вариант, при котором величина коэффициента детерминации R^2 максимальна, при обязательном условии, что $T \geq 0.5$.

Описание программной реализации методов. Для удобства и быстроты реализации предложенных методов было разработано программное приложение с использованием языка программирования Visual Basic for Applications.

Работа программы содержит следующие этапы:

1. Пользователю предлагается ввести исходные данные в определенные ячейки (рис.2).

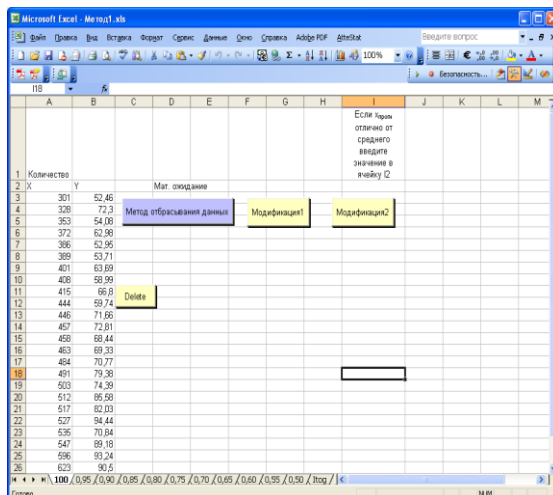


Рис. 2 – Вид окна программы после введения данных

2. По умолчанию прогнозное значение $Y_{\text{прогн}}$ определяется при $X_{\text{прогн}}=X_{\text{ср}}$. Если же это не так, пользователь вручную может ввести в соответствующую ячейку произвольное значение $X_{\text{прогн}}$ при котором будет рассчитана величина $Y_{\text{прогн}}$.

3. Поскольку каждый из двух методов имеет по две модификации необходимо определить, следует ли применять сам метод или достаточно использовать одну из его модификаций. В случае если исследователь уверен в исходных значениях независимой переменной X (например, значения X_i даны заранее и в них не может быть случайных ошибок), следует применять первую модификацию метода. Она заключается в том, что область, в пределах которой, располагаются "надежные" данные представляет собой не прямоугольник, а коридор, границы которого равноудалены от линии исходного регрессионного уравнения. Расстояние между границами данного коридора составляет $2k\sigma_x$. В отличие от первой модификации метода повышения качества прогнозной модели во второй модификации находятся 2 линии параллельные графику, построенному перпендикулярно исходного. Расстояние между ними составляет $2k\sigma'_x$. При использовании второй модификации, отсекаются данные, которые являются аномальными и ненадежными по X . Величина коэффициента детерминации R^2 в отдельных случаях уменьшается. Однако это связано с тем, что крайние точки являются определяющими для уравнения регрессии, и если они удалены на значительное расстояние от остальных точек, то они оказывают большое влияние на исходное уравнение. Следует отбрасывать (перемещать) не более 10-20% исходных статистических данных.

Реализация описанных выше модификаций проще, чем реализация предложенного метода, однако и первая и вторая модификации несколько снижают эффективность метода, поскольку в случае использования метода исключаются (переносятся) аномальные и ненадежные значения и по X и по Y одновременно. Т.о. картина является более точной.

После этого пользователь выбирает соответствующую кнопку. Нажатие кнопки вызывает следующие действия:

- а) данные сортируются по возрастанию (по независимой переменной X);
- б) рассчитывается количество введенных статистических данных;
- в) определяются коэффициенты линейного регрессионного уравнения и уравнение график которого перпендикулярен исходному;
- г) рассчитывается значение коэффициента детерминации R^2 ;
- д) находится исходная величина доверительного интервала;
- е) выводятся все вспомогательные значения (величины σ_e , σ'_e , $k\sigma_e$, $k\sigma'_e$ и т.д.) (рис.3).

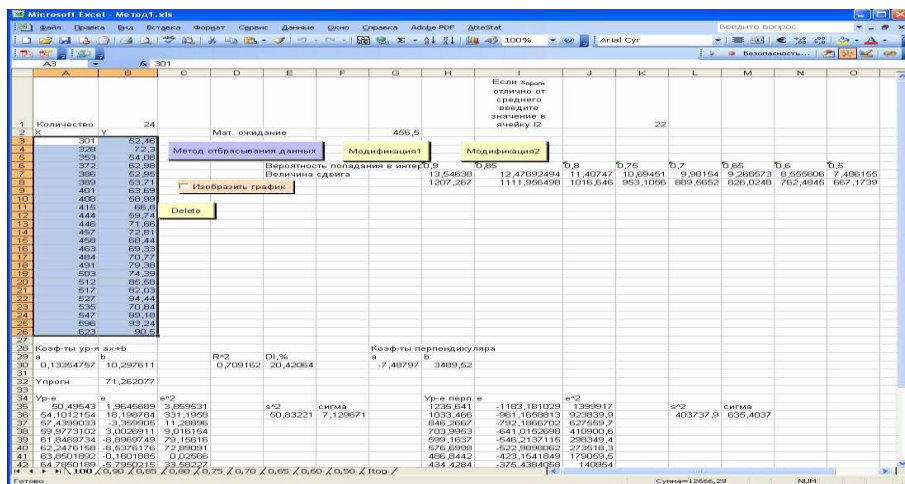
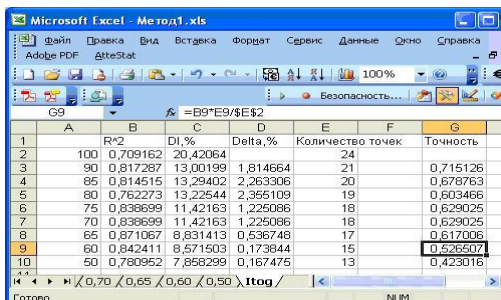


Рис. 3 – Вид исходного листа после нажатия кнопки "Метод отбрасывания данных"

ж) Для каждой вероятности P попадания исходных статистических данных в определенную область (начиная с P=1 и до P=0.5 с шагом 0.05) предусмотрен отдельный рабочий лист, на котором отображаются результаты произведенных расчетов (выводятся оставшиеся или скорректированные данные, в зависимости от применяемого метода, определяются новые коэффициенты уравнения, коэффициенты детерминации R^2 , величина доверительного интервала, величины смещения, количество элементарных операций и величина

точности). Пользователь может переключаться между этими рабочими листами и смотреть всю необходимую ему информацию;

з) формируется итоговый рабочий лист, на котором в виде таблицы представлены данные по всем основным критериям эффективности метода для каждого значения вероятности попадания в заданную область. На основании данной таблицы пользователь может сделать вывод о том, до каких пределов следует уменьшать вероятность попадания в "коридор", а также увидеть полученный от применения метода выигрыш (рис.4).



	A	B	C	D	E	F	G
1		R ²	DI, %	Delta, %	Количество точек		Точность
2	100	0,709162	20,42064		24		
3	90	0,817287	13,00199	1,814664	21		0,715126
4	85	0,814515	13,29402	2,263306	20		0,678763
5	80	0,762273	13,22544	2,365109	19		0,603466
6	75	0,838699	11,42163	1,225096	18		0,629025
7	70	0,838699	11,42163	1,225096	18		0,629025
8	65	0,871067	8,831413	0,536748	17		0,617006
9	60	0,842411	8,571503	0,173844	15		0,626507
10	50	0,780952	7,888299	0,167475	13		0,423016

Рис. 4 – Вид итоговой таблицы

4. Для наглядности пользователю доступна возможность отображения графиков, изображающих соответствующие области с выделением наблюдений, которые не попадают в заданную область (см. рис.5). Одной из особенностей разработанной программы является возможность построения графика перпендикуляра, т.к. стандартные инструменты Microsoft Excel этого сделать не позволяют.

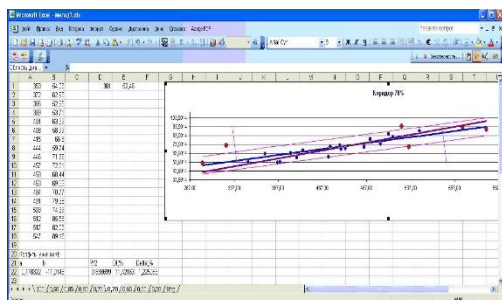


Рис. 5 – Вид листа «0.70» после нажатия кнопки «Изобразить график»

5. По окончании работы программы, все полученные данные можно сохранить с помощью стандартных инструментов программы Excel, а также очистить все рабочие листы нажатием кнопки "Delete".

Выводы. В данной статье были представлены методы, повышающие точность линейных регрессионных прогнозных моделей. Основными достоинствами данных методов является:

- простота понимания и применения;
- возможность нахождения и обработки нескольких аномальных и ненадежных измерений одновременно, что позволяет сократить время исследования, за счет отказа от простого перебора данных;
- хорошая формализуемость, что позволило реализовать данные методы в компьютерных технологиях.

Разработанное программное приложение имеет следующие преимущества:

1. Доступность понимания и использования.
2. Сокращение времени расчетов, по сравнению с ручной обработкой данных.
3. Возможность вводить произвольное количество исходных статистических данных.
4. Расчет прогнозного значения $Y_{\text{прогн}}$ для любого $X_{\text{прогн}}$.
5. Вывод основных результатов на отдельном рабочем листе для соответствующей вероятности попадания в определенную область, что позволяет проследить за основными изменениями, полученными в результате использования методов или их модификаций.
6. Отображение итоговой таблицы, которая позволяет определить выигрыш от применения метода и сделать вывод о величине, до которой следует уменьшать вероятность попадания в область.
7. Наглядное изображение областей, соответствующих выбранным вероятностям и данным, которые при этом являются "аномальными" и ненадежными.
8. Возможность изображения на графике перпендикулярных прямых, путем расширения стандартных инструментов Excel.
9. Не требуется предварительная установка на компьютере, котором уже есть Microsoft Office.
10. Маленький размер программного приложения (около 1 МБ).

Список литературы

1. Кобзарь А.И. Прикладная математическая статистика. Для инженеров и научных работников. – М.: ФИЗМАТЛИТ, 2006. – 816 с.
2. Дрейпер Н.Р., Смит Г. Прикладной регрессионный анализ. 3-е изд.: Пер. с англ. – М.: Вильямс, 2007. – 912 с.
3. Rawlings, John O. Applied regression analysis: a research tool. — 2nd ed. / John O. Rawlings, Sastry G. Pentula, David A. Dickey – USA.: Springer, 1998.

4. Мудров В.И., Кушко В.Л. Методы обработки измерений: Квазиправдоподобные оценки. – Изд. 2-е, перераб. и доп. – М.: Радио и связь, 1983. – 304 с.

5. Лемешко Б. Ю, Лемешко С. Б. Расширение области применения критериев типа Граббса, используемых при отбраковке аномальных измерений// Измерительная техника. – 2005. – № 6 – С.13-20

6. Смирнов А.В., Рычка О.В. Метод повышения качества прогнозных регрессионных моделей. // Наукові праці Донецького національного технічного університету. Серія "Інформатика, кібернетика та обчислювальна техніка". Випуск 12(165) – Донецьк: ДВНЗ "ДонНТУ". – 2010. – С.141-147.

7. Смирнов А.В., Рычка О.В. Новый метод улучшения качества прогнозных регрессионных моделей. // Наукові праці ДонНТУ Серія "Інформатика, кібернетика та обчислювальна техніка". Випуск 13(185) – Донецьк: ДВНЗ "ДонНТУ". – 2011. – С.168-172.

8. Ллойд Э. Справочник по прикладной статистике. В 2-х т. Т.1: Пер. с англ./ Под ред. Э. Ллойда, У. Ледермана, Ю.Н. Тюрина. – М.: Финансы и статистика, 1989. – 510 с.

9. Абрамовиц М. Справочник по специальным функциям с формулами, графиками и математическими таблицами./ Под. ред. М.Абрамовица и Н. Стигана: Пер. с англ. под ред. В.А. Диткина и Л.И. Кармазиной. – М.: Наука, 1979. – 830 с.

УДК 004.4

Разработка графической оболочки для операционной системы Android Калиниченко Г.А.

Донецкий национальный технический университет
Кафедра программной инженерии
E-mail: glebjn@gmail.com

Аннотация

Калиниченко Г.А. Разработка графической оболочки для операционной системы Android. Рассмотрена разработка графической оболочки для ОС Android. Определены основные структуры данных для хранения пользовательских данных. Рассмотрено взаимодействие отдельных модулей программной системы. Рассмотрены основные средства защиты личной информации пользователя.

Общая постановка проблемы. В последние годы активное развитие получила сфера мобильных технологий. Каждый год крупные компании предоставляют новые версии операционных систем, в которых предлагаются новые технологические решения. В данный момент существует широкий список лаунчеров для разных операционных систем, которые доступны в соответствующих магазинах приложений. Но не все графические оболочки имеют грамотно выстроенный и понятный пользовательский интерфейс, что довольно важно в мобильных устройствах, у которых объем свободного пространства ограничен. Также важным моментом для карманных девайсов является быстродействие, т.к. медленная работа может негативно повлиять на мнение пользователя насчет программного продукта. Касательно графической оболочки это скорость получения информации от базовых подсистем операционной системы. Исключительным фактом, является то, что далеко не все лаунчеры предоставляют защиту личных пользовательских данных.

Разработка. UI часть приложения представлена классом MainActivity и набором фрагментов для каждого экрана. В нем происходит загрузка основного фрагмента PhantomHomePageItemFragment, которое содержит основное меню приложения. Также в MainActivity происходит реакция на событие back, для того, чтобы можно было вернуться к следующему фрагменту, который храниться в стеке фрагментов. [1] Отдельные пункты меню представлены специальными виджетами ImageButton, по клику на которые пользователь переходит на другой экран приложения. На рис. 1 показана схема архитектуры фрагментов приложения. Для их создания в приложении используется базовый класс BaseFragment с событием для последующего реагирования при нажатии на back. Существуют экраны, в которых используется action bar. Для добавления фрагмента с action bar, необходимо наследоваться от класса BaseFragmentWithActionBar, в котором происходит инициализация как фрагмента содержащий action bar, так и основного фрагмента, который содержит контент.

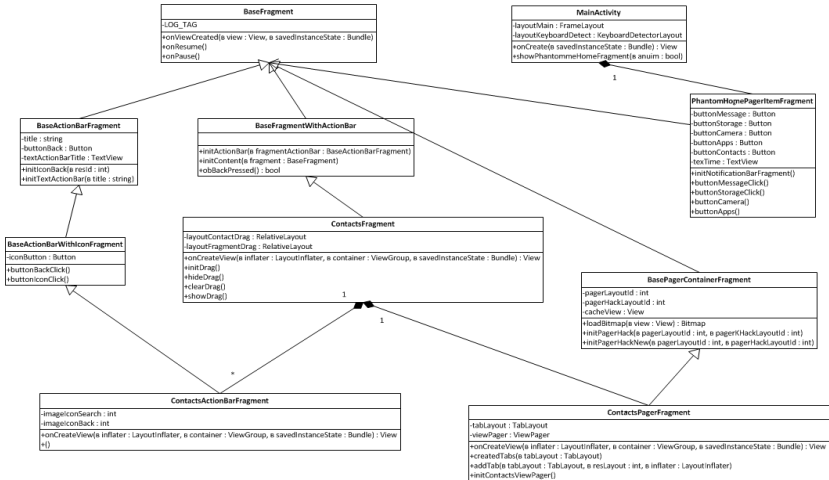


Рис. 1. Диаграмма классов для работы с фрагментами

Для того, чтобы создать фрагмент с action bar необходимо наследоваться от базовых классов BaseActionBarFragment или BaseActionBarWithIconFragment. Класс BaseActionBarFragment позволяет создать фрагмент с action bar без кнопки опции. В нем содержится заглавие action bar, кнопка back и сам textView для отображения заголовка, а также их инициализация. К примеру, есть фрагмент ContactsFragment, он выводится с action bar, поэтому в нем находится фрагмент с action bar – ContactsActionBar и фрагмент контента ContactsPagerFragment.

В операционной системе предусмотрена работа с базой данных. В пакете android-sdk есть реляционная база данных SQLite, в которой сохраняется информация о файлах. Данные о файлах пользователя сохраняется в базе данных. Модуль базы данных для контактов состоит из: классов сущностей, классов таблиц и классов dao. Диаграмма классов для классов сущностей представлена на рис. 2. Классы сущности предназначены для хранения структур данных приложения. Класс AbstractDBEntity содержит описание id поля для каждой таблицы. Класс ContactDBEntity содержит полное описание информации о пользователе. Класс FieldDBEntity содержит описание полей для контактов, т.к. у каждого контакта есть свое поле, к которому он относится (телефон, email). Класс FieldsToContactDBEntity – сущность для таблицы пересечения, т.к. один и тот же контакт может относиться к нескольким полям. Класс AbstractTypeDBEntity содержит id для сущности и абстрактный метод, который возвращает непосредственно тип контакта в виде строки. Классы, которые являются потомками для AbstractDBEntity, описывают непосредственно типы полей в контакте (у поля телефон может быть тип рабочий, мобильный).[2]

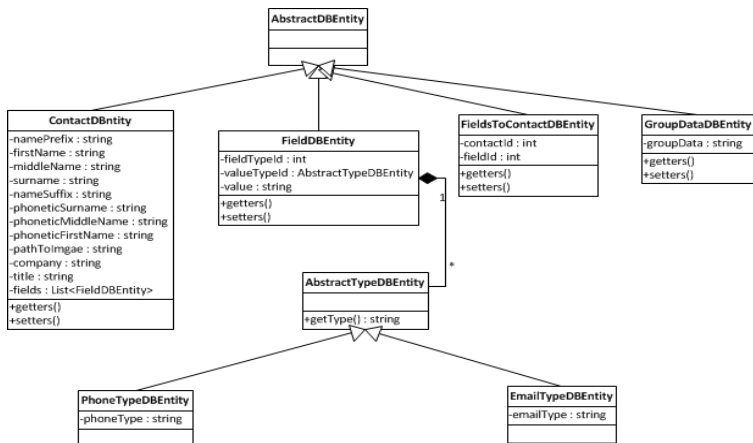


Рис. 2. Диаграмма классов для сущностей базы данных

Диаграмма классов таблиц представлена на рис. 3. Класс AbstractTable описание id поля, а также структуру описания типа поля. Класс AbstractField содержит в себе структуру для каждого поля: тип самого поля (Integer, Text), уникальное ли поле. Класс ContactMetadataTable содержит структуру для таблицы с информацией о пользователе и непосредственное ее создание. Класс FieldMetadataTable содержит описание таблицы, которая хранит информацию о полях, к которым может относиться контакт. Класс FieldsToContactTable содержит описание таблицы пересечения для контактов и полей, которые там содержатся. Остальные таблицы описывают таблицы-типы, которые могут быть у классов полей.

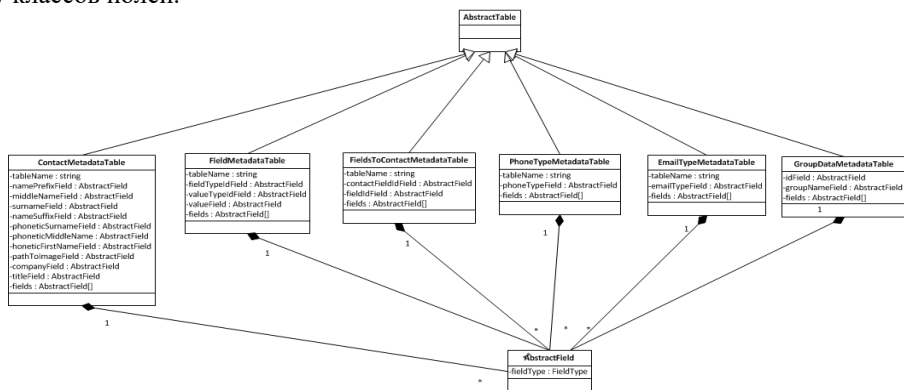


Рис.3. Диаграмма классов для таблиц

Диаграмма классов для dao-классов представлена на рис. 4. Класс dao содержит основную логику работы с базой данных. В ней происходит формирование основных запросов: на добавление, обновление и удаление данных из соответствующих таблиц. Запросы на добавление, обновление и удаление хранятся в AbstractDao и они выполняются во всех dao-потомках.

В ContactMetadataDao содержится ссылка на FieldMetadataDao и FieldsToContactDao, т.к. FieldsToContactDao таблица пересечения содержит информацию о контактах и полях.

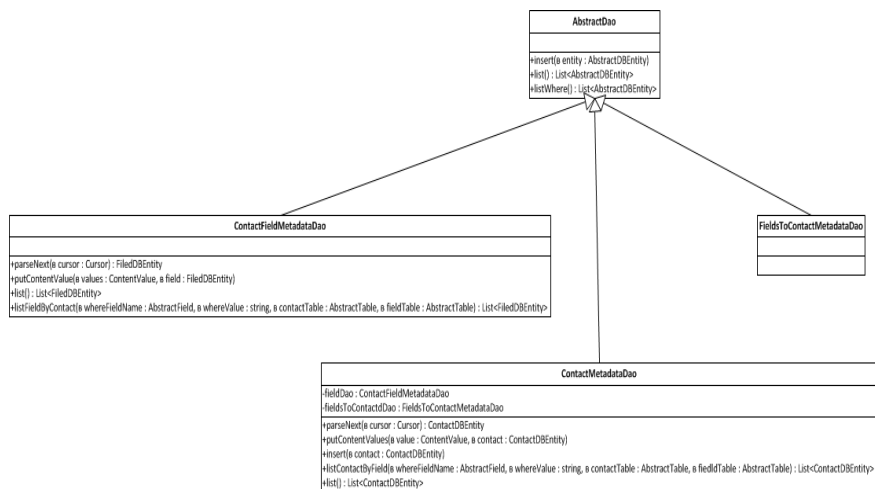


Рис.4. Диаграмма классов dao-классов

Личные данные пользователя должны храниться в защищенном формате. Для повышения уровня безопасности данные пользователя сохраняются во внутренней памяти устройства, таким образом, устройство, которое имеет root-права, не сможет увидеть эти данные, что увеличивает защиту устройства. Также конфиденциальная информация о пользователе можно сохранять в скрытых директориях, что предусмотрено возможностями ядра операционной системы Linux. После того, как пользователь выполнит действие по фотографированию объекта, изображение будет сохранено в скрытой временной директории. После этого, происходит занесение информации о файле в базу данных SQLite, затем выполняется шифрование файла. Для шифрования файла можно применить криптографическую библиотеку Scrupto от Facebook. Она предназначена для шифрования файлов с большим объемом данных. Отличительной особенностью данной библиотеки является ее быстрдействие. Библиотека Scrupto основывается на криптографических алгоритмах из OpenSSL,

и эффективно управляет памятью как в нативном коде так и на Java коде. Также используются режимы AES GCM, HMAC. AES GCM – режим, который использует в качестве основного алгоритма блочного шифрования AES с размером ключа 128 или 256 бит. HMAC – механизм проверки подлинности сообщения (аналог механизма электронно - цифровой подписи). HMAC – использует стандарт MAC - стандарт, описывающий способ обмена данными и способ проверки целостности передаваемых данных с использованием секретного ключа, по которому два клиента, использующие MAC, как правило, разделяют общий секретный ключ. [3]

После шифрования файла временный не зашифрованный файл удаляется, при этом сохраняется его путь во внутренней памяти устройства, затем это путь присваивается зашифрованному файлу, который помещается во внутреннюю память пользовательского девайса. Шифрование файлов выполняется в отдельном потоке, это позволяет быть UI – потоку приложения более отзывчивым и не тратить свои ресурсы на операцию шифрования.

Изначально для шифрования и дешифрования файла нужно подключить сам Crypto объект. Он подключается через нативную крипто-библиотеку, которая поставляется вместе с файлом Crypto. Для шифрования необходимо создать выходной поток файла, который нужно зашифровать. Также необходимо создать выходной зашифрованный поток, который является непосредственно надстройкой над обычным Java выходным потоком. В зашифрованный выходной поток необходимо передать обычный выходной поток файла, содержащий фото или видео файл, а также значение самого ключа. Впоследствии можно получить обычный выходной поток и сохранить его в файл. Для дешифрования все действия аналогичны, только вместо выходного потока создается и передается входной поток.

Т.к. фото и видео файлы пользователя хранятся в памяти смартфона, необходимо чтобы при непосредственном их использовании они дешифровывались на лету. Дешифровка фото выполняется по механизму выше и фото хранятся в памяти. Но дешифрование видео файлов несколько отличается, тем, что высокая скорость чтения/записи при шифровании/дешифровании достигается путем параллельной обработки данных, т.е. байты шифруются и дешифруются не последовательно, а параллельно и возможна проблема получения массива байт видео файла не в правильном порядке. Поэтому для видео файла дешифровка должна происходить непосредственно через временный файл. Изначально, зашифрованное видео дешифруется во временный файл в скрытой директории. Для воспроизведения файла используется данный файл, а зашифрованный остается нетронутым. После воспроизведения или отмены воспроизведения, вышеуказанный файл удаляется из скрытой директории.

Выводы. Разработка графической оболочки является емкой задачи. Из-за далеко не самых мощных процессоров находящихся в карманных устройствах,

следует обращать внимание на использование ресурсов процессора. При разработке лаунчера следует придерживаться следующих требований: наличие понятного пользовательского интерфейса, т.к. количество свободного пространства на дисплее ограничено, высокий уровень отзывчивости приложения. Также оптимизация работы с базой данных, т.к. это может увеличить производительность приложения, что является не маловажным. Особое место в разработке графической оболочки следует выделять безопасности, из-за того, что стандартные средства не предоставляют должный уровень защиты личных пользовательских данных.

Список литературы

1. Сатия Коматинени, Дэйв Маклин. Android 4 для профессионалов. Создание приложений для планшетных компьютеров и смартфонов = Pro Android 4. — М.: Вильямс. — 880 с. — ISBN 978-5-8459-1801-7
2. Голощапов А. Google Android: программирование для мобильных устройств. — СПб.: БХВ-Петербург, 2010. — 448 с. — ISBN 978-5-9775-0562-8
3. Блэк У. Интернет протоколы безопасности. Москва: издательство «Питер». 2001. ISBN 5-318-00002-9 (ISBN оригинала на английском языке: ISBN 0-13-014249-2)

УДК 004.386

Анализ проблем построения эвм с использованием троичной логики

Бычкова Е.В., Нестеренко В.С.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
E-mail: nes.vadim.s@gmail.com

Аннотация

Бычкова Е.В., Нестеренко В.С. Анализ проблем построения ЭВМ с использованием троичной логики. Проанализированы современные проблемы построения ЭВМ с использованием троичной логики. Выдвинуты предположения авторов о будущем современных методов реализации ЭВМ с использованием троичной логики.

Ключевые слова: троичная логика, транзистор, триты, трайты, кубиты, алгоритм RSA.

Постановка проблемы. В настоящее время подавляющее большинство ЭВМ используют двоичную систему счисления, причем развитие технологий идёт в направлении увеличения скорости вычислений, не предавая внимание

другому подходу развития - замене двоичной логики троичной. Троичная логика имеет больший вмещаемый запас чисел, что даёт ей преимущество над двоичной. В связи с этим проблема построения ЭВМ на троичной логике на сегодняшний день весьма актуальна.

Анализ литературы. Проведен анализ подходов построения ЭВМ с использованием троичной логики, необходимой для многократного увеличения вычислительных мощностей ЭВМ. Приводятся основные характеристики, определяющие практическую ценность троичного кода и трехзначной логики.

Цель статьи - выявить и проанализировать актуальные проблемы построения ЭВМ с использованием троичной логики.

Трехзначная логика. Трехзначная логика (первая и наиболее известная многозначная логика) представляет собой три возможных состояния: «истина», «ложь» и «не определено». Такая логика существует в двух вариантах, несимметричная и симметричная. В несимметричной троичной логике применяются цифры «0, 1, 2», а в симметричной троичной системе счисления знаки «-, 0, +».

Преимущества симметричной системы состоят в том, что, во-первых, имеет место естественное представление чисел со знаком, т.е. не нужно использовать прямой, обратный или дополнительный код - число отрицательно, если его ведущий разряд отрицателен, и наоборот, а инвертирование (смена знака) числа производится путем инвертирования всех его разрядов; во-вторых, округление здесь не требует каких-то специальных правил и производится простым обнулением младших разрядов.

При применении троичной логики проще производится сравнение чисел по величине, поскольку не нужно обращать внимание на знак числа; в соответствии с этим команда ветвления по знаку в троичной машине занимает в два раза меньше времени, чем в двоичной. Ряд исследователей апеллирует к тому, что трехзначная логика значительно более натуральна и свойственна человеческому мышлению, чем двоичная.

Кроме того, из всех позиционных систем счисления троичная наиболее экономична - в ней можно записать больший диапазон чисел, нежели в любой другой системе при равном количестве используемых ячеек. Самой экономичной можно считать систему счисления с основанием, равным числу Эйлера ($e \approx 2,718\dots$), и 3 — наиболее близкое к нему целое.

Поиск основания системы счисления по минимуму объёма памяти для заданного количества информации равен основанию натуральных алгоритмов:

$$Y(a) = \frac{\ln y(a)}{a} = \frac{\ln a}{a} . \quad (1)$$

Функция достигает максимума в точке $a = e$, что и означает, что у троичной системы памяти плотность записи выше.

Если в привычных нам двоичных компьютерах информация измеряется в битах и байтах, то компьютеры на троичной системе счисления оперируют новыми единицами: тритами и трайтами. Трит — это один троичный разряд, принимающий значения (+), (0) или (-). Один трайт традиционно равен шести тритам и может принимать 729 различных значений (байт — только 256). Впрочем, возможно, в будущем трайты станут 9- или 27-разрядными, что естественнее, так как это степени тройки.

ЭВМ с шестнадцатиразрядной шиной (подсистемой, служащей для передачи данных между функциональными блоками компьютера) в двоичной логике обеспечивает поддержку 2^{16} адресов памяти, тогда как в троичной логике - 3^{16} , что составляет около сорока трёх миллионов адресов.

Троичная логика всё чаще используется в области телекоммуникаций для новых высокоскоростных модемов с применением трёхчастотного способа передачи. Такую полосу частот формируют два трёхчастотных генератора, способных за один такт передать 9 значений.

При организации защиты каналов связи: даже при наличии прямого доступа к такому каналу для расшифровки данных необходим специальный дешифратор для троичной логики.

Построение троичной ЭВМ с использованием импульсных элементов

При разработке троичной ЭВМ «Сетунь» в вычислительном центре МГУ в далеком 1959 году Н.П. Брусенцов разработал троичную феррит-диодную ячейку. В такой ячейке один трит записывался в два двоичных разряда, а четвёртое состояние двух двоичных разрядов не использовалось, т.е. (0, 0) - это «0», (1, 1) - это «0», (0, 1) - это «-1», (1, 0) - это «+1». Однако эта разработка не получила распространения и последующей популяризации.

ЭВМ с использованием трёхуровневых элементов. При реализации троичной логики с использованием транзисторов требуется больше элементов, чем в двоичной логике. Однако преимущество состоит в том, что чем больше будет добавлено троичных элементов, тем меньше потребуется соединений, а в определённый момент добавления таких элементов их количество начнёт расти нелинейно. Если в двоичной ЭВМ приходится каждый раз удваивать количество элементов, то в троичной ЭВМ нужно будет увеличить количество элементов в 1,5 раза, затем в 1,3 раза и т. д. Однако троичная система не эффективна на полупроводниках: если кодируется троичный сигнал напряжением, то для того чтобы получить третье состояние, нужно расширять диапазон между уровнями сигнала и экранировать полупроводники от электрических помех качественнее, т.е. увеличивать защиту элементной базы от взаимного влияния электрических полей и внешних электрических полей с помощью металлического экрана. Такой экран уменьшает энергию электромагнитных волн при помощи поглощения этой энергии проводящей средой или при помощи отражения энергии в месте границы двух сред.

Допустим, существует 2 уровня напряжения – U_0, U_1 . В двоичной логике современные микросхемы работают с напряжением, равным приблизительно 1,5В, т.е. напряжение ниже 1В в логике будет считаться нулём, а напряжение, равное 1В и выше – единицей. Поскольку всегда существуют электрические помехи, то появляются скачки и перепады напряжения, в результате которых напряжение не всегда является стабильным, что приводит к возникновению ситуации, при которой нельзя считать напряжение в полупроводнике равное 0,8-1,2В достоверным. Получается так называемый “зазор”, из-за которого невозможно определить, будет ли считанное значение в данном диапазоне равным единице или нулю. В троичной логике микросхемы на полупроводниках могли бы работать с тремя напряжениями, а, значит, такой зазор был бы незначительным. Следовательно, пришлось бы сделать его большим, что привело бы к увеличению напряжения и диапазона между уровнями напряжения, а это, в свою очередь, привело бы к уменьшению быстродействия и увеличению выделения тепла, которое пропорционально квадрату напряжения.

Квантовые компьютеры. Идея использования квантовых вычислений впервые была высказана в 1980 году советским математиком Ю.И. Маниным в его монографии «Вычислимое и невычислимое». В 1982 году была опубликована статья на ту же тему американского физика-теоретика, нобелевского лауреата Ричарда Фейнмана. Он заметил, что определенные квантово-механические операции нельзя в точности переносить на классический компьютер. Это наблюдение привело его к мысли, что подобные вычисления могут быть более эффективными, если их осуществлять при помощи квантовых операций. С этого момента началось развитие квантовых вычислений и квантовых компьютеров.

В 1994 году американским ученым Питером Шором были разработаны алгоритмы факторизации чисел и вычисления дискретного логарифма. Алгоритм факторизации позволил разложить число на простые множители за время $O(\log^2 N \log^3(\log N))$, используя $O(\log N)$ логических кубитов.

В 2001 году специалисты компании IBM реализовали алгоритм, предложенный Шором, на своем прототипе квантового компьютера. Несмотря на то, что в эксперименте квантовый компьютер разложил на множители всего лишь двузначное число, реализация масштабируемого квантового компьютера может привести к непригодности алгоритма шифрования RSA и других алгоритмов, опирающихся на односторонние функции, входные значения которых легко вычисляются с помощью квантовых компьютеров. Например, факторизация 155-значного числа (512 бит) на современном компьютере займет около 35 лет, квантовый же справится за пару минут.

Канадская компания D-Wave заявила в феврале 2007 года о создании образца квантового компьютера, состоящего из 16 кубитов (квантовых аналогов битов).

Согласно некоторым исследованиям, использование кубитов (аналогов тритов в

квантовом компьютере) вместо кубитов позволит упростить реализацию квантовых алгоритмов и компьютеров.

ЭВМ на иных компонентах. На сегодняшний день такими компаниями как IBM, Motorola и Texas Instruments ведутся разработки и исследования с кремниво-германиевыми сплавами (SiGe), используя который можно реализовать электронные схемы, работающие с тремя и более уровнями сигнала.

В научной работе «Архитектура троичного оптического компьютера» Джин Йи, Хи Хуакан и Лу Янгтиан предлагают использовать два состояния поляризации света с ортогональными направлениями колебаний, используя кольцо оптоволокна в качестве регистра, а полупроводниковую память для троичной ячейки и жидкие кристаллы в качестве модуляторов и сумматоров.

В настоящее время исследуются гетеропереходные проводники, которые состоят из слоев различных сред, что позволяет им генерировать несколько уровней сигнала, позволяющих не увеличивать количество элементов, однако такая разработка на сегодняшний день довольно затратная. Поэтому бытующее мнение о том, что не предпринимаются попытки создать троичную ЭВМ и разработать элементную базу для нее, неверно.

Выводы. В заключение можно отметить, что ЭВМ, основанные на троичной логике и троичной системе счисления, превосходят их двоичные аналоги. Главная проблема реализации троичной ЭВМ на сегодняшний день - отсутствие элементной базы, которая способна находится в трёх состояниях и при этом работать без помех при переходах между состояниями. Однако уровень технологического развития не стоит на месте. Вполне возможно, что, исследуя новые технологии, в будущем появится элементная база, способная заменить двоичную по затратам на производство и стабильности при смене состояний.

Дональд Кнут отмечал, что из-за массового производства двоичных компонентов для компьютеров, исследованию и разработке троичных компонентов отводится малое место в науке. Однако троичная логика элегантнее и эффективнее двоичной и в будущем, возможно, вновь вернуться к её разработке.

Список литературы

1. Попов И.Ю. Квантовый компьютер и квантовые алгоритмы. Учебное пособие / И.Ю. Попов - СПб.: СПбГИТМО, 2007. - 88 с.
2. Бычкова Е.В., Витер А.С., Голубенко А.О. Стойкость современных криптосистем с открытым ключом перед техническим прогрессом. / Е.В. Бычкова, А.С. Витер, А.О. Голубенко // Информационные управляющие системы и компьютерный мониторинг (ИУС КМ – 2013): IV Всеукраинская научно-техническая конференция студентов, аспирантов и молодых ученых, 24-25 апреля 2013, Донецк: сб. докл./ Донец. национал. техн. ун-т.- Донецк: ДонНТУ, 2013. В 2тт.-Т1. С. 544 - 548.
3. Боярченков М.А. Магнитные элементы автоматики и вычислительной

техники / М.А.Боярченко, А.Г. Черкашина. –М.: «Высшая школа», 1976. – 383 с.

4. Кнут Д. Искусство программирования для ЭВМ. Том 1. Основные алгоритмы: пер. с англ. / Д. Кнут. - М.: МИР, 1977. - 720 с.

5. Брусенцов Н.П. Опыт создания троичных цифровых машин. / Брусенцов Н.П., Жоголев Е.А., Маслов С.П., Рамиль Альварес Х. // Компьютеры в Европе, Прошлое, настоящее и будущее. - Киев, «Феникс», 1998. С. 67 - 71.

6. Брусенцов Н.П. Пороговая реализация трехзначной логики электромагнитными средствами / Брусенцов Н.П., Шауман А.М. // Вычислительная техника и вопросы кибернетики - Вып. 9. -М.: Изд-во Моск. ун-та, 1972. С. 3 - 35.

7. Брусенцов Н.П. Заметки о троичной цифровой технике. / Брусенцов Н.П., Шауман А.М. // Архитектура и программное оснащение цифровых систем. М.: Изд-во Моск. ун-та, 1984. С. 114 - 123.

УДК 004.934.5

Применение систем синтеза речи

Пикалев Я.С.

Государственное учреждение «Институт проблем исследования искусственного интеллекта»

E-mail: pikaliiov@gmail.com

Аннотация

Пикалев Я.С. Применение систем синтеза речи. В работе проанализированы существующие средства синтеза речи. Отмечены голосовые движки, имеющие, имеющие наилучшие характеристики: большие словарные базы данных; поддержка Microsoft SAPI 4, SAPI 5; возможность создавать и редактировать словарные базы данных; поддержка технологий для улучшения синтезируемой речи; простота использования. Описана технология подключения голосовых движков к программам для ОС Windows, а также рассмотрен способ решения проблем, возникающих при подключении Microsoft SAPI к современной среде разработки Microsoft Visual Studio Community 2015.

Общая постановка проблемы. В области человеко-компьютерного взаимодействия появляется понятие «естественно-интуитивные интерфейсы» [1]. Возможность взаимодействия с машинами посредством естественно-интуитивных интерфейсов становится трендом в области компьютерной индустрии. Они предполагают взаимодействие с виртуальной реальностью посредством таких интерфейсов как распознавание образов, распознавание

миимики и положения лица, распознавание жестов рук и пальцев, а также распознавание и синтез речи.

Синтез речи – это одна из хорошо проработанных областей, в которой имеется много решений и есть даже определенные стандарты. Но не так часто программы используют эту возможность. В данной работе необходимо выбрать соответствующие средства из числа готовых программных систем и освоить технологию использования средств синтеза речи.

Обзор программных средств синтеза речи. Синтез речи в простейшем случае – это преобразование текста в звучащую речь. Синтезаторы речи могут применяться в различных сферах. Наиболее распространенными сферами применения синтеза речи являются средства телекоммуникаций (чтение электронных сообщений, голосовые оповещения), системы обучения (озвучивание видеороликов и обучающих курсов, языковые обучающие программы), помощь людям с ограниченными возможностями (клавиатуры и программы быстрого составления предложений), мультимедиа (создание электронной музыки, озвучивание фильмов). Касательно систем синтеза речи на русском языке есть некоторые сложности. Они связаны со спецификой языков славянской группы.

Ряд синтезаторов речи находится в свободном доступе – Govorilka, Балаболка, Розмовлялка, Reader TTS, ICE Book Reader. Есть также платные программы синтеза речи – TextSound, Verbose, YeoSoft Text to MP3 Speaker, Alive Text to Speech, TextAloud.

Помимо пользовательских программ синтеза речи имеются средства для разработчиков – так называемые голосовые движки. Перечислим голосовые русскоязычные движки и «голоса» для них, которые удалось обнаружить в свободном доступе: UkrVox (Igor Murashko) [2], Digalo (Nikolai) [3], Apple's Speech Synthesizer или VoiceOver (Alex) [4], Microsoft Speech API (Irina, Pavel) [5], MBROLA (возможно создать эталон по образцам голоса) [6]. Есть также платные русские голосовые движки – Vocalizer (Milena) [7], Acapela TTS (Alyona, Nikolai) [8], IVONA (Tatyana, Maxim) [9], Loquendo Embedded TTS (Olga), Sakrament TTS (Vyacheslav, Olga, Irina), Nuance (Katerina) [10].

Список существующих голосовых движков намного шире, чем приведенный. Вышеописанные движки были выделены из-за присущих им характеристик: относительно большие словарные базы данных; поддержка Microsoft SAPI 4, SAPI 5 (кроме Voiceover); возможность создавать и редактировать словарные базы данных; поддержка технологий для улучшения синтезируемой речи; простота использования (кроме MBROLA).

Создание программы синтеза речи. Согласно поставленной задаче необходимо было освоить технологию использования средств синтеза речи для последующего внедрения в пользовательские программы для ОС Windows. Также необходимо было использовать бесплатные средства разработки.

Проведен анализ программ с открытым исходным кодом, для дальнейшего внедрения в приложения или проекты. Обнаружены исходные коды на C++ программ синтеза речи на основе движков Espeak [11] и Festival [12]. Они используют библиотеку MFC, которая является платным модулем. Также обнаружены исходные коды программ, использующих движки Vox Machina [13] и VoiceOver [4]. Однако это программы для Mac OS.

При написании требуемой программы использованы Microsoft SAPI 5.1/5.4, документация [5], русскоязычный голосовой движок компании IVONA [9] (голос Maxim), а также фрагменты кодов [14, 15].

Рассмотрим последовательность действий, которая рекомендована в SAPI SDK.

1) Создать простое консольное приложение с поддержкой ATL-библиотек. После создания проекта необходимо добавить путь к файлам sapi.h и sapi.lib для реализации возможностей SAPI SDK. Найти эти файлы можно по тому пути, куда была выполнена установка SAPI SDK (по умолчанию это *C:\Program Files\Microsoft Speech SDK x*, где x – номер версии SAPI).

В диалоге настроек проекта («Проект»->«Свойства») на вкладке «C/C++» необходимо выбрать элемент «Общие» и в поле «Дополнительные каталоги включаемых файлов» ввести следующий текст (пример для SAPI 5.1): *C:\Program Files\Microsoft Speech SDK 5.1\Include*.

Чтобы установить путь к sapi.lib, нужно в том же самом диалоге настроек проекта следует выбрать вкладку «Компоновщик», на которой выбрать элемент «Ввод» и добавить в поле «Дополнительные зависимости» такой путь (пример для SAPI 5.1):

C:\Program Files\Microsoft Speech SDK 5.1\Lib\i386.

2) Допisać в заголовочном файле stdafx.h следующие строки:

```
#include "sapi.h"
#include "sphelper.h"
#define _ATL_APARTMENT_THREADED
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS
extern CComModule _Module;
```

3) Выполнить инициализацию COM через вызов функции CoInitialize(NULL). Поддержка COM должна быть инициализирована перед использованием SAPI и в течение всего времени, пока SAPI является активным, поскольку SAPI – это приложение, основанное на COM. Операция, обратная инициализации COM выполняется при помощи вызова CoUninitialize(). То есть, нужно добавить следующие строки:

```
int main(int argc, char* argv []){
    CoInitialize(NULL);
    HRESULT hr = S_OK;
    ...
```

```
CoUninitialize();
}
```

4) Создать "голос", то есть реализацию интерфейса ISpVoice. Голос – это COM-объект, создаваемый при помощи стандартного вызова CoCreateInstance(). Поиск и выбор речевых данных, таких как голосовые файлы и словари произношений осуществляют посредством следующих COM-интерфейсов: ISpObjectToken, IEnumSpObjectTokens. Также существуют интерфейсы для использования различных средств вывода звука, таких как телефон и пользовательское оборудование: ISpStream, CSpStreamFormat.

```
CComPtr <ISpVoice> cpVoice;
CComPtr <ISpObjectToken> cpToken;
CComPtr <IEnumSpObjectTokens> cpEnum;
CComPtr <ISpStream> cpStream;
CSpStreamFormat cAudioFmt;
```

```
hr = cpVoice.CoCreateInstance(CLSID_SpVoice);
```

5) Получить ссылку на указанный движок и установить его. Для этого воспользуемся функцией SpEnumTokens. В качестве второго параметра указывается указатель (ключ) на голос (движок).

```
if (SUCCEEDED(hr))
```

```
    hr= SpEnumTokens(SPCAT_VOICES, L"Name=IVONA 2 Maxim
OEM ", NULL, &cpEnum);
```

```
    if (SUCCEEDED(hr))    hr = cpEnum->Next(1, &cpToken, NULL);
```

```
    if (SUCCEEDED(hr))    hr = cpVoice->SetVoice(cpToken);
```

Информация о каждом голосе (возраст (Age), пол (Gender), язык (Language), имя (Name)) хранится в системном реестре в следующих разделах:
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voiced\Tokens (32-битной ОС) и
 HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\SPEECH\Voices\Tokens (для 64-битной ОС).

6) Указать аудиоформат при помощи функции cAudioFmt.AssignFormat, в качестве параметра указывается качество звука.

```
    if (SUCCEEDED(hr))    hr =
```

```
    cAudioFmt.AssignFormat(SPSF_22kHz16BitMono);
```

7) Задать тип вывода потока при помощи функции SetOutput.

Вызов функции для воспроизведения синтезированной речи:

```
    if (SUCCEEDED(hr))    hr =cpVoice->SetOutput(NULL,TRUE);
```

Вызов функции для сохранения синтезированной речи в файл:

```
    if (SUCCEEDED(hr))    hr =cpVoice->SetOutput(cpStream, NULL);
```

8) Для озвучивания текста вызвать метод Speak(), который принимает три параметра: указатель на строку озвучиваемого текста (Unicode-строка), набор флагов, управляющих процессом озвучивания, и указатель на long, по которому

будет размещено возвращаемое значение номера потока (в примере это значение не используется, поэтому в параметре передается NULL).

```
if (SUCCEEDED(hr))
    hr = cpVoice->Speak(L"d:\\ttstemp.txt", SPF_IS_FILENAME,
```

NULL);

9) Для записи синтезированной речи в аудиофайл – привязать поток к файлу (если выполнять совместно с синтезом речи, то нужно указывать перед кодом синтеза). Выполняется при помощи функции SPBindToFile.

```
if (SUCCEEDED(hr)) {
    hr = SPBindToFile(L"d:\\ttstemp.wav",SPFM_CREATE_ALWAYS,
    &cpStream, &cAudioFmt.FormatId(), cAudioFmt.WaveFormatExPtr());
}
```

9) Освободить потоки после окончания всех процессов, связанных с синтезом, путём вызова метода Release().

```
cpStream.Release();
cpVoice.Release();
cpEnum.Release();
cpToken.Release();
```

Указанная последовательность действий приводит к успешному созданию программ синтеза речи Microsoft Visual.NET 2003, была также протестирована её работа с голосовыми движками [2, 3, 5]. Однако, среда разработки Microsoft Visual.NET 2003 не предназначена для создания приложений для современных операционных систем (Windows Vista и выше).

При разработке версии данного приложения в среде Microsoft Visual Studio Community 2015 возникли ошибки компиляции. Для их устранения потребовалось внести некоторые изменения [17] в код заголовочного файла Microsoft SAPI «sphelper.h» (табл. 1).

Таблица 1 – Сравнение строчек кода в sphelper.h

№ Line	Исходная часть кода	Error	Измененная часть кода
Line 769	const ulLenVendorPreferred = wcslen(pszVendorPreferred)	no size_t	const size_t ulLenVendorPreferred=wcslen(pszVendorPreferred)
Line 1418	static CoMemCopyWFEX(const WAVEFORMATEX * pSrc, WAVEFORMATEX ** ppCoMemWFEX)	missing long	static long CoMemCopyWFEX(const WAVEFORMATEX * pSrc, WAVEFORMATEX ** ppCoMemWFEX)
Line 2368	const WCHAR * PropertyStringValue() const { SPDBG_ASSERT(eEventId == SPEI_PROPERTY_STRING_CHANGE); for (const WCHAR * psz = (const WCHAR *)lParam; *psz; psz++) {} return psz + 1;	moved this from init	const WCHAR * PropertyStringValue() const { SPDBG_ASSERT(eEventId == SPEI_PROPERTY_STRING_CHANGE); const WCHAR * psz = (const WCHAR *)lParam; for (; *psz; psz++) {} return psz + 1;

	}		}
Line 2560	SPPHONEID* pphoneId = dsPhoneId;	improper casting	SPPHONED* pphoneId=(SPPHONEID*)((WCHAR *) dsphoneId);
Line 2634	pphoneId += wcslen(pphoneId) + 1;	improper casting	pphoneId+=wcslen((const wchar_t *)pphoneId)+1;

Выводы. В работе проанализированы существующие средства синтеза речи. Отмечены голосовые движки, имеющие, имеющие наилучшие характеристики: большие словарные базы данных; поддержка Microsoft SAPI 4, SAPI 5; возможность создавать и редактировать словарные базы данных; поддержка технологий для улучшения синтезируемой речи; простота использования. Описана технология подключения голосовых движков к программам для ОС Windows, а также рассмотрен способ решения проблем, возникающих при подключении Microsoft SAPI к современной среде разработки Microsoft Visual Studio Community 2015.

Список литературы

1. Verplank's sketch-lecture to CCRMA HCI Technology course, Stanford University, 2000. [Электронный ресурс] – Режим доступа: <http://www.billverplank.com/Lecture/>
2. UkrVox [Электронный ресурс] – Режим доступа: <http://biblprog.org.ua/ua/ukrvox/>
3. Digalo 2000 Russian [Электронный ресурс] – Режим доступа: <http://win.tiflocomp.ru/synths/digalo/index.php>
4. Introduction to Speech Synthesis Programming Guide [Электронный ресурс] – Режим доступа: https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/SpeechSynthesisProgrammingGuide/Introduction/Introduction.html#/apple_ref/doc/uid/TP40004365-CH1-SW1
5. Microsoft Speech API (SAPI) 5.4 – Режим доступа: [https://msdn.microsoft.com/en-us/library/ee125663\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ee125663(v=vs.85).aspx)
6. The MBROLA Project [Электронный ресурс] – Режим доступа: <http://tcts.fpm.s.ac.be/synthesis/>
7. Vocalizer [Электронный ресурс] – Режим доступа: <http://www.nuance.com/for-business/text-to-speech/vocalizer/index.htm>
8. Acapela Group [Электронный ресурс] – Режим доступа: <http://www.acapela-group.com/>
9. IVONA Software is an Amazon company [Электронный ресурс] – Режим доступа: <https://www.ivona.com/>
10. Nuance Vocalizer Voices for TextAloud [Электронный ресурс] – Режим доступа: <http://sites.fastspring.com/nextup/product/nuance?store=parameters>

11. eSpeak text to speech [Электронный ресурс] – Режим доступа: <http://espeak.sourceforge.net/>

11. The Centre for Speech Technology Research. The University of Edinburgh [Электронный ресурс] – Режим доступа: <http://www.cstr.ed.ac.uk/projects/festival/>

12. Vox Machina [Электронный ресурс] – Режим доступа: <http://sveinbjorn.org/voxmachina>

13. Простой пример использования SAPI 5.x в программах на C/C++ [Электронный ресурс] – Режим доступа: http://win.tiflocomp.ru/synths/sapi/sapi5_simple.php

14. Компьютерное распознавание и порождение речи - [Электронный ресурс] – Режим доступа: <http://speech-text.narod.ru/chap5.html>

15. Сравнение предложений Visual Studio 2015 [Электронный ресурс] – Режим доступа: <https://www.visualstudio.com/products/compare-visual-studio-2015-products-vs>

16. Source SDK 2013 [Электронный ресурс] – Режим доступа: https://developer.valvesoftware.com/wiki/Source_SDK_2013

УДК 004.02

**Разработка алгоритма генерации венгерских кроссвордов
Фоминых И.Д., Рудак Л.В., Моргайлов Д.Д.**

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии

IGOS.321@gmail.com, leo2598@yandex.ua, kolleganin@yandex.ru

Аннотация

Фоминых И.Д., Рудак Л.В., Моргайлов Д.Д. Разработка алгоритма генерации венгерских кроссвордов. Разработан алгоритм для генерации игрового поля (сетки) венгерских кроссвордов. Сформулирована общая идея алгоритма и приведено его подробное описание.

Ключевые слова: венгерский кроссворд, филворд, алгоритм генерации, кластеризация.

Постановка проблемы. В последнее время все большую популярность среди владельцев смартфонов приобретают игровые приложения для решения венгерских кроссвордов (филвордов). Филворд представляет собой заполненную буквами сетку прямоугольной или квадратной формы, в которой необходимо найти спрятанные слова. Слова читаются по ломаной линии, изгибающейся только под прямым углом [1].

Наиболее ярким примером подобных программ служит мобильное приложение «Филворды: Найди Слова», суммарное количество скачиваний

которого на сервисах Google Play и App Store достигает 10 млн. [2, 3]. Несмотря на высокую популярность, приложение имеет весомый недостаток – шаблонность уровней. Приложение не генерирует уровни, а использует готовые, создаваемые и добавляемые вручную.

В открытом доступе не существует описаний и реализаций полноценного алгоритма для генерации филвордов, за исключением поверхностных решений, основанных на полном переборе вариантов [4].

Цель статьи – разработать алгоритм для автоматического заполнения словами игрового поля венгерских кроссвордов.

Идея алгоритма. Основная идея алгоритма генерации состоит в случайном разбиении матрицы игрового поля на два кластера произвольного размера с последующим делением их на подкластеры (рис. 1). Затем каждый подкластер заполняется одним или несколькими словами подходящей длины.

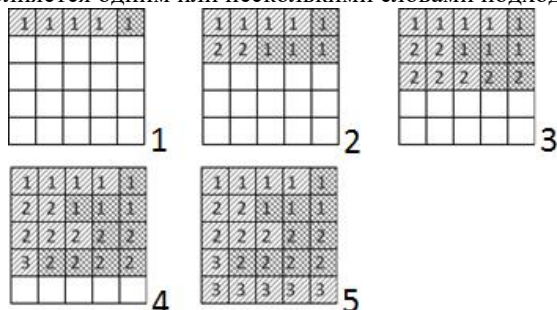


Рис. 1 – Этапы формирования игрового поля размером 5x5

На рис. 1 левый и правый кластеры обозначены разной штриховкой. Наборы клеток, помеченные одинаковыми числами, определяют соответствующие подкластеры, принадлежащие различным кластерам.

Описание алгоритма. Исходными данными алгоритма являются размер сетки игрового поля и словарь для ее заполнения. Алгоритм можно условно разделить на две стадии – кластеризация игрового поля и заполнение сформированных подкластеров словами из словаря.

На первой стадии алгоритма циклически формируются начальные N–2 строки. На каждой итерации цикла клетки очередных двух строк распределяются между кластерами и подкластерами сетки. Если текущий подкластер содержит достаточное число клеток (не меньше минимальной длины слова), то результат псевдослучайной генерации чисел определяет, добавлять ли вторую строку в подкластер или сделать ее началом следующего подкластера. В случае, когда в текущем подкластере недостаточное количество клеток, он становится двустрочным, чтобы вместить по меньшей мере слово минимальной длины.

Максимальный размер подкластера – две строки, минимальный – равен 0 (возможен только тогда, когда соседний кластер занимает всю строку). Последние две строки формируются таким образом, чтобы в итоге образовавшиеся подкластеры были достаточной длины.

По окончании формирования кластеров они последовательно заполняются словами из словаря. Длина и количество слов зависит от числа клеток заполняемого подкластера. Каждое слово с определенной вероятностью инвертируется. Максимальное количество слов в подкластере будет равняться результату целочисленного деления количества клеток подкластера на минимальную длину слова (МДС), минимальное всегда равно 1. После заполнения игровое поле поворачивается случайным образом на $0^\circ/90^\circ/180^\circ/270^\circ$.

На рис. 2 приведено дерево принятия решений, которое покрывает все частные случаи алгоритма.

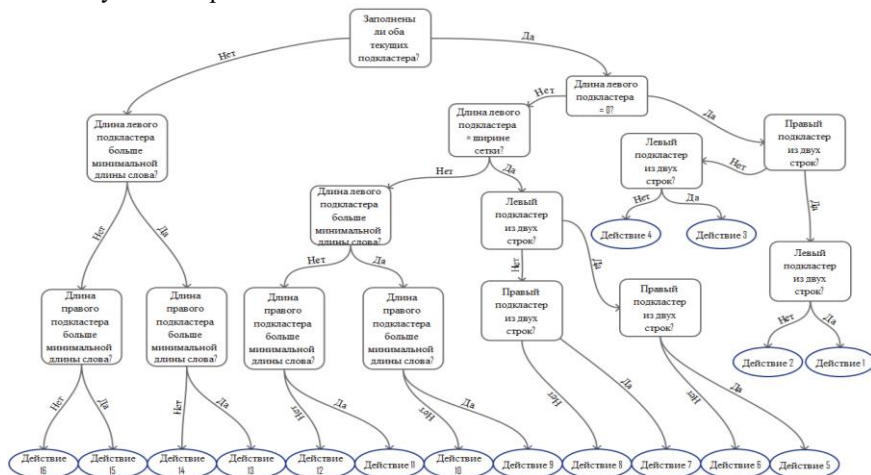


Рис. 2 – Дерево условий алгоритма

В табл. 1 описаны действия алгоритма для возможных комбинаций условий разбиения. Формулы в таблице приведены только для строк левого подкластера, так как длина строки правого подкластера всегда будет равна ширине строки (ШС) за вычетом длины строки левого.

Таблица 1 – Описание действий алгоритма

№ п/п	Описание действия
1	Для предотвращения появления двустрочного подкластера (правого или левого) со строкой нулевой длины, длина первой строки левого подкластера лежит в промежутке от 1 до ШС-1.

2	Для предотвращения появления в правом двустрочном подкластере строки нулевой длины и создания левого подкластера с недостаточным для МДС числом клеток, длина первой строки левого лежит в промежутке от МДС до ШС–1.
3	Для предотвращения появления в левом двустрочном подкластере строки нулевой длины, длина левого подкластера лежит в промежутке от 1 до ШС.
4	Для предотвращения создания левого подкластера с недостаточным для МДС количеством клеток, длина первой строки левого подкластера лежит в промежутке от МДС до ШС.
5	Для предотвращения появления двустрочного подкластера (правого или левого) со строкой нулевой длины, длина первой строки левого подкластера лежит в промежутке от 1 до ШС–1.
6	Для предотвращения создания правого подкластера с недостаточным для МДС количеством клеток и появления нулевой строки в двустрочном левом кластере, длина левого подкластера лежит в промежутке от 1 до ШС–МД.
7	Для избежания наличия в правом двустрочном подкластере со строки нулевой длины, длина первой строки левого лежит в промежутке от 0 до ШС–1.
8	Для предотвращения создания правого подкластера с недостаточным для МДС количеством клеток, длина первой строки левого подкластера лежит в промежутке от 0 до ШС–МДС.
9	Для предотвращения появления двустрочного подкластера (правого или левого) со строкой нулевой длины, длина строки лежит в промежутке от 1 до ШС–1.
10	Для предотвращения создания правого подкластера с недостаточным для МДС количеством клеток и появления строки нулевой длины в двустрочном левом подкластере, длина строки лежит в промежутке от 1 до $2 * \text{ШС} - \text{МДС}$ – длина предыдущей строки (ДПС).

Выводы. Выявлены недостатки мобильного приложения «Филворды. Найди Слова». Разработан алгоритм генерации филвордов. На данный момент алгоритм не способен реализовать все возможные варианты размещения слов на игровом поле. В частности, предложенный алгоритм не поддерживает ступенчатость (расположение слова более чем на двух строках). В дальнейшем этот недостаток может быть устранен путем усложнения процессов кластеризации и заполнения.

Список литературы

1. Филворд [электронный ресурс] // Викисловарь – многоязычный открытый словарь: [сайт]. [2016]. URL: <https://ru.wiktionary.org/wiki/филворд>

2. Филворды: поиск слов [электронный ресурс] // Google Play: [сайт]. [2016]. URL: <https://play.google.com/store/apps/details?id=com.merigotech.fillwords&hl=ru>
3. Филворды: найди слова [электронный ресурс] // iTunes [сайт]. [2016]. URL: <https://itunes.apple.com/ua/app/filvordy-najdi-slova/id849654325?mt=8>
4. Алгоритм генерации кроссворда [электронный ресурс] // SQL.ru [сайт]. [2000-2016]. URL: <http://www.sql.ru/forum/266381/algoritm-generacii-krossvorda>

УДК 004.7

Проектирование архитектуры централизованного кроссплатформенного протокола сетевой печати

Тырса, С.В. Хубеджев Д.П., Моргайлов Д.Д.

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии

tyrsastanislav@icloud.com, hubedjev.d.p@gmail.com, kolleganin@yandex.ru

Аннотация

Тырса С.В., Хубеджев Д.П., Моргайлов Д.Д. Проектирование архитектуры централизованного кроссплатформенного протокола сетевой печати. Выполнен анализ возможностей современных протоколов сетевой печати, выявлены их недостатки и ограничения. Выделены основные требования к функциональным характеристикам протоколов сетевой печати. Предложена архитектура централизованного кроссплатформенного протокола сетевой печати.

Ключевые слова: сетевая печать, протокол удаленной печати, централизованная архитектура, кроссплатформенный протокол.

Постановка проблемы. Сетевой протокол печати представляет собой набор правил и действий, которые позволяют выполнить удаленную печать документа.

На данный момент большинство протоколов удаленной печати не удовлетворяют требованиям кроссплатформенности. Зачастую они привязаны к конкретной операционной системе и не позволяют выполнять печать документов тех форматов, которые не поддерживаются операционной системой. Кроме того, большая часть современных протоколов печати реализует доступ к сетевому принтеру исключительно по его адресу, не обеспечивая при этом агрегацию компьютеров в единую виртуальную сеть для предоставления пользователю информацию о всех доступных принтерах, подключенных к ним.

Цель статьи – анализ возможностей современных протоколов сетевой печати и разработка архитектуры централизованного кроссплатформенного протокола печати.

Анализ современных протоколов печати. Наиболее популярными протоколами сетевой печати являются Microsoft Network Printing, Internet Printing Protocol, FTP Printing, SMTP/POP3 Printing и другие [1].

Протокол Microsoft Network Printing (MNP) является стандартным для операционных систем семейства Windows [1]. Он основан на универсальных протоколах SMB (англ. Server Message Block) и RPC (англ. Remote Procedure Call). Главными достоинствами MNP являются полная поддержка всех возможностей операционных систем Windows и простота в использовании. Для корректной работы протокола необходимо рассылать драйвера принтера на все устройства в данной сети, что делает печать невозможной в случае, когда в сети участвуют устройства, работающие под операционными системами семейства GNU/Linux или Apple Mac OS/OS X. Ещё одним недостатком данной системы является неполноценная работа с IPP (англ. Internet Printing Protocol), что могло бы решить большинство вышеперечисленных проблем.

Internet Printing Protocol (IPP) [2] является переосмысленной версией протокола передачи гипертекста HTTP. Использование IPP существенно упрощает установку принтера на локальной клиентской машине: достаточно указать URL/printers/ и выбрать предпочтительный для клиента принтер из списка. Проблема использования протокола заключается в том, что принтеры, по сути, не собраны в сети, а пользователю приходится явно указывать IP-адреса устройств с подключенным принтером. К тому же, для Windows-систем потребуется установка дополнительного программного обеспечения для работы с протоколом.

Протокол передачи файлов File Transfer Protocol (FTP) [3] адаптируется для нужд печати: каждый принтер, подключённый к серверу, представляется отдельным каталогом, либо существует только корневой каталог, если у сервера всего один порт для принтера. Данный протокол имеет те же недостатки, что и IPP.

Печать посредством протоколов SMTP [4] и POP3 [5] аналогична печати через FTP, только обмен происходит не файлами, а почтовыми сообщениями с вложением файлов. В режиме SMTP сервер печати выглядит как почтовый сервер, принимая подключения и извлекая все файлы из сообщений, чтобы отправить их на принтер. Выбор принтера происходит по имени почтового ящика адресата. В режиме POP3 сервер печати выглядит как почтовый клиент, периодически опрашивая заданный почтовый сервер о наличии новых сообщений, и вы отправляете задания именно на почтовый сервер, но не на сервер печати.

Проектирование архитектуры централизованного протокола печати. Разрабатываемый протокол имеет клиент-серверную архитектуру, в

которой присутствует выделенный центральный сервер, отвечающий за хранение адресов клиентов и общего списка принтеров, а также прием и отправку запросов печати (рис. 1).

Для каждого клиента создается прослушивающий поток, который отлавливает и обрабатывает команды клиента. Клиент, в свою очередь, работает по схеме, показанной на рис. 2.

При старте клиентской программы создается управляющее соединение с центральным сервером, по которому отправляется информация о списке принтеров, подключенных к клиентскому компьютеру. Сервер формирует общий список сетевых принтеров и отправляет их обратно на клиент.

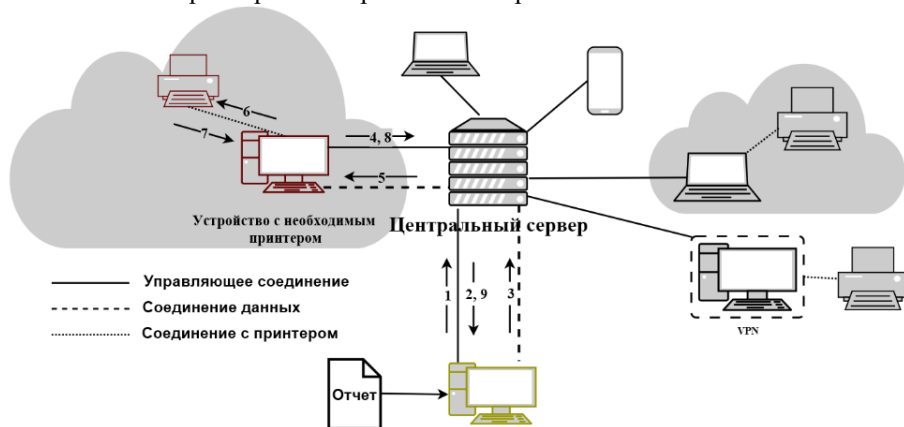


Рис. 1 – Архитектура разработанного протокола печати

(1 – запрос на подключение, отправка списка локальных принтеров, проверка очереди печати, 2 – получение списка сетевых принтеров, 3 – отправка файла на печать, 4 – проверка очереди печати, 5 – прием файла, 6 – печать файла, 7 – отправка кода состояния принтера, 8 – отправка состояния печати в очередь печати на сервере, 9 – отправка результата печати на клиент)

Для передачи файлов на печать создается соединение данных с центральным сервером. На сервере формируются очереди печати для каждого принтера, периодически опрашиваемые клиентскими программами на наличие задач печати.

Клиенты взаимодействуют с сервером посредством кодов команд. Первый байт является уникальным кодом команды. Следующие байты в зависимости от команды могут означать имена/адреса клиентов-приемников или параметры команды, если они требуются.

Центральный сервер обрабатывает запросы печати в асинхронном режиме с использованием пула потоков обслуживания, тем самым увеличивается

скорость его работы и не позволяя «медленным» запросам надолго заблокировать процесс обработки.

В основе разрабатываемого протокола печати лежит протокол TCP (рис. 3). Посредством протокола TCP осуществляется мониторинг состояния принтеров в сети, отправка и приём файлов печати.

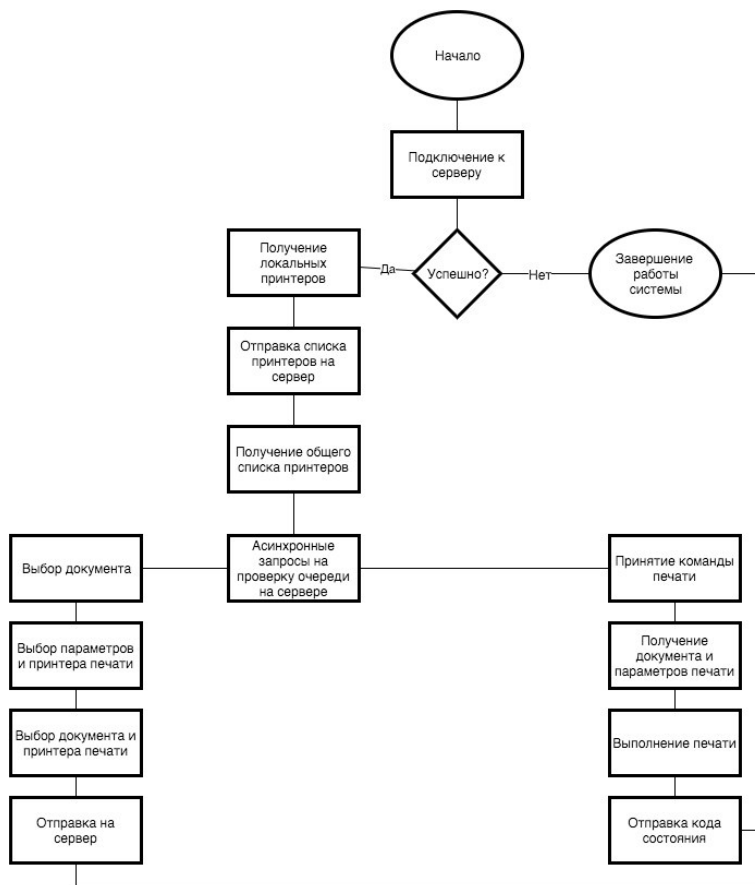


Рис. 2 – Схема работы клиентской части протокола

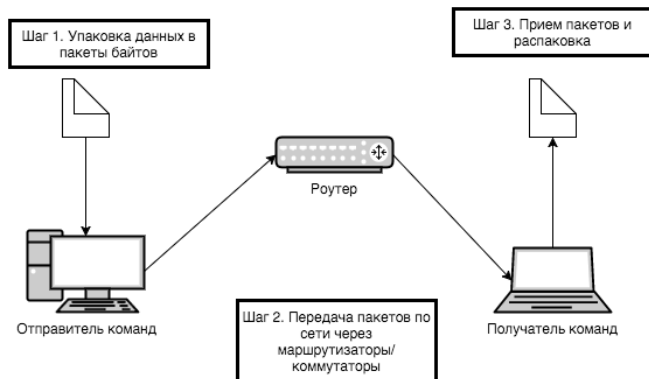


Рис. 3 – Схема работы протокола TCP

Для достижения кроссплатформенности протокола используются конвертеры. Это позволило печатать документы «неродных» для операционной системы клиента форматов. Конвертация документа осуществляется в формат .pdf, который является одним из базовых для операционных систем Windows, GNU/Linux, Apple Mac OS/OS X. Применяемые средства конвертации поддерживает форматы документов Microsoft Office, Apache OpenOffice, Apple iWork.

В качестве конвертеров используются стандартные возможности вышеперечисленных программных пакетов, поскольку такой подход гарантирует наиболее точное преобразование. Таким образом, конвертация документов происходит без потери форматирования.

Конвертация выполняется посредством утилит VBScript (MS Office for Windows), AppleScript (MS Office, iWork for Mac) и библиотеки JOD Converter (AOO Windows/OS X).

Выводы. В рамках данной работы был выполнен анализ современных протоколов удаленной печати, спроектирован кроссплатформенный централизованный протокол удаленной печати документов.

В качестве направления дальнейшей работы планируется разработка распределенной архитектуры протокола с несколькими серверами, расположенными в разных сегментах сети, а также системы «умного» поиска потенциального сервера в сети. Кроме того, планируется дополнить протокол поддержкой веб-интерфейса и популярных мобильных платформ Google Android и Apple iOS.

Список литературы

1. Протоколы сетевой печати [электронный ресурс] // More PC: [информационно-справочный портал]. [2006]. URL: http://www.morepc.ru/net/print/printing_protocols.html

2. RFC 2910. Internet Printing Protocol/1.1: Encoding and Transport [текст] / Network Working Group; R. Herriot; S. Butler; P. Moore; P. Turner; J. Wenn. [сентябрь, 2003]. – URL: <https://tools.ietf.org/html/rfc2910>
3. RFC 448. Print files in FTP [текст] / Network Working Group; R.T. Braden. [февраль, 1973]. – URL: <https://tools.ietf.org/html/rfc448>
4. RFC 5321. Simple Mail Transfer Protocol [текст] / Network Working Group; J. Klensin. [октябрь, 2008]. – URL: <https://tools.ietf.org/html/rfc5321>
5. RFC 1939. Post Office Protocol - Version 3 [текст] / Network Working Group; J. Myers; M. Rose; [май, 1996]. – URL: <https://tools.ietf.org/html/rfc1939>

УДК 004.934.2

Автоматизация исследований в области сегментации и распознавания речи Немцов Д.И.

Государственное учреждение «Институт проблем искусственного интеллекта»
отдел распознавания речевых образов
E-mail: dimanemcov@mail.ru

Аннотация

Немцов Д.И. Автоматизация исследований в области сегментации и распознавания речи. В работе предложены средства автоматизации исследований в области сегментации и распознавания речи. За счет их использования удалось сократить время подготовки и разметки аудиоданных, обеспечить повторяемость результатов экспериментов и возможность детального анализа их условий.

Общая постановка проблемы. На эффективность работ, в том числе исследовательских, влияет уровень их автоматизации. В распознавании речи целесообразно автоматизировать сбор обучающих и тестовых данных и оценку разработанных методов. Это сократит время и трудозатраты, а так же повысит эффективность исследований за счёт сохранения речевого материала, что обеспечит повторяемость результата и при необходимости возможность детального анализа условий экспериментов.

Цель работы – создание средств автоматизации исследований в области сегментации и распознавания речи. Предполагается создать средства автоматизации: сбора аудиоданных, разметки аудиоданных, построения эталонов пар аллофонов по указанному набору размеченных аудиоданных.

Предлагаемые средства автоматизации и результаты их применения. Без применения средств автоматизации для обучения системы фонемного распознавания необходимо было записать или загрузить звуковой файл, после

чего в окне отображения сигнала разметить его: указать границы и маркировать (присвоить символьные обозначения) сегменты. Далее дать команду на построение эталонов по парам маркированных сегментов.

С точки зрения функциональности, процесс обучения распадается на три шага: 1) запись wav-файла; 2) задание меток, определяющих границы и символьные обозначение сегментов; 3) построение и добавление эталонов.

Каждый из этих шагов можно автоматизировать.

Средства автоматизированного сбора аудиоданных должны позволять:

- 1) быстро формировать перечень слов, для произнесения диктором;
- 2) предусматривать удобные средства для подсказки диктору какое именно слово надо произнести и где в слове должно находиться ударение;
- 3) обеспечивать возможность контроля пользователем правильности речи и корректности ее записи системой;
- 4) предоставлять пользователю возможность повторного произнесения ошибочно произнесенного или некорректно записанного слова;
- 5) сохранять произнесение каждого слова в отдельном файле, имя которого отражает написание слова с ударением.

Такой способ именования файлов необходим для автоматизации процесса обучения и оценки средств пословного и фонемного распознавания [2].

Для автоматизации разметки набора речевых сигналов используем средства сегментации [1] речи, сохранённой в файле. В выбранном пользователем каталоге с аудиоданными для каждого wav-файла программно вызываем функцию автоматической сегментации, она сохраняет результаты сегментации в файл с расширением «.lab», имя которого совпадает с именем загруженного файла.

Эта разметка указывает границы и маркировку сегментов, соответствующих аллофонам. Такая разметка нужна для создания обучающего набора при фонемном распознавании. Для тестовых наборов и обучающего набора при пословном распознавании необходимо знать границы речи. Данную информацию можно автоматически получить из исходной разметки путём использования левой границы первого и правой границы последнего сегмента.

Используемый в работе способ именования файлов позволяет автоматически выявлять некоторые случаи некорректной сегментации. Их можно выявить по несоответствию транскрипции слова, формируемой по имени файла, и количеству или маркировке сегментов. Так как подсистема автоматической сегментации в данный момент находится на стадии тестирования, результаты её работы подлежат визуальной проверке и, при необходимости, корректируются.

Для автоматической оценки средств пословного распознавания описанным способом собираем материал как для обучения, так и для тестирования. Разметка аудиоданных содержит границы речи. Здесь при обучении из имени файла формируется имя эталона. При тестировании по имени распознаваемого файла определяется имя эталона, который должен быть

результатом. Это даёт возможность автоматически определять, правильно ли распознано слово, и если нет, то какую позицию в ранжированном списке результатов распознавания оно занимает.

Для автоматического обучения системы пофонемного распознавания набор эталонов формируем по размеченным аудиоданным из выбранного пользователем каталога. Для каждого wav-файла этого каталога выполняем следующие действия:

- 1) загружаем wav-файл и соответствующий lab-файл;
- 2) для пар контактно расположенных сегментов строим параметрическое представление, что соответствует эталону пары аллофонов;
- 3) добавляем построенные эталоны пар аллофонов в систему.

Автоматическая оценка средств пофонемного распознавания. При пофонемном распознавании словарь представляется в виде множества транскрипций слов, получаемых по их написанию с ударением. Если для тестирования такой системы необходимо создать словарь, совпадающий с тестовым набором, мы можем сегментировать его из имён wav-файлов этого набора. В процессе тестирования по имени файла можно определить транскрипцию распознаваемого слова, это даст нам возможности аналогичные описанным для пословного распознавания.

Выводы. В работе предложены средства автоматизации исследований в области сегментации и распознавания речи. За счет их использования удалось сократить время подготовки и разметки аудиоданных, обеспечить повторяемость результатов экспериментов и возможность детального анализа их условий.

Список литературы

1. Бурибаева А.К., Дорохина Г.В., Ниценко А.В., Шелепов В.Ю. Сегментация и дифонное распознавание речевых сигналов // Труды СПИИРАН. – 2013. – №31. – С. 20–42.
2. Шелепов В.Ю., Ниценко А.В. К проблеме распознавания слитной речи // Искусственный интеллект. – 2012. – № 4. – С. 272-281.

УДК 004.657

Программная система учета научно-технических мероприятий ДонНТУ
Ходич Д.В., Грищенко В.И.

Донецкий национальный технический университет,
кафедра программной инженерии
dima_khodich@mail.ru, victor.grischenko@gmail.com

Аннотация

Ходич Д.В., Грищенко В.И. Программная система учета НТМ ДонНТУ. В тексте данной статьи описан процесс работы программной системы и средства, которыми она разрабатывается. Исходя из анализа существующих программных систем учета НТМ, была предложена разработка системы учета НТМ для ДонНТУ открытым кодом, реализующая генерирование отчетности, а также рассылки по адресам электронной почты, которые записаны в БД.

Общая постановка проблемы. За многие годы в ДонНТУ было проведено множество различных научно-технических мероприятий (НТМ). Учитывая то, что по каждому из мероприятий нужно оформить большое количество подготовительных документов (на начальном этапе) и множество отчетов после проведения мероприятия, в архивах нашего университета накапливалось большое количество информации в бумажном виде.

Все эти документы время от времени приходится просматривать для подготовки отчетов за определенный период времени или годовых отчетов. Чтобы облегчить работу ответственным за проведение мероприятий было решено создать автоматизированную систему составления приказов и оформления отчетов.

Существующие программные системы учета НТМ. 1) "Учет мероприятий и семинаров" - предназначена для комплексной автоматизации компаний, занимающихся организацией семинаров, тренингов, форумов, конференций и других деловых мероприятий. Программа позволяет вести базы данных клиентов компании, мероприятий, лекторов и авторов, участвующих в мероприятиях, а также темы их выступлений. Данная программа легко и быстро настраивается под конкретные требования заказчика. Основные функции: - учет клиентов-организаций, которые участвуют в мероприятиях; - ведение базы контактов клиентов, сотрудников организаций; - учет прошедших и будущих мероприятий; - ведение календарного плана (расписание) мероприятий; - учет посещений того или иного мероприятия; - ведение базы участников мероприятий (авторов и лекторов); - ведение базы тем выступлений лекторов; - учет договоров, заключенных с клиентами; - учет сотрудников, их логинов и паролей для доступа

в программу; - минимизация ошибок ввода, уменьшение времени на обработку заявки; - возможность импорта и экспорта данных [1].

2) Программа контроля и управления мероприятиями включает журнал мероприятий по контролю каждого мероприятия, праздника и события. Контроль реализации мероприятий включает учет дохода от праздника или события. Управление организацией мероприятий обеспечивает учет всех расходов и расчет прибыли. Поднять престиж и осуществить все самые недостижимые цели вы сможете, используя учет управленческий. Автоматизированное управление технологическими процессами позволит контролировать все рабочие моменты [2].

Проектирование собственной программной системы учета НТМ

В университете ежегодно проводятся следующие виды научно-технических мероприятий НТМ:

- конференция;
- семинар;
- круглый стол;
- форум;
- симпозиум;
- олимпиада;
- конкурс;
- день науки;
- фестиваль;
- выставка;
- презентация.

Система учета мероприятий ДонНТУ является информационной системой, которая в свою очередь направлена на:

- уменьшение рутинной работы;
- сокращение времени на оформление отчетов и приказов;
- уменьшение числа возможных ошибок;
- резкое сокращение времени для обработки данных и получении нужной отчетности.

Для решения поставленной цели необходимо осуществить: - автоматизацию ввода, контроля и загрузки данных в базу НИЧ ДонНТУ;

- автоматизацию регистрации на сайте конференций;
- формирование отчетности по заданным конференциям;

Существующие аналоги программных систем учета мероприятий не подходят для ДонНТУ, исходя из того, что они не учитывают специфику ВУЗа. Цели, которые ставят перед собой аналогичные системы, не соответствуют целям нашего университета. Также база данных для учета мероприятий ДонНТУ будет иметь значительно больший объем, что поможет в дальнейшем составлять нужную отчетность. Поэтому было принято решение о разработке своей, более узкоспециализированной программной системы. Для разработки

информационной системы учета НТМ потребуется использование персональных машин, которые будут ориентированы на конечного пользователя, а также администратора. Создание системы автоматизации позволит руководителю мероприятия производить оформление документации быстрее и эффективнее.

В ходе реализации программной системы возникла необходимость создания нескольких модулей:

- модуль регистрации пользователей в системе;
- модуль проверки наличия учетной записи пользователя в системе;
- модуль отправки активации учетной записи пользователя;

Система включает в себя около 25 таблиц и справочников. Для реализации системы необходима установка и настройка Apache сервера, php, СУБД mysql, phpmyadmin. На web-сервере располагается база данных для мероприятий.

Для создания базы данных использовалась утилита phpmyadmin, которая располагается на сервере. Система для учета участия в мероприятиях состоит из двух частей: серверной и клиентской. Скрипты, расположенные на сервере, предназначены для получения данных и для обработки запросов (добавления данных, обновления данных, для удаления данных и для редактирования). [3][4].

Проектирование системы. На данном этапе спроектирована база данных и созданы таблицы. Разработаны все php скрипты для удаления, добавления и редактирования данных.

Также разработаны практически все экранные формы приложения. Главная форма представлена на рис. 2.

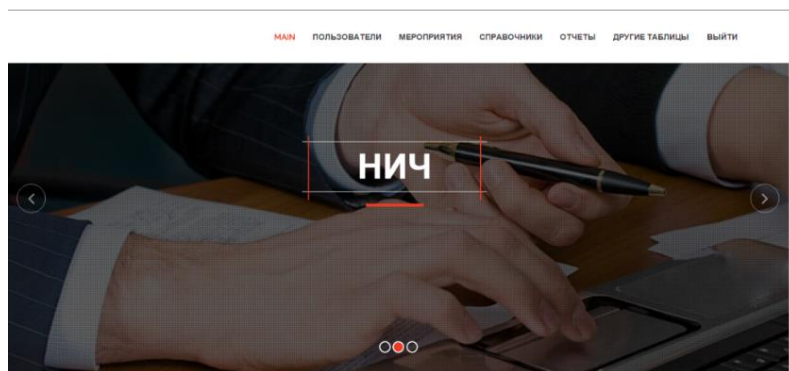


Рис. 2. Главная форма

Настроена библиотека phrword для генерации отчетности в формате docx.

Порядок работы с программной системой. Начальник НИЧ заполняет сведения о заместителях декана по науке факультетов ДонНТУ: ФИО, должность, контактные телефон и эл. почта до 1 декабря текущего года.

1 декабря осуществляется напоминающая рассылка на адрес эл. почты заместителям декана по науке о необходимости подтвердить авторизацию для новых пользователей и внести сведения в факультетский план НТМ на следующий календарный год до 20 декабря.

С 20 по 31 декабря начальник НИЧ проверяет корректность ввода сведений о НТМ в общем плане мероприятий. 31 декабря, проверенный план НТМ получает статус утвержденного, после подписания печатного варианта проректором по научной работе ДонНТУ.

10 января осуществляется рассылка на адрес эл. почты ответственных секретарей оргкомитетов о необходимости подтвердить авторизацию для новых пользователей и необходимости ознакомления с порядком работы с системой (внесения сведений, подготовки приказа и отчета и т.д.).

За 20 дней до даты проведения мероприятия осуществляется рассылка на адрес эл. почты ответственного секретаря соответствующего оргкомитета о необходимости подготовки приказа, о возможности внести окончательные изменения в сведения о мероприятии, включая его отмену.

За 10 дней до даты проведения мероприятия осуществляется рассылка на адрес эл. почты ответственного секретаря соответствующего оргкомитета о необходимости публикации пресс-релиза и размещения программы (положения) мероприятия.

Через 5 дней после даты проведения мероприятия осуществляется рассылка на адрес эл. почты ответственного секретаря соответствующего оргкомитета о необходимости публикации пост-релиза о проведенном мероприятии.

Через 10 дней после даты проведения мероприятия осуществляется рассылка на адрес эл. почты ответственного секретаря соответствующего оргкомитета о необходимости сдачи отчета о проведенном мероприятии.

В случае отсутствия сведений о подготовке приказа (за 17 дней), публикации пресс-релиза (за 8 дней), программы или положения (за 8 дней), пост-релиза (через 7 дней), отчета (через 13 дней) начальнику НИЧ приходит уведомление на эл. почту.

По требованию начальника НИЧ формируется отчет о проведенных мероприятиях за месяц, квартал, полугодие, год или за выбранный период времени. На основе введенных данных выполняется расчет показателей НТМ.

Выводы. Разрабатываемая программная система позволит упростить работу с мероприятиями, проводимыми в ДонНТУ. Значительно ускорит процесс составления документации по данным мероприятиям. Повысит качество составляемой отчетности. Позволит ответственным за проведение конкретного мероприятия, т. е. заместителям деканов по науке на каждом факультете, формировать документацию в автоматизированном виде, что также позволит избежать множества ошибок.

Список литературы

1. Конфигурация "Учет мероприятий и семинаров"//Простой софт программы для дома и офиса. [Электронный ресурс]. – Режим доступа: <http://prostoysoft.ru/Events.htm>
2. Учет мероприятий // Автоматизация работы компании по организации и проведению мероприятий. [Электронный ресурс]. – Режим доступа: http://usu.kz/uchet_meropriyatiy.php
3. Осипов, Д. Л. InterBase и Delphi. Клиент-серверные базы данных / Д.Л. Осипов. - М.: ДМК Пресс, 2015. - 536 с.
4. Растрингин, Л.А. Адаптация сложных систем. Методы и приложения / Л.А. Растрингин. - Москва: РГГУ, 1981. - 501 с.

УДК 004.056

Защита web-ресурса оценки и ранжирования стран мира по показателям их развития

Багимова А.Д., Губенко Н.Е.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
E-mail: anja_positiv@mail.ru

Аннотация

Багимова А. Д., Губенко Н. Е. Защита web-ресурса оценки и ранжирования стран мира по показателям их развития. Рассмотрены проблемы защиты web-ресурса оценки и ранжирования стран мира по показателям их развития. Изложена актуальность проблемы защиты информации и структура web-ресурса. Проанализированы существующие угрозы безопасности и выявлены пути решения вопросов защиты.

Ключевые слова: web-ресурс, показатели, база данных, сервер, функциональные модули, ранжирование, пользователь, методы, технологии.

Постановка проблемы. Проблема защиты информации на настоящем этапе развития информационных телекоммуникационных технологий является безусловно актуальной и определяется рядом объективных и субъективных факторов, среди которых можно отметить наиболее существенные:

- обострение противоречий между объективно существующими потребностями общества в расширении свободного обмена информацией и чрезмерными или наоборот недостаточными ограничениями на ее распространение и использование;

- расширение сферы использования компьютеров, многообразием и повсеместным распространением информационно-управляющих систем, высокими темпами роста и развития парка средств вычислительной техники и связи;
- повсеместное использование локальных и глобальных сетевых технологий людьми и организациями, не подготовленными в сфере защиты информации;
- концентрация больших объемов информации различного назначения и принадлежности на электронных носителях;
- отношение к информации, как к товару, переход к рыночным отношениям в области предоставления информационных услуг с присущей им конкуренцией и промышленным шпионажем;
- многообразие видов угроз и возникновение новых возможных каналов несанкционированного доступа к информации
- увеличение потерь (ущерба) от уничтожения, фальсификации, разглашения или незаконного тиражирования информации (возрастанием уязвимости различных затрагиваемых субъектов)
- развитие рыночных отношений (в области разработки, поставки, обслуживания вычислительной техники, разработки программных средств, в том числе средств защиты).

Цель статьи - защитить web-ресурс для оценки и ранжирования стран мира по показателям их развития, проанализировав существующие угрозы безопасности и выявить пути решения вопросов защиты.

Структура web-ресурса. Функциональные модули web-системы для оценки и ранжирования стран мира по показателям их развития разработаны с помощью языка программирования PHP и базы данных MySQL. Скрипты и база данных сохраняется на отдалённом сервере. Доступ к которому осуществляется через протокол http. Защита от угрозы нарушения конфиденциальности осуществляется с помощью традиционной авторизации. Для регулирования прав доступа используется модель политики безопасности. Система имеет три роли – администратор, модератор и незарегистрированный пользователь. Администратор имеет возможность добавлять пользователей с существующими ролями.

Данная система реализована Web-приложениями, ее структура представлена на рисунке 1.

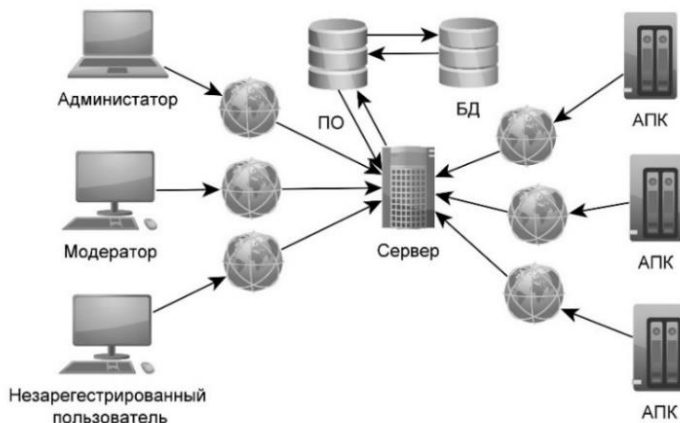


Рис. 1 - Структура web-ресурса

Угроза безопасности web-ресурса. Угроза информационной безопасности web-ресурса - это возможность воздействия на информацию, обрабатываемую в системе, приводящего к искажению, уничтожению, копированию, блокированию доступа к информации, а также возможность воздействия на компоненты информационной системы, приводящего к нарушению секретности, утрате, уничтожению или сбою функционирования носителя. Реализация любого из этих факторов может привести к нарушению конфиденциальности, целостности информации или средства управления программно-аппаратным комплексом системы и, как следствие, к потере основных функциональных свойств системы.

Угроза нарушения конфиденциальности данных включает в себя любое умышленное или случайное раскрытие информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую. К нарушению конфиденциальности ведет как умышленное действие, направленное на реализацию несанкционированного доступа к данным, так и случайная ошибка программного или неквалифицированного действия оператора, приведшая к передаче по открытым каналам связи незащищенной конфиденциальной информации.

Угроза нарушения целостности включает в себя любое умышленное или случайное изменение информации, обрабатываемой в информационной системе или вводимой из первичного источника данных. К нарушению целостности данных может привести как преднамеренное деструктивное действие некоторого лица, изменяющего данные для достижения собственных целей, так и случайная

ошибка программного или аппаратного обеспечения, приведшая к безвозвратному разрушению данных.

Анализ угроз безопасности. Первый шаг в анализе угроз — их идентификация. Рассматриваемые виды угроз следует выбирать исходя из соображений здравого смысла, но в пределах выбранных видов провести максимально подробный анализ.

Необходимо не только провести работу по выявлению и анализу самих угроз, но и изучить и описать источники возникновения выявленных угроз. Такой подход поможет в выборе комплекса средств защиты. Например, нелегальный вход в систему может стать следствием воспроизведения начального диалога, подбора пароля или подключения к сети неавторизованного оборудования. Очевидно, для противодействия каждому из перечисленных способов нелегального входа нужны свои механизмы безопасности.

Средства защиты web-ресурса. Существуют различные методы защиты баз данных. Они условно делятся на две группы: основные и дополнительные.

К основным средствам защиты относится: защита паролем; шифрование данных и программ; разграничение прав доступа к объектам базы данных;

К дополнительным средствам защиты БД можно отнести такие, которые нельзя прямо отнести к средствам защиты, но которые непосредственно влияют на безопасность данных. Их составляют следующие средства:

- встроенные средства контроля значений данных в соответствии с типами;
- повышение доверенности вводимых данных;
- обеспечение целостности связей таблиц;
- организации совместного использования объектов БД в сети.

В разрабатываемом ресурсе используется защита паролем (для регулирования прав доступа рядового пользователя и администратора) и разграничение прав доступа к объектам базы данных.

Обеспечения информационной безопасности базы данных нашей web-системы, основывается на двух фундаментальных принципа: проверка полномочий и проверка подлинности (аутентификация).

Проверка полномочий основана на том, что каждому пользователю или процессу информационной системы соответствует набор действий, которые он может выполнять по отношению к определенным объектам. Проверка подлинности означает достоверное подтверждение того, что пользователь или процесс, пытающийся выполнить санкционированное действие, действительно тот, за кого он себя выдает.

Система назначения полномочий имеет в некотором роде иерархический характер. Самыми высокими правами и полномочиями обладает системный администратор или администратор сервера БД. Традиционно только этот тип пользователей может создавать других пользователей и наделять их определенными полномочиями.

СУБД в своих системных каталогах хранит как описание самих пользователей, так и описание их привилегий по отношению ко всем объектам.

В нашей СУБД вводится следующий уровень иерархии пользователей — это администратор БД. В системной СУБД один сервер MS SQL Server может управлять множеством СУБД. В стандарте SQL не определена команда создания пользователя, но практически во всех коммерческих СУБД создать пользователя можно не только в интерактивном режиме, но и программно с использованием специальных хранимых процедур. Однако для выполнения этой операции пользователь должен иметь право на запуск соответствующей системной процедуры.

Защита паролем представляет собой простой и эффективный способ защиты БД от несанкционированного доступа. Пароли устанавливаются пользователями или администраторами БД. Учет и хранение паролей выполняется самой СУБД. Обычно, пароли хранятся в определенных системных файлах СУБД в зашифрованном виде. После ввода пароля пользователю СУБД предоставляются все возможности по работе с БД.

Парольная защита является достаточно слабым средством, особенно если пароль не шифруется. Основной ее недостаток состоит в том, что все пользователи, использующие одинаковый пароль, с точки зрения вычислительной системы неразличимы. Неудобство парольной системы для пользователя состоит в том, что пароль надо запоминать или записать. При небрежном отношении к записям пароль может стать достоянием других.

В целях контроля использования основных ресурсов СУБД во многих системах имеются средства установления прав доступа к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта, а также администратор БД имеют все права. Остальные пользователи к разным объектам могут иметь различные уровни доступа. Разрешение на доступ к конкретным объектам базы данных сохраняется в файле рабочей группы.

Наибольший эффект с точки зрения защищенности и производительности системы достигается только при реализации элементов безопасности во время проектирования и разработки системы в целом и каждого отдельного её элемента, поэтому дополнительно необходимо решить вопрос преобразования форматов существующих и накопленных в мире баз данных показателей развития разных стран и регионов в формат My SQL без нарушения их достоверности и целостности.

Выводы. Рассмотрены проблемы защиты web-ресурса для оценки и ранжирования стран мира по показателям их развития. Изложена актуальность проблемы защиты информации и цель статьи. Изучена структура web-ресурса. Проанализированы существующие угрозы безопасности и выявлены пути решения вопросов защиты.

Список литературы

1. Защита информации в базах данных [электронный ресурс] // Казахстанский университет [сайт]. [2014-2016]. URL: <http://bourabai.ru/dbt/dbms/13.htm>
2. Особенности защиты информации в базах данных [электронный ресурс] // Zavgorodnii Design: [сайт]. [2012-2014]. URL: <http://sumk.ulstu.ru/docs/mszki/Zavgorodnii/11.7.html>

УДК 004.4

Разработка интегрированной CASE-системы для обучения студентов программированию

Воробьев Л.О., Полетаев В.А., Моргайлов Д.Д.

Донецкий национальный технический университет, г. Донецк
кафедра программной инженерии

lev4411@gmail.com, poletaevvladislav@gmail.com, kolleganin@yandex.ru

Аннотация

Воробьев Л.О., Полетаев В.А., Моргайлов Д.Д. Разработка интегрированной CASE-системы для обучения студентов программированию. Выполнен сравнительный анализ возможностей современных CASE-средств, выявлены их основные ограничения. Обоснована актуальность создания интегрированной CASE-системы с поддержкой генерации алгоритмического программного кода. Приведено описание внутреннего устройства разработанной системы и результатов ее работы.

Ключевые слова: CASE-система, язык UML, проектирование, генерация исходного кода, обучение студентов.

Постановка проблемы. Сложность и объемы разрабатываемых программных систем с каждым годом неуклонно повышаются. Усложнение процессов анализа, проектирования, разработки, тестирования и сопровождения программных продуктов приводит к необходимости увеличения числа участников проекта и ужесточению требований к качеству и срокам выхода продукта на рынок [1].

Для решения этой проблемы создаются автоматизированные программные инструменты (CASE-системы), главным назначением которых является сокращение трудозатрат на выполнение рутинных операций в процессе жизненного цикла разработки программного обеспечения, улучшение качества программных продуктов, поддержание унифицированного и согласованного стиля работы [1, 2].

На данный момент разработано множество различных CASE-инструментов, облегчающих процесс проектирования и документирования программных систем. Они поддерживают построение на специализированном формальном языке набора визуальных диаграмм и спецификаций, описывающих структурные и поведенческие аспекты проектируемого продукта.

Наиболее популярными являются системы IBM Rational Rose, Paradigm Plus, BPWin, Sybase PowerDesigner, Silverrun и прочие [3, 4]. Большинство из них ориентированы на применение унифицированного языка моделирования UML как основного модельного аппарата CASE-технологий при объектно-ориентированном подходе [5].

Однако главным ограничением таких систем является узкие возможности генерации исходного кода, поскольку генерация объектно-ориентированного кода по статическим моделям программы (например, по диаграммам классов и компонентов языка UML) дает на выходе исключительно каркас («скелет») программного продукта, представляющий собой набор программных модулей (файлов исходного кода). Модули каркаса состоят из сгенерированного объявления классов с полями, сигнатурами и «заглушками» методов, но не содержат при этом алгоритмического кода методов класса.

Выявленное ограничение является обоснованным при создании реальных программных продуктов и объясняется многообразием современных средств разработки: языков, технологий, фреймворков, платформ, – и их непрерывным развитием. Однако в случае, если объектом разработки является учебный программный проект, использующий только стандартный набор библиотек языка, такие ограничения можно обойти, реализовав поддержку генерации исполняемого кода по графической записи алгоритма.

Кроме того, использование интегрированной CASE-системы с возможностью генерации программного кода для решения учебных задач позволит улучшить понимание студентами взаимосвязей между различными моделями представления программной системы, а также между моделями системы и ее исходными кодами.

Цель работы – разработка интегрированного CASE-средства для повышения эффективности обучения студентов программированию.

Разработка интегрированной CASE-системы. На языке программирования Java была разработана программная система, позволяющая создавать логическую модель программного продукта, а также описывать реализацию методов классов в виде графической модели алгоритма.

Логическая модель программной системы представлена в виде UML-диаграммы классов. При этом, в отличие от других CASE-систем на диаграмме отображаются связи между сущностями, представленными на диаграмме в зависимости от указанных пользователем типов свойств, методов, аргументов методов. Это позволяет оптимизировать время программиста, направленное не на

разработку визуальной составляющей диаграммы, а на разработку самой модели (рис. 1).

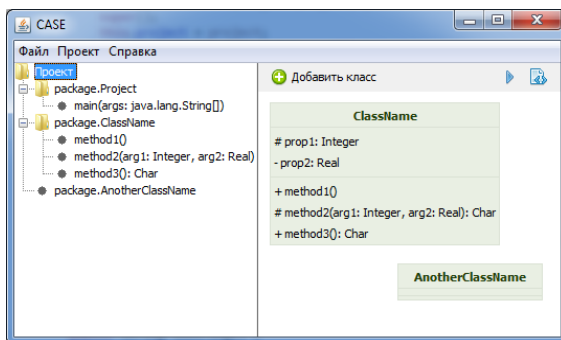


Рис. 1 – Пример UML-диаграммы, спроектированной с использованием разработанной системы

Другой важной особенностью разрабатываемой системы является возможность создания функциональной модели работы программы – описания алгоритмов – реализации методов классов, определенной в UML-диаграмме классов. Для редактирования алгоритма разработанная система предоставляет пользователю интерфейс типа Drag&Drop для манипулирования операторами программы и отношениями между ними (рис. 2).

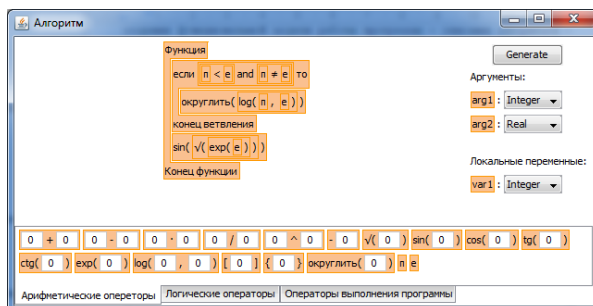
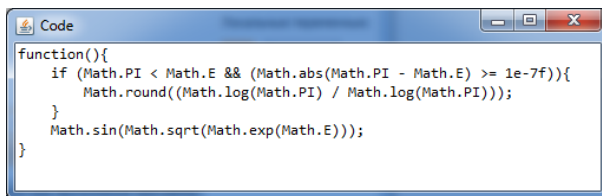


Рис. 2 – Пример алгоритма, спроектированного с использованием разработанной системы

С помощью данного интерфейса пользователь может изменять реализацию методов и генерировать на их основе программный код. Например, для изображенного на рис. 2 алгоритма код будет иметь вид, показанный на рис. 3.



```
function(){
  if (Math.PI < Math.E && (Math.abs(Math.PI - Math.E) >= 1e-7f)){
    Math.round((Math.log(Math.PI) / Math.log(Math.PI)));
  }
  Math.sin(Math.sqrt(Math.exp(Math.E)));
}
```

Рис. 3 – Пример сгенерированного программного кода

Модель представления данных условно разделяется на модель представления алгоритмов функций и модель представления UML-диаграммы классов. Для взаимодействия с графической частью используются механизм обработки событий. События необходимы для корректного отображения UML-диаграммы, поскольку изменение одного элемента (например, типа аргумента) может изменить схему связей между классами.

В модели представления UML диаграммы широко используется Java Reflection API, который позволяет с легкостью использовать существующие классы в разрабатываемой UML-диаграмме. Элементы языка программирования (классы, методы, члены классов и локальные переменные) могут быть преобразованы из программного кода и выведены на экран, благодаря использованию этой библиотеки.

На рис. 4 изображена UML-диаграмма классов, которые реализуют внутреннее представление проектируемой диаграммы классов. Основным элементом диаграммы является класс UMLClass, который содержит в себе поля, методы и вложенные классы, которые также могут содержать вложенные члены. Класс для представления классов на UML-диаграмме предоставляет список полей, методов и вложенных классов.

Алгоритмы составляются из блоков, которые обозначают действия или могут содержать коллекцию действий. Каждое такое действие может иметь тип возвращаемого значения, или может содержать набор действий. Эти действия называются операторами. Для представления всего разнообразия операторов используется библиотека JSON, которая позволяет хранить набор значений в текстовых файлах. Таким образом, набор операторов может быть адаптирован для любого языка программирования путем изменения структуры текстового файла в формате JSON.

За реализацию методов отвечает класс Method, который хранит список параметров, локальных переменных и тип возвращаемого значения. Класс Method содержит также коллекцию операторов, которая является реализацией метода. Один оператор может быть либо литералом, либо простым или составным оператором. Составные операторы примечательны тем, что они являются объединением нескольких операторов, которые также могут быть составными. По сути, реализация метода – это также составной оператор.

Метод не всегда может содержать реализацию. Так, например, абстрактные или native-методы не могут иметь тела. Однако, поскольку в разработанной CASE-системе все алгоритмы записываются исключительно в терминах языка Java, поэтому необходимость работы с native-методами возникает лишь при использовании классов, которые их содержат.

CASE-система по построенным моделям генерирует программный код на языке программирования Java, однако она может быть адаптирована для использования с любым объектно-ориентированным языком программирования.

Выводы. Исследованы особенности существующих средств автоматизированной разработки программных средств, проанализированы их преимущества и недостатки. Была разработана CASE-система с учетом обнаруженных недостатков существующих решений.

Разработанная система применима для быстрой разработки несложных программных продуктов, а также может быть применена при обучении программированию учащихся школ и студентов, поскольку пользователь программы оперирует только функциональными и структурными диаграммами, абстрагируясь от особенностей конкретного языка программирования.

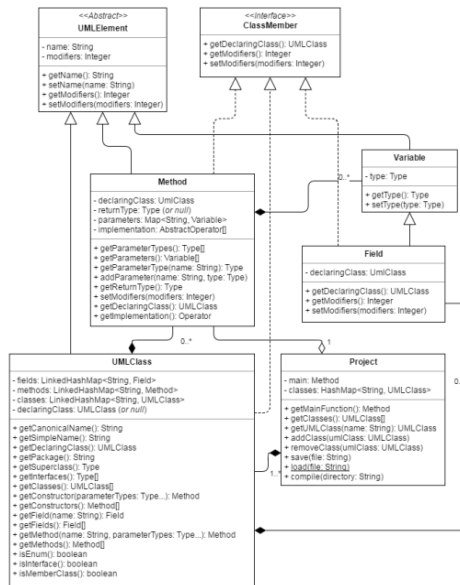


Рис. 4 – Диаграмма классов модели представления каркаса приложения

Список литературы

1. Одинцов И.О. Профессиональное программирование. Системный подход. – 2-е изд. – СПб.: БХВ-Петербург, 2004. – 624 с.
2. Kuhn D. L. Selecting and effectively using a computer aided software engineering tool / D.L. Kuhn // Annual Westinghouse computer symposium; 6-7 Nov 1989; Pittsburgh, PA (USA).
3. Вендров А.М. CASE-технологии Современные методы и средства проектирования информационных систем / А.М. Вендров. URL: <http://www.interface.ru/fset.asp?Url=/LOGWORKS/caset/java1/case.htm>
4. Маклаков С.В. ВРwin и Erwin. CASE-средства для разработки информационных систем – М: Диалог-МИФИ, 2000. – 256 с.
5. Алещенко А.В., Бузовский О.В. Система верификации графических схем алгоритмов и генерации программных кодов / А.В. Алещенко, О.В. Бузовский // Проблемы информатизации и управления. – № 2 (50). – Киев, 2015. – С. 32-35.

УДК 004.9

Современные средства кроссплатформенности для использования и разработки ПО Маруга М.М.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного
E-mail: killerssurprise@mail.ru

Аннотация

Маруга М.М. Современные средства кроссплатформенности для разработки и использования ПО. Выполнен анализ распространенных средств реализации кроссплатформенных приложений. Рассмотрены средства, позволяющие использовать и разрабатывать кроссплатформенное ПО на различных платформах.

Ключевые слова: кроссплатформенные приложения, мультиплатформенные приложения, кроссплатформенность.

Постановка проблемы. В процессе работы с современными мультиплатформенными приложениями, порой приходится сталкиваться с различными проблемами в их работе, такими как: плохая производительность, высокие технические требования для устройств и т.д. Существует множество различных проблем использования и разработки кроссплатформенного ПО, которое необходимо рассмотреть и проработать.

Цель статьи - Рассмотреть и оценить различные современные методы использования и разработки различного кроссплатформенного программного обеспечения. Рассмотреть преимущества и недостатки каждой технологии.

Анализ существующих методов кроссплатформенности. Для начала определим, что же является кроссплатформенностью - это способность работать более чем на одной аппаратной платформе и/или ОС.

Исходя из современных тенденций существует несколько способов достижения кроссплатформенности:

1. Кроссплатформенные языки программирования;
2. Кроссплатформенные среды исполнения;
3. Кроссплатформенный интерфейс.

К кроссплатформенным можно отнести практически любой существующий высокоуровневый язык программирования, программы, которые написаны на популярных языках программирования, например С, С++ и т.д., в случае, если исходный код не привязан к аппаратной платформе и не имеет зависимости от системы, может использоваться в различных средах компиляции.

Основные сложности использования высокоуровневых языков программирования заключаются в том, что они должны быть абсолютно аппаратно независимыми для исполнения, что весьма ограничивает их при реализации какого-либо приложения. Например, для простой консольной программы их будет более чем достаточно, гарантирована высокая скорость работы, так как компилятор сгенерирует машинный код для данной платформы, который будет выполняться в разы быстрее, чем любой интерпретируемый язык программирования. Но если же будет необходимо создать какое-либо графическое приложение, то сразу же возникнут большие проблемы, мы не сможем использовать встроенные средства отображения и визуализации системы (стандарты, типы, шрифты и т.д.), необходимо будет их подгружать автоматически, что усложняет и удорожает стоимость разработки данного программного обеспечения. Данный тип кроссплатформенной разработки больше подойдет для создания различных библиотек, с последующим их использованием. В повседневной жизни мы можем встретить немало примеров такого использования. Например, популярная библиотека OpenCV для работы с компьютерным зрением, ее «движок» написан на языке С/С++ и может использоваться в различных языках программирования, таких как Java, Python, Ruby, Matlab, Lua и т.д. Эта библиотека является прекрасным примером использования высокоуровневых языков программирования для кроссплатформенной разработки

Так же к кроссплатформенным можно отнести различные интерпретируемые языки программирования. Интерпретируемым можно назвать язык, который исполняется без преобразования его в машинный код, а использует для этого интерпретатор. Например, языки PHP, Perl, Python, Ruby –

кроссплатформенные интерпретируемые. Для каждого из них существует интерпретатор для многих платформ. Их основной недостаток заключается все в том же интерпретаторе. Для работы программы необходимо провести промежуточные работы по установке и настройке интерпретаторов. Так же к недостаткам можно отнести относительно низкую скорость работы, по сравнению с нативным кодом и промежуточным байт-кодом. Но, несмотря на свои недостатки, большинство из них нашло свою область применения и пользуются большой популярностью. Не зря многие из признанных университетов мира, в числе которых и Massachusetts Institute of Technology, используют в своей обучающей программе язык Python. Основные характеристики компилируемых от интерпретируемых языков программирования можно рассмотреть на рисунке 1, а их различия исполнения кода на рисунке 2.

КОМПИЛИРУЕМЫЙ		ИНТЕРПРЕТИРУЕМЫЙ	
ГОТОВ К ЗАПУСКУ	НЕ КРОСС-ПЛАТФОРМЕННЫЙ	КРОСС-ПЛАТФОРМЕННЫЙ	ПОСТОЯННО ИНТЕРПРЕТИРОВАТЬ
РАБОТАЕТ БЫСТРЕЕ	НЕГИБКИЙ	ПРОЩЕ ТЕСТИРОВАТЬ	РАБОТАЕТ МЕДЛЕННЕЕ
СКРЫТ ИСХОДНЫЙ КОД	ДОПОЛНИТЕЛЬНЫЕ ШАГИ	ПРОЩЕ ОТЛАЖИВАТЬ	ДОСТУПЕН ИСХОДНЫЙ КОД

Рис.1. Основные характеристики компилируемых и интерпретируемых языков программирования.

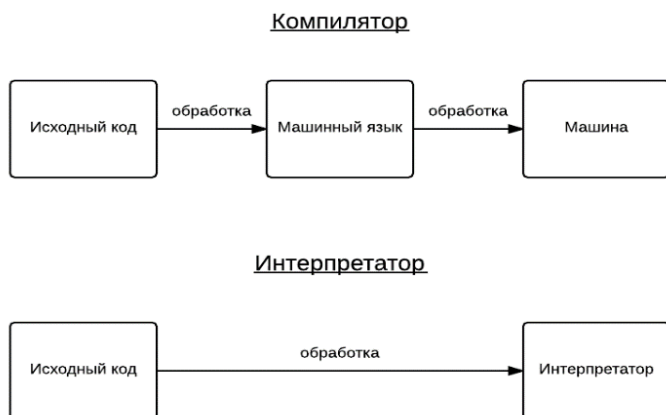


Рис. 2. Различия исполнения кода компилятора и интерпретатора.

Иная технология кроссплатформенного программирования – использование сред исполнения. Например, Java Virtual Machine, разработанный компанией Oracle и .NET, разработанный компанией Microsoft. Их главное отличие в том, что в них подается байт-код, который является промежуточным представлением программы. В отличие от исходного кода, эта программа является особым представлением, где все реализовано для лучшей и быстрой обработки данного кода машиной. В ней закодированы все области видимости, типы, и т.д. С технической точки зрения байт-код является кодом низкого уровня. Его плюс в том, что он исполняется быстрее, чем в различных интерпретируемых языках программирования. Данная технология очень популярна, взять за пример язык Java (Sun Microsystems, в дальнейшем выкуплен Oracle). Изначально идея создания данной концепции языка была продумана для различных бытовых электронных устройств (моб. телефоны, домашняя техника и т.д.), но в последующем произошла переоценка качеств и свойств данной технологии. В последующем Java стал использоваться для написания клиентских и серверных приложений. Его основным достоинством является то, что он полностью не зависит от операционной системы и оборудования, а так же является довольно защищенным, в следствии использования собственной гибкой системы безопасности, которая полностью контролируется виртуальной машиной. Любые операции, не имеющие нужных полномочий, прерываются. Основной принцип исполнения программы Java можно рассмотреть на рисунке 3.

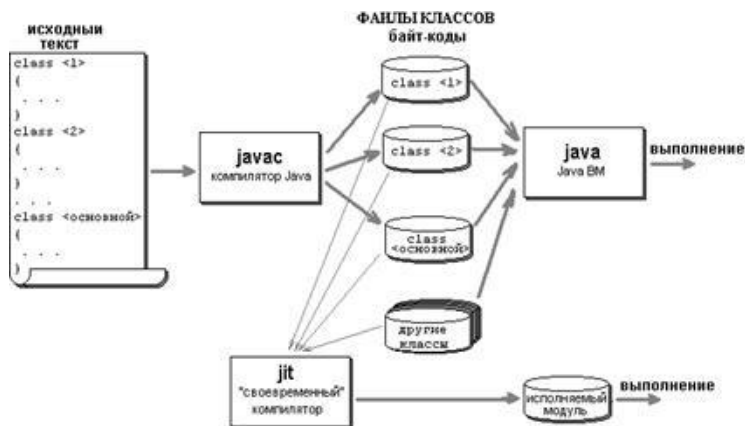


Рис. 3. Внутреннее представление Java-программы.

Но все же, насколько бы технология не была хорошей, в ней имеются и свои недостатки. Например, производительность. Для использования языка Java, необходимо установить дополнительный набор утилит для исполнения, таких как Java Virtual Machine, Java Development Kit. Каждая из этих утилит имеет, пусть

небольшой, но определенный размер и определенные трудности с установкой. Думаю, что почти каждый порой сталкивался с проблемами несовместимости установленных Java версий на компьютере. Так же нередко можно было слышать о «слетевшей», т.е. имеющие некие дефекты работы Java на устройстве, после чего ее необходимо было переустанавливать.

В добавок к вышесказанному, Java код пусть и является байт-кодом, но все же он работает медленнее Native-кода, так как требует определенной интерпретации.

Справедливости ради следует отметить, что существует несколько способов, которые позволяют ускорить работу системы, а именно:

Применение технологии трансляции байт кода в машинный код во время выполнения (JIT);

Использование native-код в стандартных библиотеках;

Иные аппаратные средства ускоренной обработки байт-кода (например, технология Jazelle под некоторые процессоры фирмы ARM).

Данные методы позволяют ускорить работу Java. Однако, согласно статистики shootout.alioth.debian.org и исследований Google в среднем показатели скорости исполнения кода в среднем в 1.5-2 раза ниже, чем в C/C++, а использование оперативной памяти Java-машиной было в 10-30 раз выше, чем при C/C++.

В случаях, когда необходима разработка кроссплатформенного приложения сразу для нескольких устройств с высокой степенью оптимизации и быстродействия, используют различные готовые решения, позволяющие из кода одного проекта продуцировать продукт сразу под несколько платформ. В большинстве случаев они представляют собой готовые среды разработки (MS Visual Studio, Unreal Engine, Unity3D и т.д.), внутри которых создаются различные прослойки, интерпретирующие разработанный проект под определенную платформу, учитывая ее особенности. Так, например, Unreal Engine позволяет создавать проекты под ОС Android, IOS, Windows, Linux, MacOS, т.е. на все наиболее популярные платформы. Данный метод разработки является одним из наиболее распространенных, так как по итогу создается нативный код, дающий преимущество для каждой платформы, хотя проект состоит из одного исходного проекта.

Несмотря на все достоинства данной технологии, у нее существуют и большие недостатки, такие как: закрытость ПО, стоимость готовой IDE, некорректный перенос кода под определенную платформу.

Так же для разработки кроссплатформенного ПО приходится часто пользоваться технологией эмуляции. Эмуляция представляет собой комплекс различных средств, для копирования функций другой вычислительной системы таким образом, чтобы повторить ее свойства и поведение. В современных реалиях повсеместно используются технологии эмулирования VirtualBox и VMware. Данные технологии чаще всего используются для запуска различных устаревших

систем (наверняка, многие пытались эмулировать какие-либо устаревшие игровые консоли, чтобы вновь окунуться в детство). Но так же им есть место и в современном мире. Например, для отладки различного кроссплатформенного ПО необходимо протестировать поведение на различных ОС. Обычно, для этого потребовалось бы 3 устройства с предустановленными ОС. Это намного удорожает процесс написания ПО. В большинстве случаев используют либо машину с несколькими ОС и поочередной их загрузкой, что в свою очередь сказывается на временных затратах. Либо же, как в нашем случае, используют технологии эмулярования, которые позволяют исследовать поведение ПО на других платформах непосредственно на разрабатываемой машине. Данный подход хорош, однако следует отметить, что в этом случае необходимы весьма внушительные параметры для оборудования. Из минусов так же следует отметить необходимость установки виртуальной машины и ее настройки.

Иногда, когда выбор разработчиков падает на использование слоев совместимости. Например, wine и cygwin. Первый используется для совместимости WinApi для Unix, MacOS и прочих систем. Второй же представляет собой слой совместимости Linux для ОС Windows. Хотя в текущем этапе развития эти программы имеют достойную производительность, обусловленную тем, что они поддерживают переносимость на бинарном уровне, они имеют все те же недостатки что и для предыдущих способов: установка, настройка и некорректное использование программ при некоторых ситуациях.

Выводы. Рассмотрены и оценены различные современные методы разработки и использования кроссплатформенного программного обеспечения. Изучены преимущества и недостатки каждой технологии. Проведена примерная оценка производительности для сравнения отдельных технологий. Рассмотрели такие технологии кроссплатформенной реализации ПО как высокоуровневые языки программирования, интерпретируемые языки программирования, различные среды исполнения, современные средства.

Список литературы

1. Хорошилов Алексей Владимирович Россия, Москва, Институт системного программирования РАН, Силаков Денис Владимирович Россия, Москва, Институт системного программирования РАН, **МОБИЛЬНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СОВРЕМЕННЫХ УСЛОВИЯХ.**

2. Дж. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд.: Питер, 2013.-896с.

3. David Thomas. The Pragmatic Programmer: From Journeyman to Master 1st Edition. AddisonWesley Professional; 1 edition (October 30, 1999).

4. Герберт Шилдт. Java 8. Полное руководство, 9-е издание// Вильямс, ISBN: 978-5-8459-1918- 2, 2015. -1377с.

5. Маруга Михаил. СОВРЕМЕННЫЕ ПРОБЛЕМЫ МОБИЛЬНОСТИ ПО И ВОЗМОЖНЫЕ МЕТОДЫ ИХ РЕШЕНИЯ. Сборник статей студенческой научно-технической конференции «В мире компьютерных технологий» г. Севастополь 4-8 апреля 2016г.

6. Java. About. [Электронный ресурс]. Официальный сайт. URL: <https://www.java.com/ru/about/>

7. Внутреннее представление Java. [Электронный ресурс] URL: <http://ermak.cs.nstu.ru/trans/Trans471.htm>.

8. Компилированные и интерпретируемые языки программирования. [Электронный ресурс] URL: <https://sozdaj-sam.com/javascript/kompilirovannye-i-interpretiruemye-jazyki.html>

УДК 519.173

**Разработка и исследование метода контроля карты рабочей среды агента
Черкас Д.В., Кравец Т.Н.**

Донецкий национальный технический университет
Кафедра искусственного интеллекта и системного анализа
cherkassdmitry@gmail.com

Аннотация

Черкас Д.В., ст. пр. Кравец Т.Н. Разработка и исследование метода контроля карты рабочей среды агента. Рассмотрены и проанализированы известные методы и алгоритмы проверки соответствия (изоморфизма) рабочей среды агента и карты этой среды. Определен метод, а также разработан алгоритм проверки изоморфизма графов блуждающим агентом.

Общая постановка проблемы. При исследовании взаимодействий автоматов и графов возникает актуальная проблема анализа (распознавания) свойств графа при разной априорной информации, такой как вид графа, и разных способах взаимодействия автомата и графа. При этом данная проблема имеет в себе, как теоретические, так и практические аспекты.

Проблема определения, являются ли два графа изоморфными, неожиданно оказывается трудной. Для произвольных графов все известные алгоритмы, гарантирующие правильный ответ, экспоненциальные. Для того, чтобы доказать, что графы изоморфны, их надо необходимо проанализировать, то есть каким-то образом обойти все вершины графов и дать результат о их изоморфности. В этом и заключается главная проблема, для анализа графов необходимо использовать корректный алгоритм, который со 100% вероятностью

и за «адекватное» время будет выдавать результат о изоморфности графа. Наличие этой проблемы делает данную тему актуальной.

Во время разработки методики проверки соответствия, было решено использовать подход навигации мобильных роботов. Под мобильными роботами в данной работе следует иметь ввиду неких агентов, которые будут обходить и помечать вершины графа. Этот подход основывается на том, что топологическая среда рассматривается как конечный неориентированный граф с помеченными концами ребер — вершинами графа.

Исследования. Разрабатываемый алгоритм основан на алгоритме, который описан в работе Грунского И.С. и Степкина А.В., он базируется на обходе графа в глубину, пока это возможно, затем агент возвращается назад, ищет другой путь с еще не посещенными вершинами и не пройденными ребрами. В случае обнаружения смежной вершины, окрашенной в «чужой» цвет, агент метит все перешейки из текущей вершины в чужую область и сообщает второму агенту через АЭ о необходимости распознавания помеченных перешейков. В случае отсутствия других возможных вариантов перемещения, кроме как пометить новый перешеек, первый агент останавливается до того момента, пока второй агент не распознает все помеченные перешейки.

В отличие от предыдущего алгоритма, мой основан не только на стратегии поиска в глубину, но и в ширину одновременно и заключается в следующем: агент помечает элементы среды и карты, как зеленые (когда «агент» прошел по ребру), красные (когда «агент» впервые посетил вершину) и черные (когда у вершины нет не пройденных путей).

Входными данными для алгоритма сравнения карты и неизвестной среды мобильным агентом являются два конечных, неориентированных графа без петель и кратных ребер, один из которых соответствует полностью известной карте, а другой — неисследованной среде, которая исследуется с помощью агента и карты.

Разработанный алгоритм сравнения карты и среды мобильным агентом состоит из следующих этапов:

4. Начало. Агент начинает работу в стартовой позиции (она может быть задана пользователем или же выбирается случайно).

5. Создание путей на карте. Агент считает валентность стартовой вершины (количество ребер у вершины). Находит на карте все вершины с такой-же валентностью и создаёт пути, с началом в каждой из найденных вершин.

6. На этот момент: *текущая вершина среды* (ТВС) — стартовая вершина, каждый путь состоит из одной вершины, которая является *текущей вершиной пути* (ТВП).

7. 3. Поиск следующей вершины. Агент ищет в *среде* не пройденное ребро у ТВС (ребро, не помеченное зелёным цветом).

8. Если ребро найдено: перекрашиваем его в зелёный цвет, помечаем вершину как красную, а если все рёбра зелёные, то как чёрную. Переходим по найденному ребру к вершине, которая становится ТВС. Переход в пункт 4.

9. Если ребро не найдено, то проделываем «шаги назад» в *среде* (и в каждом *пути* на карте), пока не найдём красную вершину. Если нашли — помечаем найденную вершину, как ТВС идём к пункту 3; иначе к пункту 6.

10. 4. Поиск продолжения путей. Перебираем все пути на карте, построенные на момент начала пункта 4, и ищем их продолжение:

11. - берём *путь*, среди соседних с ТВП (кроме вершины, предшествующей ТВП) ищем - все вершины с валентностью ТВС;

12. - для каждой найденной вершины, создаём путь, как копию данного, с добавлением найденной вершины, помечаемой как ТВП;

13. - удаляем путь;

14. 5. Шаг. Если, нет ни одного пути, то *среда* и *карта* не изоморфны, иначе идём к пункту 3.

15. 6. Результат. Если количество рёбер в *пути* и в *карте* одинаково, то карта и среда изоморфны. Иначе — не изоморфны.

16. В результате работы был разработан метод и алгоритм проверки соответствия (изоморфизма) рабочей среды агента и карты этой среды. На тестовых задачах алгоритм показал себя хорошо, сравнение с аналогами не проводилось.

Выводы. Разработанный алгоритм может быть использован для решения задач автоматопоподобных систем, в теоретических и прикладных исследованиях, связанных с проблемой навигации мобильных роботов, в моделях представленных знаний, в интеллектуальных системах принятия решений, в системах поиска в сетях ЭВМ и в мультиагентных системах.

Список литературы

1. Проблема изоморфизма графов / Интернет-ресурс. - Режим доступа: <http://xn--90abr5b.xn--p1ai/exams/%D1%81%D0%B0%D0%BE%D0%B4/41.htm>

2. Килибарда Г. О поведении автоматов в лабиринтах [Текст] - 1992, с. 3-28.

3. И.С. Грунский, А. В. Степкин. Распознавание неориентированных графов коллективом агентов [Текст] – 2008, с.84-86.

4. Распознавание конечного графа блуждающим по нему агентом / Интернет ресурс. - Режим доступа: <http://masters.donntu.org/2013/fknt/shumilina/library/article5.pdf>

УДК 004.942

**Анализ существующих подходов к разработке программ для анализа
прогноза погоды**

Сидоренко А.С., Коломойцева И.А.

Донецкий Национальный технический университет
Кафедра программной инженерии

Аннотация

Сидоренко А.С., Коломойцева И.А. Анализ существующих подходов к разработке программ для анализа прогноза погоды. В статье рассмотрены использованные средства для разработки приложения-анализатора прогноза погоды согласно данным из нескольких источников, выделены их преимущества и недостатки. Исследованы аналогичные программные продукты сторонних компаний на предмет удобства в использовании, указаны их проблемы совместимости с другими программами.

Общая постановка проблемы. Прогноз погоды всегда был, есть и будет одним из важнейших данных, которым интересуется любой современный человек. Давным-давно его пытались определять по теням, поведению животных другим признакам. С появлением метеостанций, которые с большей точностью определяют грядущую погоду, узнать прогноз стало возможным из газет и ТВ. Сегодня же, с появлением Интернета стало доступно множество сайтов, которые со всеми подробностями покажут, какой будет погода в любое время и в любом месте земного шара.

Назревает вполне логичный вопрос: а кому же верить, если источников информации много, но данные разнятся в ту или иную сторону? Абсолютное большинство пользователей сервисов всемирной паутины используют для поиска прогноза только те порталы, которые дают более-менее правдивые сведения о погоде именно в их регионе. Остальные же вряд ли задумываются о существовании других погодных сайтов или же просто не имеют достаточных навыков в пользовании ПК.

На базе любой ОС существует несметное множество клиентских программ, которые отображают текущую погоду в указанном месте, а также прогноз на ближайшие 3-4 дня. Такие программы легки в использовании, имеют интуитивно понятный интерфейс, в котором разберется даже ребенок. Одним из примеров является погодный гаджет для рабочего стола в Windows Seven. Он не занимает много места даже в развёрнутом виде и, в то же время, достаточно информативен. Но у него, как и всех остальных программ, есть существенный недостаток: программа заточена для работы только на определенном сайте и невозможно его использовать для поиска прогноза на других.

В рамках данной работы будет разработана программа, которая получает информацию о прогнозе с нескольких Интернет-источников, а также реальные сведения о погоде за минувшее время для последующего анализа и вывода сведений о достоверности или ложности прогнозов в виде графиков. В качестве языка программирования был выбран язык Java и интегрированная среда разработки NetBeans IDE.

Исследования. У большинства погодных агентов функционал ограничен отправкой запроса на сервер, получением данных, выводом их на экран и постоянным обновлением данных через определенный интервал, который или фиксирован, или может задаваться пользователем. В процессе исследования не было обнаружено ни одной такой программы, которая собирает данные с нескольких источников. Такой подход имеет много плюсов, главный из которых - возможность зрительного анализа полученных данных. Пользователь, увидев погоду с нескольких источников, может сравнить их и для себя решить, какому из них можно верить, тем более если данные как минимум с двух порталов совпадают. На втором месте, конечно же, время. Для поиска информации всегда необходимо время, даже если все навигация по всем сайтам доведена у пользователя до автоматизма. Но когда речь идет о количестве потребленного трафика, который может быть неограниченным, много тяжелых страниц с рекламой и прочими дополнениями вряд ли захочется открывать.

Так как основной из задач приложения является получение данных из веб-страниц, то для этой цели было принято решение использовать библиотеку Jsoup версии 1.9.1.

Jsoup – это библиотека Java для работы с реальной HTML-страницей. Библиотека обеспечивает очень удобный API для извлечения и обработки данных, используя лучшие методы из DOM, CSS, JQuery и других библиотек, которые чаще всего используются при разработке веб-приложений[1].

Основные функции, которые выполняет библиотека Jsoup:

- парсинг HTML-страницы, которую можно получить, указав URL, путь к файлу или строке;
- поиск и извлечение данные, используя DOM-обход или CSS-селекторы;
- манипуляция HTML-элементами, атрибутами и текстом;
- чистка пользовательского контента, для предотвращения XSS - атаки на веб-системы, заключающейся во внедрении в выдаваемую веб-системой страницу вредоносного кода (который будет выполнен на компьютере пользователя при открытии им этой страницы) и взаимодействии этого кода с веб-сервером злоумышленника[3].
- аккуратный выход из HTML-страницы.

Помимо вывода данных в текстовом виде, в программе будет предусмотрена функция вывода графиков, которые показывают погрешность в прогнозе по двум критериям: температура и осадки. За вывод информации в виде

графиков отвечает библиотека XChart. Она является небольшой по размерам и удобной для построения графиков, имея некоторый ряд данных. Библиотека позволяет отобразить график в программе или же сохранить его отдельным файлом[2].

Особенности библиотеки:

- не требуются никакие внешние зависимости, библиотека самостоятельна;
- размер в формате *.jar не превышает 200КБ;
- линейные графики;
- точечные диаграммы;
- диаграммы с областями;
- гистограммы;
- гистограмма графики;
- логарифмические оси;
- обширные настройки;
- много вариантов оформления (тем);
- определяемые пользователем диапазон осей;
- пользовательское размещение легенды;
- экспорт в PNG, JPG, BMP и GIF.

Ниже приведены основные программы для поиска погоды и краткое описание каждой из них.

YoWindow – это программа, целью которой является визуальное отображение погодных условий в вашем регионе. Подобных утилит на данный момент много, но отличие YoWindow от остальных состоит в том, что кроме текстовых показателей погодных условий, вы можете посмотреть погоду прямо на картинке, так, как будто пользователь сам смотрит в свое окно. Так, например, если в данный момент по прогнозу облачно, то на Flash изображении будут передвигаться облака, а если дует сильный ветер, то элементы изображения будут сильно колыхаться. Кроме того, солнце на картинке перемещается соответственно времени дня, а если открыть программу ночью, то и «за окном» ночной пейзаж.

Показатели, которые поддерживаются программой:

- температура;
- облачность;
- скорость и направление ветра;
- давление;
- влажность;
- время суток и день календаря.

После установки программы нужно выбрать свою страну и город. В программе возможно найти любую страну и любой город, даже самый маленький.

Среди ключевых особенностей и функций программы можно выделить следующие:

- качественные изображения и анимации покажут вам погодные условия окном;
- картинку легче запомнить, чем отчет о погоде;
- поддерживаются различные погодные явления;
- с изменением погодных условий меняется и изображение[4].

К недостаткам программы можно отнести то, что она является условно бесплатной. Это означает, что после загрузки и установки пользователь получает урезанную часть программы на 3 дня, что не дает в полной мере оценить данное приложение.

Weather Watcher – программа, которая автоматически получает текущие условия, прогноз на час, на день, детальный прогноз, предупреждения о сложных погодных условиях, и погодные карты для более 77000 городов мира. При чем, предупреждения о сложных погодных условиях доступно только для городов США.

Удобству при работе с программой способствует то, текущие условия быстро просматриваются наведением курсора мышки на значок системы Weather Watcher в трее.

Weather Watcher позволяет автоматически получать данные о погоде в заданные интервалы, отображать текущие условия в значке в трее, выбирать информацию, отображаемую в окне Weather Watcher, отображать погодные условия различными вариантами.

Программа имеет ряд особенностей и преимуществ, которые, по заявлению разработчиков, ставят ее на порядок выше среди других погодных клиентов:

- прогнозы для 77,000+ городов по всему миру;
- текущие погодные условия;
- 46-часовой прогноз;
- прогноз на 10 дней/ночей;
- сигналы о тяжелых погодных условиях из National Weather Service (Национальной Службы Погоды);
- более 450 карт;
- более 18 единиц выбора;
- возможность локально экспортировать и регистрировать данные о погоде;
- скины интерфейса/прогнозов;
- настраиваемая в системном трее
- двойной трей иконок для температур и состояния изображений;
- поддерживает прокси-сервер;
- использует меньше памяти, чем другие программы при получении информации о погоде;
- 7 бонусов прогноз/интерфейс-скины.

В работе программы недостатков замечено не было, но большинством антивирусов программа распознается как Adware – программа, которая предназначена для показа назойливой рекламы на компьютере, перенаправления поисковых запросов и сбора маркетинговой информации о пользователе, то есть какие интернет-магазины посещает, чем в них интересуется и что покупает. Такая реакция антивирусов не зря: программа предлагает скачать и установить тулбар, в котором нет необходимости - программа работает и без него, а также пытается изменить поисковый сервер по умолчанию для веб-браузеров, установленных в операционной системе[5].

PC Weather Machine - интегрированная виртуальная метеостанция, которая предлагает больше метеорологической информации, чем большинство погодных модулей. Она позволяет держать пользователя в курсе информации о погоде, отображая метеорологические условия, карты погоды и городов для мест на выбор. Эта информация о погоде доступна по всему миру бесплатно.

Основной модуль отображает пять типов датчиков - барометр, термометр, гигрометр, анемометр и флюгер. Они показывают текущие погодные условия в указанном пользователем городе. Эти датчики отображают текущее значение, минимальные и максимальные показания зарегистрированных и перепады за последние 24 часа. Второй указатель анемометра и термометра показывают скорость порывов ветра и точки росы, соответственно.

Тенденции в течение последних 24 часов нанесены на пять графиков: атмосферное давления, температура, влажность, скорость и направление ветра.

Модуль просмотра карт отображает восемь полноцветных изображений, которые показывают метеорологические условия, прогнозы, снимки со спутников, изображения с радаров и карты погоды для местности. У пользователя есть возможность задать ссылки, чтобы наблюдать изменения в режиме реального времени. Эти изображения могут быть скопированы, распечатаны и сохранены[6].

Выводы. В ходе работы будет разработана программа для анализа прогноза погоды на предмет достоверности согласно данным, полученным из нескольких источников. Основными используемыми библиотеками будут Jsoup и XChart. Это мощные средства для получения данных из веб-страниц и вывода их в виде графиков соответственно. Также рассмотрены аналогичные программы для просмотра погоды, наиболее мощной можно считать PC Weather Machine. Она представляет самый широкий функционал, а все данные можно просматривать в любом виде в режиме реального времени. Но ни одна из исследованных программ не производит анализа достоверности прогноза погоды, поэтому было принято решение разработать программу, решающую эту задачу.

Список литературы

1. Jsoup [электронный ресурс]. – Режим доступа: <https://jsoup.org/apidocs/>
2. Knowm.org | XChart [электронный ресурс]. – Режим доступа: <http://knowm.org/open-source/xchart/>
3. Википедия. Межсайтовый скриптинг [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Межсайтовый_скриптинг
4. YoWindow [электронный ресурс]. – Режим доступа: <http://soft.mydiv.net/win/download-YoWindow.html>
5. Weather Watcher [электронный ресурс]. – Режим доступа: <http://soft.mydiv.net/win/download-Weather-Watcher.html>
6. PC Weather Machine [электронный ресурс]. – Режим доступа: <http://soft.mydiv.net/win/download-PC-Weather-Machine.html>

УДК 004.7

Модели информационной безопасности

доц., к.ф.-м.н. Швыров В.В.

Луганский государственный университет имени Тараса Шевченко

Аннотация

Швыров В.В. Модели информационной безопасности. При построении моделей безопасности информационных систем достаточно трудно установить взаимосвязи между субъектами и объектами системы и дать строгое описание характеристик защищенности системы. Удобным аппаратом для такого рода задач является аппарат нечетких множеств и нечетких бинарных отношений. В работе изучаются методы оценки защищенности узлов компьютерной сети с помощью построения нечетких бинарных отношений.

Ключевые слова: модели безопасности, нечеткие множества, нечеткие бинарные отношения, компьютерные сети.

Введение. Практическая задача обеспечения информационной безопасности состоит в разработке модели представления системы информационной безопасности, которая на основе научно-методического аппарата, позволяла бы решать задачи создания, использования и оценки эффективности системы защиты информации для проектируемых и существующих уникальных информационных систем.

Выделяют несколько подходов для классификации моделей безопасности [3]:

- общая модель процесса защиты информации;
- обобщенная модель системы защиты информации;
- модель общей оценки угроз информации.

Сложность процесса принятия решений, отсутствие математического аппарата приводят к тому, что при оценке и выборе приходится использовать и обрабатывать качественную экспертную информацию. Перспективным направлением разработки методов принятия решений при экспертной исходной информации является лингвистический подход на базе теории нечетких множеств и лингвистической переменной [6].

Основные определения. Определение. Нечетким числом называется нечеткое множество A , определенное на множестве действительных чисел R , если его функция принадлежности нормальна и выпуклая, т.е.

$$\sup_{x \in R} \mu^A(x) = 1,$$

$$x \leq y \leq z \Rightarrow \mu^A(y) \geq \min(\mu^A(x), \mu^A(z)).$$

Для нечетких чисел удобно представление в LR-форме вида:

$$\mu^A(x) = \begin{cases} L\left(\frac{m-x}{\alpha}\right), \alpha > 0, \forall x \leq m, \\ R\left(\frac{m-x}{\beta}\right), \beta > 0, \forall x \leq m, \end{cases}$$

где L, R - функции обладающие свойствами

а) $L(-x) = L(x), R(-x) = R(x),$

б) $L(0) = 1, R(0) = 1,$

в) L, R - монотонно убывают на промежутке $[0, +\infty]$

Здесь - m - среднее значение нечеткого числа A , α, β - отклонения слева и справа, соответственно. Таким образом, LR-форма нечеткого числа может быть задана тройкой вида $A = \{m_A, \alpha_A, \beta_A\}$, тогда арифметические операции примут вид

$$A + B = \{m_A, \alpha_A, \beta_A\} + \{m_B, \alpha_B, \beta_B\} = \{m_A + m_B, \alpha_A + \alpha_B, \beta_A + \beta_B\},$$

$$A - B = \{m_A, \alpha_A, \beta_A\} - \{m_B, \alpha_B, \beta_B\} = \{m_A - m_B, \alpha_A - \alpha_B, \beta_A - \beta_B\},$$

$$A \cdot B = \{m_A, \alpha_A, \beta_A\} \cdot \{m_B, \alpha_B, \beta_B\} = \{m_A \cdot m_B, m_A \alpha_A + m_B \alpha_A, m_A \beta_A + m_B \beta_B\}.$$

На практике удобно представлять нечеткие числа треугольной формой и трапециевидной формой функций принадлежности.

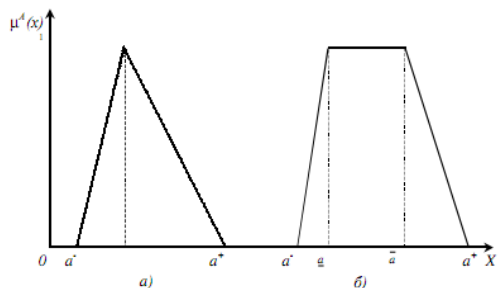


Рис. 1. Форма функции принадлежности

Определение. Нечеткое множество $A \subset X$ представляет собой набор пар $\{(x, \mu^A(x))\}$, где $x \in X$ и $\mu^A : X \rightarrow [0, 1]$ — функция принадлежности, которая представляет собой некоторую субъективную меру соответствия элемента нечеткому множеству $\mu^A(x)$ и может принимать значения от нуля, который обозначает абсолютную не принадлежность, до единицы, которая, наоборот, говорит об абсолютной принадлежности элемента x нечеткому множеству A .

Если нечеткое множество A определено на конечном универсальном множестве $X = \{x_1, x_2, \dots, x_n\}$, то его удобно обозначать следующим образом:

$$A = \mu^A(x_1) / x_1 + \mu^A(x_2) / x_2 + \dots + \mu^A(x_n) / x_n = \sum_{i=1}^n \mu^A(x_i) / x_i$$

где $\mu^A(x) / x$ — пара "функция принадлежности/элемент", называемая синглтоном, а "+" — обозначает совокупность пар.

Подобно бинарным отношениям на обычном множестве введем понятие нечеткого бинарного отношения.

Определение. Пусть $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_m\}$. Нечетким бинарным отношением R называется нечеткое множество определенное на прямом произведении $X \times Y$, которому соответствует функция принадлежности $\mu^R : X \times Y \rightarrow [0, 1]$, где $\mu^R(x, y)$ отражает степень зависимости между $x \in X$, $y \in Y$.

Описание модели системы. Таким образом, математическую модель безопасности информационной системы (например, в корпоративной сети) удобно описать с помощью терминологии нечетких множеств.

Пусть $K = \{A_1, A_2, \dots, A_n\}$ - конечное множество объектов корпоративной сети. Определим множество $S = \{B_1, B_2, \dots, B_m\}$ как множество субъектов доступа к сети.

Для каждого объекта $\{A_1, A_2, \dots, A_n\}$ методом экспертных оценок можно определить доступность данного объекта для каждого из субъектов $\{B_1, B_2, \dots, B_m\}$. Причем, полагаем, что 0 означает полную недоступность (в идеальном случае), 1 - полный доступ к объекту. Таким образом, мы задали тотальное нечеткое бинарное отношение $\sigma \subset K \times S = \{(a_i, b_j) \mid a_i \in K, b_j \in S\}$.

Матрица смежности нечеткого бинарного отношения σ характеризует доступность объектов корпоративной сети.

$$M_\sigma = \begin{pmatrix} \mu^\sigma(a_1, b_1) & \mu^\sigma(a_1, b_2) & \dots & \mu^\sigma(a_1, b_m) \\ \mu^\sigma(a_2, b_1) & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \mu^\sigma(a_n, b_1) & \dots & \dots & \mu^\sigma(a_n, b_m) \end{pmatrix}$$

Дополнением нечеткого бинарного отношения будет отношение $\bar{\sigma}$, которое означает степень угрозы для объектов сети.

$$M_{\bar{\sigma}} = \begin{pmatrix} 1 - \mu^\sigma(a_1, b_1) & 1 - \mu^\sigma(a_1, b_2) & \dots & 1 - \mu^\sigma(a_1, b_m) \\ 1 - \mu^\sigma(a_2, b_1) & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ 1 - \mu^\sigma(a_n, b_1) & \dots & \dots & 1 - \mu^\sigma(a_n, b_m) \end{pmatrix}$$

Определим также нечеткое бинарное отношение $\omega \subset K \times K$, и отношения $\gamma \subset S \times S$ на множествах. Отношение ω - показывает зависимости между объектами сети исходя из количественных оценок трафика сети

$$M_{\omega} = \begin{pmatrix} \mu^{\omega}(a_1, a_1) & \mu^{\omega}(a_1, a_2) & \dots & \mu^{\omega}(a_1, a_n) \\ \mu^{\omega}(a_2, a_1) & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \mu^{\omega}(a_n, a_1) & \dots & \dots & \mu^{\omega}(a_n, a_n) \end{pmatrix},$$

Значения функции принадлежности для пар элементов (a, a) показывает защищенность конкретного объекта исходя из внутренних настроек операционной системы или конфигурации оборудования объекта корпоративной сети.

При оценке защищенности сети следует учесть также факторы социальной инженерии, а именно степень знакомства сотрудников (субъектов корпоративной сети). В данном примере отношение γ показывает данный вид связей. Обозначим матрицу отношения γ через M_{γ} .

Таким образом, после составления деревьев опасностей для каждого возможного пути реализации атаки мы определяем источник угрозы. В нашем случае это будут субъекты корпоративной сети $S = \{B_1, B_2, \dots, B_m\}$. Также, необходимо выявить конкретные объекты угроз исходя из построенных деревьев. Таким образом, мы определяем нечеткое бинарное отношение определяющее степень угрозы от субъекта корпоративной сети на определенное множество ее объектов. Матрица данного бинарного отношения будет иметь вид

$$M_{\rho} = \begin{pmatrix} \mu^{\rho}(a_1, b_1) & \mu^{\rho}(a_1, b_2) & \dots & \mu^{\rho}(a_1, b_m) \\ \mu^{\rho}(a_2, b_1) & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \mu^{\rho}(a_n, b_1) & \dots & \dots & \mu^{\rho}(a_n, b_m) \end{pmatrix},$$

Нечеткое бинарное отношение $\sigma \cup \rho$ будет определять степень угроз для объектов с учетом смоделированных деревьев опасностей

$$M_{\rho \cup \sigma} = \begin{pmatrix} \mu^{\rho \cup \sigma}(a_1, b_1) & \mu^{\rho \cup \sigma}(a_1, b_2) & \dots & \mu^{\rho \cup \sigma}(a_1, b_m) \\ \mu^{\rho \cup \sigma}(a_2, b_1) & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \mu^{\rho \cup \sigma}(a_n, b_1) & \dots & \dots & \mu^{\rho \cup \sigma}(a_n, b_m) \end{pmatrix}$$

Используя операции над нечеткими числами необходимо выполнить корреляцию построенных матриц нечетких бинарных отношений. Следующим этапом будет сведение всех полученных данных в единую матрицу, которая будет давать удобную характеристику для всех узлов сети.

Выводы. Использование аппарата нечетких множеств при построении модели безопасности информационной системы позволяет систематизировать разрозненную информацию и связи между субъектами и объектами в системе. Задавая различным образом бинарные отношения между объектами сети возможно выявление скрытых закономерностей и связей, что позволит более точно оценить степень защищенности всей системы.

Список литературы

1. Баричев С.Г. Основы современной криптографии. / С.Г. Баричев, Р.Е. Серов. — М.: Горячая линия - Телеком, 2002. — 152 с.
2. Глухих В.И. Информационная безопасность и защита данных: учебное пособие / В.И. Глухих. — Иркутск: Изд-во Иркутского государственного технического университета, 2011. — 250 с.
3. Домарев В.В. "Безопасность информационных технологий. Методология создания систем защиты" / В.В. Домарев – К.: ООО "ТИД "ДС", 2002 – 688 с.
4. Завгородний В.И. Комплексная защита информации в компьютерных системах. / В.И. Завгородний. — М.: Логос, 2001. — 264 с.
5. Романец Ю.В. Защита информации в компьютерных системах и сетях. / Ю.В. Романец, И.А. Тимофеев, В.Ф. Шаньгин. — М.: Радио и связь, 1999. — 328 с.
6. Кофман А. Введение в теорию нечетких множеств М.: Радио и связь, 1982-432с.
7. Поспелов Д.А. Нечеткие множества в моделях управления и искусственного интеллекта. -М.: Наука, 1986-312с.
8. Шварц М. Сети связи: протоколы, моделирование и анализ. – М.: Наука, 1992. – Ч.1–2.
9. Шеннон Р. Дж. Имитационное моделирование систем – искусство и наука. – М.: Мир, 1978. – 418 с.

УДК 004.273

Использование облачных технологий в игровой индустрии. Визуализация и уменьшение требований к «железу» клиента

Шумский А.А., Бычкова Е.В.

Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
E:mail elena.bichkova@gmail.com

Аннотация

Шумский А.А., Бычкова Е.В. Использование облачных технологий в игровой индустрии. Визуализация и уменьшение требований к «железу» клиента. Проанализированы современные виды игровых продуктов, а также способы их распространения и разработки. Приведены инновационные варианты развития индустрии игр, с использованием облачных технологий обработки данных. Раскрыты основные проблемы разработки облачных игр и сервисов.

***Ключевые слова:** Cloud technologies, облачный движок, VR, технология распределенной обработки данных, контент, игровые локации, OnLive, Grid.*

Постановка проблемы. В настоящее время индустрия развлечений не стоит на месте и занимает внушительную часть программного обеспечения. Существует большое количество различных видов игр, использующих облачные технологии (cloud technologies) для решения тех или иных задач.

Большой проблемой является имеющаяся разница в уровне производительности и архитектуре «железа» игроков. Это накладывает серьезные ограничения на проводимые разработки и не позволяет реализовать определенные идеи авторов. В связи с этим проблема разработки облачных игр в настоящее время стала крайне актуальной.

Анализ литературы. Проведен анализ литературы и рынка игровых приложений и сервисов. В статье описаны новые методы разработки приложений из области развлечений с использованием облачных вычислений, а также рассматриваются подходы к их реализации. Проведены параллели с существующими облачными технологиями и сервисами, которые занимают лидирующие позиции на рынке.

Цель статьи – проанализировать возможность использования облачных вычислений как средств реализации концепции игры на любом устройстве.

Облачные технологии в игровой индустрии. Облачные вычисления (cloud computing) — это технология распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис.

Структуру облака можно представить в виде пирамиды (рис. 1). Основание пирамиды «инфраструктура» – это набор физических устройств (серверы, жесткие диски и т.д.), над ней выстраивается «платформа» - набор услуг и верхушка – программное обеспечение, доступное по запросу пользователей.

Все чаще облачные технологии используются в сфере компьютерных игр. Начиная от однопользовательских игр с «соревновательными» моментами и заканчивая браузерными многопользовательскими играми. Особую популярность они обрели за счет социальной составляющей.



Рис. 1. Структура облака

В большинстве случаев основная часть расчетов ложится на «железо» клиента, что накладывает ограничения на разработку масштабных игр для широкой аудитории. От этого страдает качество, что влечет за собой уменьшение количества продаж и снижение популярности.

Возможность переложить все вычисления на сторону сервера и требовать наличия лишь интернета и минимального по производительности устройства с возможностью ввода-вывода, является краеугольным камнем будущего развития игровой индустрии в целом.

На данный момент существует несколько облачных сервисов, с помощью которых можно играть практически на любом устройстве. OnLive, Grid и некоторые более современные аналоги используют серверные машины, на которых запускаются отдельные лицензионные копии игр. «Железо» пользователя выступает в качестве терминала. С его стороны на сервер идут пакеты, содержащие информацию с устройств ввода, обычно отвечающие за контроль игрового персонажа, а в ответ игрок получает видеоряд со звуком. Благодаря подобной технологии пользователь сервиса может играть в любые игры, доступные в списке, при наличии достаточно скоростного интернет-соединения.

У таких облачных сервисов (OnLive, Grid и подобные) есть минусы, среди которых:

- высокие требования к качеству и скорости интернет-соединения;
- ограниченный список доступных проектов;
- дороговизна обслуживания серверов, что влечет за собой рост стоимости подписки;
- более низкое качество графики, если сравнивать с классическими играми на мощных персональных компьютерах.

Стоит отметить, что рост пользователей пропорционален требованию к вычислительной мощности на сервере. С одной стороны, это хорошо, так как можно рассчитать рост требований к мощности сервера в будущем, имея данные о росте числа пользователей. С другой стороны, желательно снижение требований к серверам, учитывая то, что выполняется код одной игры, содержащий один и тот же контент. Это особенно актуально для многопользовательских игр в реальном времени, когда игроки существуют одновременно на одном игровом поле.

Логичным решением данной проблемы является новая концепция облачных игровых движков, которые смогут использовать общие ресурсы сервера для обработки графики и физики всех игроков на сцене. Идея состоит в том, чтобы не повторять одни и те же расчеты для каждого клиента, а использовать единую систему для этих задач. Данный подход требует низкий уровень доступа к «железу» сервера для ядра движка. Подобная реализация логики позволит создавать и обслуживать огромные и крайне сложные игровые локации с большим количеством контента. Также будет достигнута простота работы с обновлениями. Необходимость достижения корректной работы игры на большом количестве конфигураций «железа» вовсе отпадет.

Основные преимущества описанной технологии. Одной из главных особенностей данной концепции для игроков является «равенство» в играх, которое заключается в предоставлении сервером одинаковой картинки, а также скорости реакции на игровые события, зависящие только от интернет-соединения. Уйдут в прошлое такие проблемы:

- падение производительности из-за большой нагрузки на железо пользователя, что было особенно заметно на компьютерах разной мощности, и давало преимущество игрокам;
- ошибки, вызванные старыми версиями драйверов или библиотек;
- значительные задержки сервера из-за большого наплыва игроков.

Последняя проблема особенно актуальна в многопользовательских играх с большим «открытым миром», где множество игроков могут сойтись на одной игровой локации. Ярким примером работы сервера с большими нагрузками является игра «Eve Online», в которой часто сходятся в битвах друг против друга сотни, если не тысячи, игроков. В данной игре сервер обрабатывает данные о взаимодействии игроков, а за графику отвечает персональный компьютер клиента.

Обычно в таких случаях обработка данных с локации перебрасывается на отдельный сервер, и это обеспечивает большую производительность и масштабируемость. Тем не менее, зачастую этого мало, и потому сильно страдает качество игры.

Новая концепция облачного игрового движка тоже будет подвержена подобной проблеме, однако в значительно меньшей степени. Нет необходимости передавать огромные массивы данных о синхронизации действий и эффектов по сети между игроками через сервер. Эти данные будут существовать лишь в

пределах самого сервера, где скорости значительно выше, а масштабируемость лучше. При огромных волновых нагрузках такой сервер будет куда устойчивей, а просадка должна быть менее заметна. Как результат, пользователи не будут ощущать дискомфорт и смогут насладиться настоящей масштабностью и красотой игрового мира без старых ограничений.

Следует упомянуть, что мощность современных вычислительных систем быстро растет и подчиняется закону Мура. Это позволит в обозримом будущем создать реалистичную физическую модель поведения. Люди смогут полностью реализовать любые идеи в игровых вселенных, где будут действовать законы физики, аналогичные реальному миру. Уже сейчас активно развиваются игры-песочницы с реалистичной графикой, однако с улучшением графики уменьшается их количество. Это тоже связано с ростом передаваемых данных между клиентскими машинами, проблему которой и решает новая концепция.

Последним, тем не менее важным преимуществом концепции облачного игрового движка, является невозможность взлома. В прошлое уйдет пиратство и изменение игровых данных с целью получения преимущества над другими игроками. Благодаря закрытости данных система полностью защищена от подобного рода воздействия.

Основные недостатки описанной технологии. В современных многопользовательских играх преобладают две версии монетизации: free to play (бесплатная игра с внутриигровыми покупками) и pay to play (оплата по подписке каждый месяц). Первая сейчас очень распространена и активно продвигается, но она долго не сможет существовать в новой концепции из-за дороговизны поддержки. А вот вторая отлично подходит. Разработчики точно смогут рассчитать стоимость обслуживания и установить соответствующую плату для подписки.

Для минимизации задержек ввода-вывода необходимо разместить сервера как можно ближе к клиенту, что увеличит затраты и создаст серьезные затруднения в реализации кросс-серверной игры из-за архитектуры облачного движка. Однако благодаря быстрым темпам развития коммуникационных технологий данная проблема скоро может стать неактуальной.

На данный момент стоимость обслуживания серверов, разработки и поддержки низкоуровневого программного обеспечения, а также сложности монетизации делают разработку инновационного облачного движка практически нерентабельной.

Перспективы развития облачных технологий в игровой индустрии. Создание первых облачных игровых движков может затянуться из-за производителей аппаратной составляющей домашних компьютеров и серверов. Nvidia, Intel и AMD – крупные игроки, которые не захотят терять большую долю рынка комплектующих. Необходимость обновлять свой персональный компьютер, чтобы иметь возможность играть в самые современные проекты, попросту исчезнет. Все это ляжет исключительно на плечи издателя и разработчика.

«Железо» серверов для облачных игр обойдется значительно дешевле, чем аналогичное для персональных компьютеров. Единственным возможным вариантом будет искусственное повышение цен на серверные комплектующие, что тоже может сильно замедлить развитие облачных игровых движков.

Активное развитие может получить популярная в настоящее время виртуальная реальность (VR). Однако для ее реализации требуется сократить задержку между вводом и выводом, т.к. это вызывает негативное воздействие на самочувствие игрока. Из-за относительно низкой скорости передачи данных по сети, а также высокой задержки требуется быстрый скачок в развитии способов передачи данных для реализации VR в облачных игровых движках.

Выводы. В связи с последними тенденциями в развитии программного обеспечения и коммуникационных технологий нет никаких сомнений в скорой революции в сфере облачных игровых технологий и сервисов. Скорость передачи, задержка и стоимость обслуживания оборудования позволит совершить качественный скачок уже в обозримом будущем. Облачные технологии находятся в ранней стадии развития, однако их потенциал в мире развлечений не имеет себе равных. Уже сейчас идет активная борьба между корпорациями в этой сфере. Данные и расчеты перенесли в облака, лишив пользователей необходимости задумываться о замене своих устройств на более производительные, что спровоцировало гонку серверов. Игры являются одним из крайне востребованных продуктов на рынке программного обеспечения и создание облачных игровых движков просто неизбежно.

Список литературы

1. Облачные технологии для земных пользователей. [Электронный ресурс]. - Режим доступа: <http://sonikelf.ru/oblachnye-texnologii-dlya-zemnyx-polzovatelej/> – Облачные технологии. Обо всем, понемногу.
2. Прогноз игровой индустрии на завтра: облачно, местами кэш. [Электронный ресурс]. - Режим доступа: <http://gmbbox.ru/hardware/prognoz-igrovoy-industrii-na-zavtra-oblachno-mestami-kesh> – Впечатления от игры.

УДК 519.688

**Разработка приложения для организации спортивных мероприятий по системе с одним (кубковая) или двойным выбыванием (double elimination)
Кацель С.Л., Савицкая И.В.**

Государственное образовательное учреждение высшего профессионального образования

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа

E-mail: peacemaker.donetsk@gmail.com

Аннотация

Кацель С.Л. Разработка приложения для организации спортивных мероприятий по системе с одним (кубковая) или двойным выбыванием (double elimination). Рассмотрен процесс разработки программного продукта для иностранного заказчика для облегчения организации и проведения спортивных мероприятий по игре корнхолл. Изучены вопросы построения основной и дополнительной турнирных сеток, методы распределения матчей по игровым площадкам.

Исходная постановка задачи. Некоторое время назад было получено обращение заказчика с предложением разработать программное обеспечение для организации и проведения соревнований по игре корнхолл. Существующие решения заказчика не устраивали, т.к. либо требовали доступ в сеть Интернет для работы (различные онлайн-генераторы турнирных сеток), либо были довольно дороги, либо не обладали всем необходимым функционалом. Поэтому было решено разработать десктопное приложение, не требующее подключения к сети Интернет, и сочетающее в себе все функции, необходимые заказчику, в одном месте. Были определены следующие системные требования: операционная система Windows XP или выше, фреймворк .NET версии 4.0, построение форм приложения – с использованием WinForms, язык программирования – C#. Требования к функциональности: создание или загрузка списка команд из файла (количество команд – от 3 до 128 включительно); генерация основной и дополнительной (в случае создания турнира double elimination) турнирных сеток, распределение команд по основной сетке; ввод результатов и отображение продвижения команд по турнирным сеткам; составление расписания игр и распределение матчей по игровым площадкам; печать расписания и направлений для команд на конкретный матч (матч-тикетов); сохранение и загрузка турнира в файл/из файла.

Реализация. Для турнира, построенного по т.н. «Олимпийской системе» (с одиночным выбыванием), равно как и для турнира по схеме с двойным выбыванием, оптимальное число команд составляет $2^N - 2$, 4, 8, 16 и т.д. - в таком

случае турнирная сетка заполняется без пробелов, и в основной сетке все команды играют одинаковое количество игр. Однако такая ситуация бывает далеко не всегда, и в общем случае количество команд не оптимально. В таком случае список команд дополняется до ближайшего числа, равного степени двойки «пустыми» командами, которые не участвуют турнире. Команда, которая получила в соперники такую фантомную команду, считается победившей и автоматически перемещается в следующий раунд соревнований. Это автоматически вводит в турнир такое понятие, как «рейтинг» - т.е. команда (или участник), которая находится в списке выше (имеет более высокий рейтинг), имеет больше шансов в первом неполном раунде не принимать участия и перейти сразу во второй.

Генерация основной сетки турнира. Построение основной сетки турнира начинается с «корня» турнирного дерева – одной игры, предельный случай, когда в турнире всего два участника. При этом сумма номеров команд в матче равна количеству участников на данном этапе плюс один (это правило справедливо для всех матчей в основной сетке турнира). К этому матчу добавляются два матча как «листья» бинарного дерева, имеющие «подчинённое» отношение к первому матчу. Команды из исходного матча перемещаются по одной в эти дочерние матчи.

Для дочерних матчей необходимо добавить вторую команду, чтобы узнать её номер, необходимо, как мы помним, отнять от количества участников на данном этапе отнять номер команды, уже имеющейся в матче, и прибавить единицу (рис. 1). Этот процесс продолжается, пока количество уровней в турнирном дереве не станет равно $\min \log_2 N; N \geq K$, где K - количество участников в турнире.



Рис. 1. Порождение дочерних матчей и вычисление номера новых команд.

Если турнир создаётся по системе с одиночным выбыванием, то на этом этапе есть возможность добавить матч за 3-е место, в который будут добавлены команды, проигравшие полуфинальные матчи.

Генерация дополнительной сетки. Если создаётся турнир по схеме с двойным выбыванием, то команда, проигравшая один матч, не покидает соревнование, а переходит в дополнительную сетку, где продолжает играть, пока не потерпит второе поражение. Отличие дополнительной сетки от основной в том, что команды, попавшие в неё, сначала играют между собой, после чего в следующем победитель получает в соперники команду, выбывшую из основной сетки на раунд позже (рис. 2)

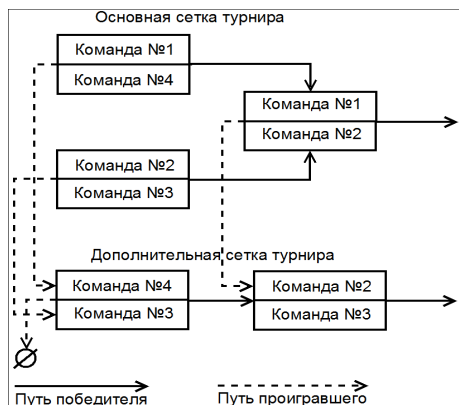


Рис. 2 Передвижение победивших и проигравших команд по турниру с двойным выбыванием.

Достоинства такого рода турниров в том, что команда, даже проиграв один раз, сохраняет шансы на победу в турнире, однако для этого ей придётся сыграть практически вдвое больше матчей, чем победителю основной сетки соревнования.

Распределение матчей по игровым площадкам (кортам)

При распределении матчей по кортам первоначально был использован метод «поиска в ширину» для обеспечения равномерной загрузки игровых площадок. Однако в ряде случаев в расписании матчей на конкретном корте возникали перерывы, т.к. время начала матча зависело от времени начала и завершения матчей, победители (или проигравшие) которых должны принимать участие в конкретном матче. Заказчика такой вариант не устроил, и для распределения матчей был использован «поиск в глубину». Такой подход позволил избавиться от разрывов в расписании матчей, однако теперь нагрузка игровых площадок была крайне неравномерной, и максимальное количество игр приходилось на первые площадки в списке доступных, что было полностью одобрено заказчиком, т.к. упрощало организацию размещения болельщиков и наблюдателей.

Для облегчения проведения соревнования была реализована система распечатки т.н. «матч-тикета», в котором указывались названия команд, принимающих участие в конкретном матче, его порядковый номер, номер игрового корта и время начала игры. Распечатка выполняется на бумаге с перфорацией (рис. 3), после чего отдельные «матч-тикеты» выдаются соответствующим командам. После завершения матча результат вносится в этот же «матч-тикет», который сдаётся организаторам для внесения в программу.



PF43 - 2 5/6" x 1 3/8" - 3 Columns & 8 Rows

Рис. 3 Бумага с перфорацией для распечатки «матч-тикето»

Выводы. В результате проделанной работы было разработано приложение, позволившее упростить и автоматизировать подготовку организационной структуры спортивных соревнований по системе с одним или двойным выбыванием. Реализованы отображение и печать основной и дополнительной турнирных сеток в текущем виде, учёт и корректная обработка результатов матчей, отображение и печать расписания, печать «матч-тикето» для матчей.

Для отображения турнирных сеток были созданы два простых элемента управления «матч» (содержащий всю доступную информацию о нём) и «соединительная линия» для ускорения отрисовки вместо использования GDI-графики Windows.

Достоинства приложения – очень малый размер бинарного файла (всего 72,5 КБ), отсутствие необходимости соединения с сетью Интернет, компактное сохранение текущего состояния турнира в файл путём бинарной сериализации. Недостатки – разработанное приложение разрабатывалось под требования заказчика и потому не совсем универсально: невозможна организация многодневного турнира; невозможен учёт результатов матчей с «теннисной» и подобными нотациями; при создании турниров с количеством участников более

40 отрисовка турнирных сеток может выполняться несколько секунд, что обусловлено ограничениями технологии WinForms.

Список литературы

1. Алексеев С.: Спортивный менеджмент. Регулирование организации и проведения физкультурных и спортивных мероприятий. - Советский спорт, 2015. 687 с.
2. Олимпийская система с выбыванием после двух поражений. / Интернет-ресурс. – Режим доступа: [www/URL:https://ru.wikipedia.org/wiki/Олимпийская_система_с_выбыванием_после_двух_поражений](http://www.URL:https://ru.wikipedia.org/wiki/Олимпийская_система_с_выбыванием_после_двух_поражений). – Загл. с экрана.
3. Системы проведения соревнований. / Интернет-ресурс. –Режим доступа: [www/URL:http://tennis-i.com/tennisnaya-entsiklopediya/sudejstvo-tennisa/sistemy-provedeniya-sorevnovanij.html](http://www.URL:http://tennis-i.com/tennisnaya-entsiklopediya/sudejstvo-tennisa/sistemy-provedeniya-sorevnovanij.html). – Загл. с экрана.

УДК 004.056.5

Проектирование обфускатора для языка JavaScript Медгаус С.В., Чернышова А.В.

Донецкий национальный технический университет,
кафедра программной инженерии
medgaus-sergey@yandex.ru, alla@pmi.dgtu.donetsk.ua

Аннотация

Медгаус С.В., Чернышова А.В. Проектирование обфускатора для языка JavaScript. В тексте данной статьи описан процесс обфускации и существующие обфускаторы для языка JavaScript. Исходя из анализа существующих обфускаторов, была предложена разработка обфускатора языка JavaScript с открытым кодом, реализующая сложные преобразования исходного кода, а также основные алгоритмы обфускации, которые будут использоваться в реализации программного продукта.

Общая постановка проблемы. На сегодняшний день проблема сохранения прав на владение интеллектуальной собственностью стоит очень остро. В то время как настольные приложения могут обеспечить себя различными средствами лицензирования и защиты, веб-приложения не могут защитить свой код, расположенный на стороне клиента от просмотра и изучения.

Частично эту проблему решает перенос особенно важных частей кода на сторону сервера, что повлечёт за собой увеличение нагрузки на сервер и может потребовать увеличения мощности сервера или покупки дополнительного оборудования.

Также решению этой проблемы может поспособствовать обфускация программного кода, передающегося клиенту. Обфускация не решит эту проблему полностью, так как возможность использовать реверс-инжиниринг остаётся, но вот потенциальная сложность деобфускации и большие человеческие ресурсы должны отбить желание у злоумышленников разбирать чужой код и извлекать из него алгоритмы.

Обфускация (от лат. obfuscare — затенять, затемнять; и англ. obfuscate — делать неочевидным, запутанным, сбивать с толку) или запутывание кода — приведение исходного текста или исполняемого кода программы к виду, сохраняющему его функциональность, но затрудняющему анализ и понимание алгоритмов работы.

Существуют специальные программы, производящие обфускацию, так называемые обфускаторы [1].

Существующие обфускаторы. На данный момент существует несколько обфускаторов для JavaScript, но необходимо отметить, что все обфускаторы кроме Google Closure Compiler являются в первую очередь упаковщиками или минификаторами (программы, которые минимизируют размер исходного кода, выполняют оптимизацию для увеличения производительности), а уже как следствие, они являются обфускаторами. Были рассмотрены такие популярные упаковщики как YUI Compressor [2] и Packer [3].

Однако данные программы не являются обфускаторами, которые выполняют сложные преобразования над исходным кодом программы, как упоминалось выше.

Проектирование собственного обфускатора. Так как в настоящее время нет обфускатора с открытым исходным кодом, который сможет предоставлять продвинутые способы обфускации, то создание такого обфускатора для языка JavaScript является актуальным.

После анализа существующих обфускаторов для языка JavaScript, была разработана диаграмма прецедентов для будущего обфускатора (см. рис. 1). На ней изображён один актёр (Пользователь), который может загрузить файл для обфускации, включить нужные продвинутые режимы обфускации и обфусцировать файл. После чего, может сохранить запутанный файл.

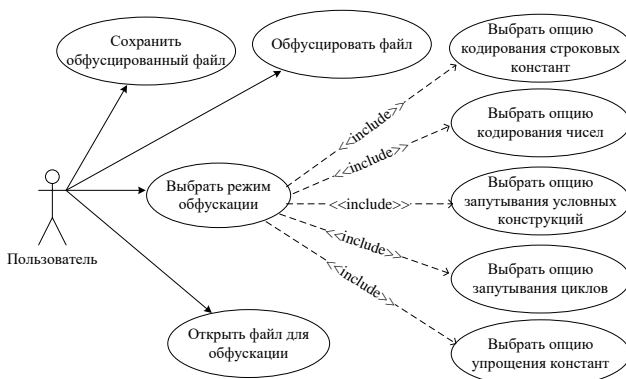


Рис. 1. UML диаграмма прецедентов

Исходя из диаграммы прецедентов, была разработана диаграмма классов (см. рис. 2), на которой отображена основная структура проектируемого приложения.

Пользователь будет напрямую взаимодействовать с классом MainForm, так как все методы для манипуляции исходным файлом (загрузка в программу, обфускация и сохранение) будут находиться в нём. Также в этом классе будет предоставлена возможность пользователю выбрать режим обфускации с помощью флажков, расположенных на форме. Класс MainForm содержит в себе класс Obfuscator, который и предоставляет возможность обфусцировать исходный код.

Класс Obfuscator содержит список объектов, реализующих функциональный интерфейс Mangler, который содержит всего один метод – обфусцировать исходный код, а уже реализации этого интерфейса конкретизируют, как именно они будут запутывать код. Этот подход обеспечивает унифицированность при работе с объектами-запутывателями, позволяя вызывать один метод для каждого объекта при разной реализации (принцип полиморфизма). Более того, полиморфизм позволяет с лёгкостью добавлять новые алгоритмы обфускации, что намного упрощает процесс разработки.

Далее была разработана диаграмма состояний для демонстрации состояний главной формы приложения, в которых она может находиться в процессе функционирования (см. рис. 3). Таких состояний несколько: ожидание действий пользователя, загрузка исходного кода из файла, обфускация исходного кода и сохранение обфусцированного кода в файл.

UML Class Diagram - JavaScript Obfuscator

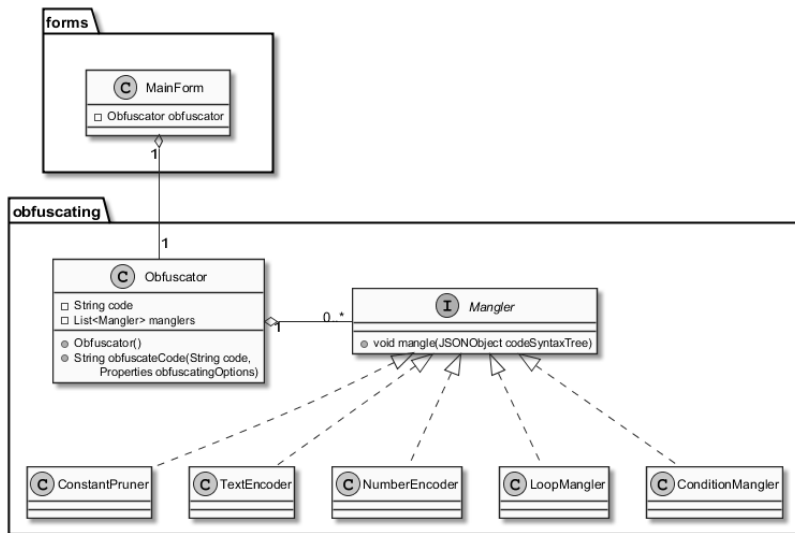


Рис. 2. UML диаграмма классов

UML State Diagram for MainForm

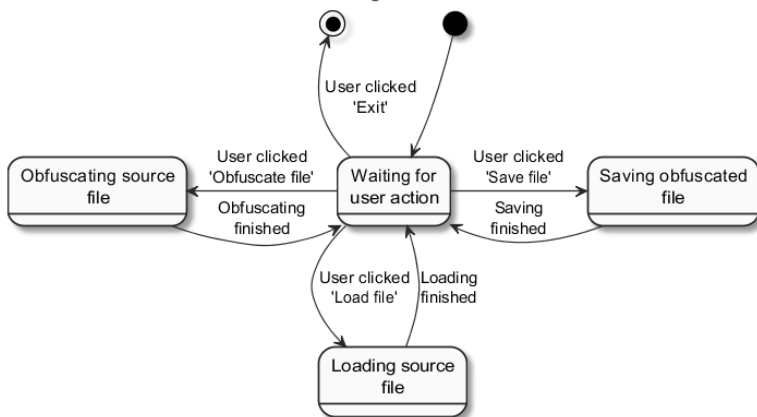


Рис. 3. UML диаграмма состояний

Для того чтобы проводить обфусцирующие преобразования с кодом необходимо получить абстрактное синтаксическое дерево (АСД). Эту задачу отлично выполняет библиотека от JQuery Esprima [4]. Она позволяет получить АСД из исходного кода JavaScript в формате JSON (см. рис. 4). Эту библиотеку можно использовать и для генерации кода по АСД, что значительно упрощает проведение обфускации.



Рис. 4. Пример разбора исходного кода

Способы обфускации

Планируется реализовать следующие алгоритмы запутывания:

- подстановка константных переменных в коде;
- запутывание условных конструкций;
- преобразование циклов;
- кодирование строковых и числовых констант.

Подстановка константных переменных заключается в том, чтобы заменить использованные константы в коде на их значение, таким образом появятся «магические числа», которые затрудняют понимание кода.

Способ запутывания условных конструкций сводится к добавлению «непрозрачных предикатов», значение которых известно на этапе запутывания, но его трудно узнать на этапе разбора кода. Также используется вызов функций, которые возвращают случайные булевы значения. Их можно использовать таким образом, чтобы они не влияли на общее значение логического выражения, но в то же время мешали ясному пониманию кода.

Преобразование циклов включает в себя несколько различных видов преобразований. Первый вид – это трансформация циклов, заключающаяся в том, чтобы один цикл разбить на несколько, просто разделив между ними итерации. Второй – это развёртка циклов. Суть этого вида преобразования в том, что если на этапе разбора кода мы знаем, сколько итераций цикл будет выполняться, то в процессе обфускации мы можем вынести инструкции тела цикла за цикл и повторить там несколько раз, уменьшив при этом итерации самого цикла. И последний вид преобразований циклов – это разделение циклов. Это преобразование заключается в том, чтобы разделить инструкции исходного цикла

и поместить в два цикла с исходным количеством итераций. Для большего эффекта запутывания рекомендуется применять все виды преобразований для циклов [5].

Кодирование строковых констант может включать в себя: преобразование констант в кодировку BASE64; добавление новых функций, которые могут «собирать по частям» константы; записывание строк в виде кодов Unicode-символов, которые можно представлять в виде целых чисел и впоследствии можно запутать.

Кодирование числовых констант достаточно сложное преобразование. Это преобразование рационально разделить на преобразование целых и вещественных чисел. Алгоритм кодирования числовых констант представлен на рисунке 5.



Рис. 5. Алгоритм кодирования числовых констант

Выводы. В данной работе рассмотрены понятия обфускатора и обфускации. Также спроектированы UML диаграммы прецедентов, классов и состояний для разрабатываемого обфускатора. Рассмотрена возможность использования вспомогательной библиотеки Esprima для разбора исходного кода и генерации кода по АСД. Уточнены алгоритмы обфускации, а также способы их реализации.

Список литературы

1. Википедия. Обфускация [электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Обфускация>
2. YUI Compressor [электронный ресурс]. – Режим доступа: <https://github.com/yui/yuicompressor>
3. Packer [электронный ресурс]. – Режим доступа: <http://dean.edwards.name/packer/>
4. Esprima [электронный ресурс]. – Режим доступа: <https://www.esprima.org/>
5. LynX. Обфускация и защита программных продуктов // CITForum. [электронный ресурс]. – Режим доступа: <http://citforum.ru/security/articles/obfus/>

УДК 004.91

Обзор и сравнение программ для чтения текстовых файлов для ОС Android

Серёженко О.А., Коломойцева И.А.

Донецкий национальный технический университет
кафедра программной инженерии
E-mail: seryozhenko-94@mail.ru

Аннотация

Серёженко О.А., Коломойцева И.А. Обзор программ для чтения текстовых файлов для ОС Android. В данной статье описаны 4 самых популярных программ для чтения текстовых файлов под ОС Android, их свойства и возможности: “FBReader”, “Cool Reader”, “Moon+ Reader”, “AlReader”.

Введение. Мобильные устройства давно перестали быть чем-то необычным и великолепно справляются со своей функцией – являются средством коммуникации между людьми. При этом, недавно появившиеся, но уже прочно вошедшие в нашу жизнь смартфоны настолько функциональны, что трудно сказать, чего они не умеют: это и плеер, и фотоаппарат, и возможность использования Интернет-ресурсов, и прочее.

Одним из распространенных применений мобильных устройств является их использование в роли устройства для чтения текстовых файлов – «читалки».

Школьники устанавливают на смартфон «читалку» для освоения школьной программы, студенты – для продуктивной работы со справочной литературой, аспиранты – для получения новой информации в компактном виде, а все остальные – для развлечения и расширения своего кругозора.

Такие «читалки» сегодня пребывают на пике популярности, а современная молодежь таким прогрессивным методом расширяет свой кругозор и увеличивает словарный запас.

FBReader

Данная программа пользуется наибольшей популярностью среди пользователей. Программой поддерживаются:

- основные форматы: fb2 (и fb2.zip), ePub, Kindle;
- другие форматы: rtf, html, plain text;
- при установке дополнительных плагинов появляется поддержка PDF и DjVu.

При первом открытии пользователь попадает на чтение встроенной "книги" о FBReader-е, в ней можно прочитать краткую инструкцию как пользоваться программой. В дальнейшем в результате запуска программы пользователь будет попадать на последнюю читаемую им книгу, в том месте, где он закончили чтение.

Программа имеет такие преимущества:

- первоначальный простой внешний вид, без красивых книжных полок, которые можно скачать отдельным плагином (что положительно сказывается на экономии заряда батареи);
- простое управление и отсутствие большого количества сложных настроек (многое уже по умолчанию настроено так, как надо);
- высокая скорость работы;
- возможность формирование библиотеки текстов с любыми критериями сортировки или поиска книги по наиболее популярным мировым онлайн библиотекам;
- возможность создания закладок;
- наличие ночного режима;
- возможность чтения вслух;
- дополнительные плагины скачиваются и устанавливаются бесплатно [1].

Недостатков в программе не обнаружено.

На рис. 1 показан пример работы программы.

Moon+ Reader. Имеет две версии приложения – платная и бесплатная. Официальная, за которую не нужно платить может читать все электронные форматы книг.

Программа подойдет в первую очередь для тех пользователей, кому необходимо максимально возможное количество функций, настроек и поддерживаемых форматов. Также программу наверняка оценят и не слишком требовательные пользователи – она довольно простая в управлении и сходу предоставляет все нужное без дополнительных регулировок [2].

Преимуществами программы являются:

- простой и удобный интерфейс;
- поддержка бесплатной библиотеки;
- широкие возможности настроек.

Недостатки программы:

- отсутствие голосового чтения книги;
- программа не показывает некоторые картинки;
- наличие рекламы, которую можно убрать в платной версии. Купив

«Moon+ ReaderPRO», появится ряд дополнительных функций, таких как: двухстраничный режим, чтение книги вслух при встряхивании, управление через гарнитуру.

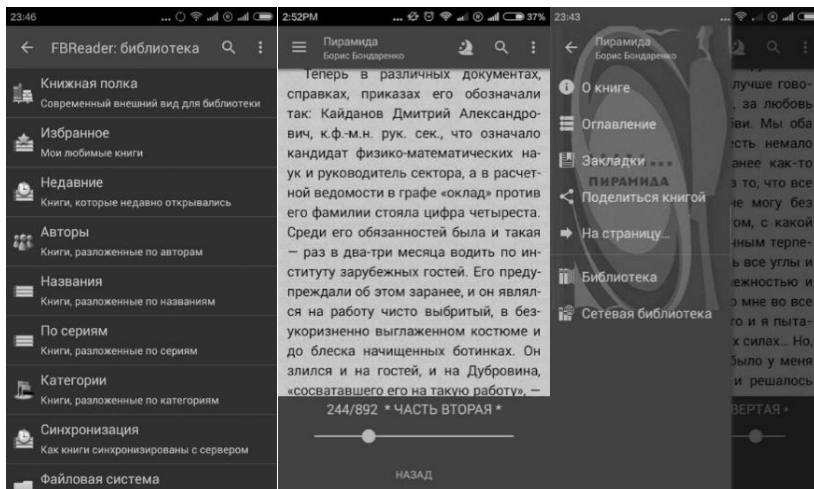


Рис. 1. Пример работы программы «FBReader»

На рис. 2 показан пример работы программы.

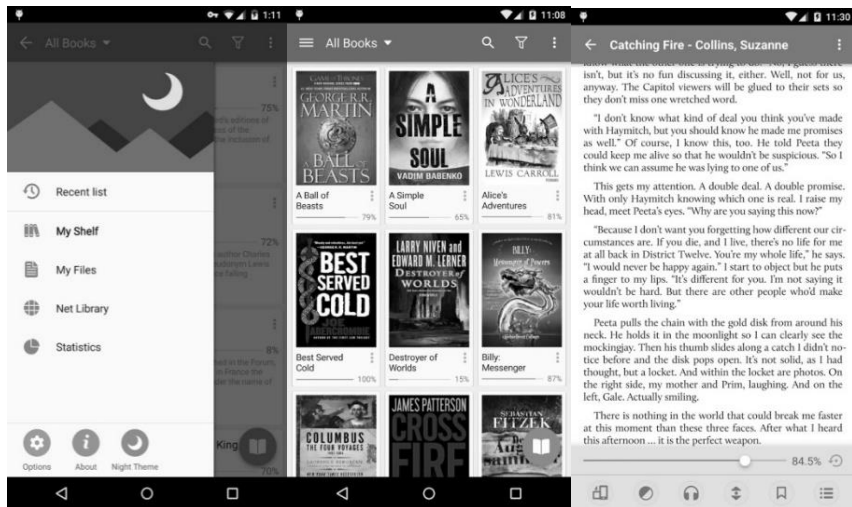


Рис. 2. Пример работы программы «Moon+ Reader»

Cool Reader. Приложение имеет не самый интуитивный интерфейс, однако если потратить немного времени на его настройку, можно получить удобную адаптированную программу для чтения.

Программа не содержит встроенных магазинов, однако заранее загруженную книгу можно открыть из любой директории устройства.

Преимущества программы:

- распознает все форматы книг.
- автоматически разбивает книги на категории: по авторам, по алфавиту, по дате загрузки. Это очень удобно при поиске интересных электронных книг.
- автоматически создается закладка страницы;
- красивая анимация при перелистывании страниц;
- поддержка онлайн-каталога;
- возможность установить на мобильное устройство со старой прошивкой;
- возможность чтения вслух.
- Недостатками являются:
- периодическое зависание;
- не сохранения настроек;
- закрытие самой по себе программы, не сохраняя закладку, поэтому приходится искать вручную место, где остановился.

На рис. 3 показан пример работы программы.

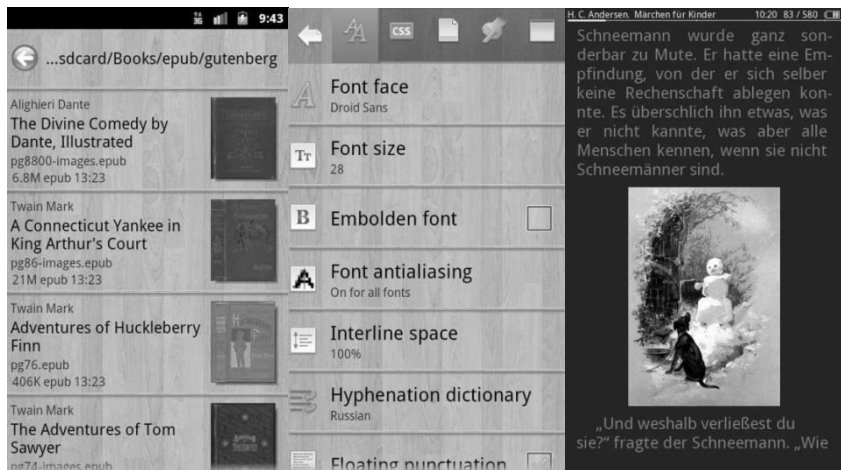


Рис. 3. Пример работы программы «Cool Reader»

AlReader. В программе по умолчанию установлен легко воспринимаемый с расстояния большой шрифт. Если же пользователя не устраивают чересчур крупные буквы, их размер всегда можно отредактировать – для регулировки шрифтов достаточно сделать характерное щипковое движение. Также можно настроить удобную опцию, которая отображает в процентном и абсолютном выражении количество прочитанных страниц.

- качественное внешнее оформление;
- хорошая функция регулирования подсветки в ночном режиме;
- много разных обоев;
- большое количество настроек;
- возможность скачивать книги на предоставляемом интернет-ресурсе.

Недостатков в программе не обнаружено.

На рис. 4 показан пример работы программы.

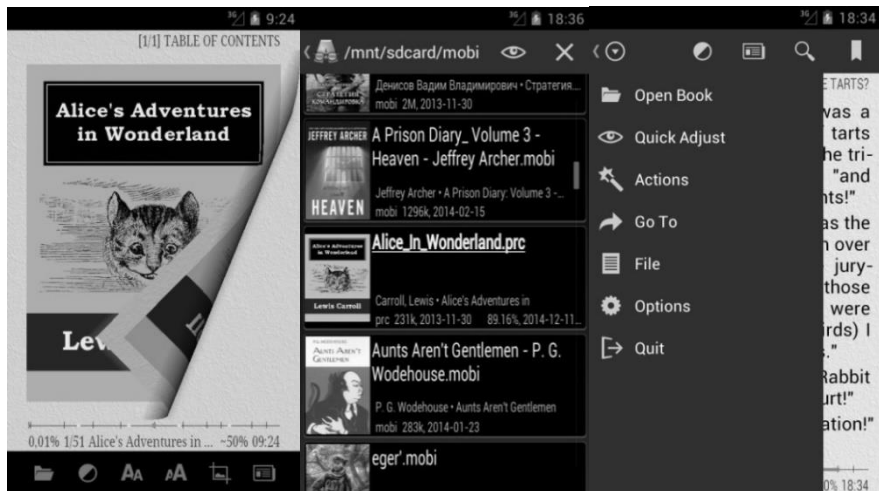


Рис. 4. Пример работы программы «AIReader»

Выводы. Можно выделить три момента, которые должны присутствовать в хорошей «читалке». Во-первых, это настройки внешнего вида. Сюда можно включить работу с текстом, стилями, фоном, дополнительными панелями, анимацией. Во-вторых, не менее важными являются настройки управления и системы. Настраиваемые действия для зон экрана, использование всех аппаратных кнопок, быстрый доступ к настройкам, программное изменение яркости, - все эти мелочи делают чтение максимально удобным. Наконец, третье место отводится красивой или хотя бы удобной книжной полке. Ее наличие необязательно, но в таком случае приложение должно обладать отличным набором настроек и поддерживать все популярные форматы, чтобы «компенсировать» недостатки интерфейса [3].

Но ни в одной из исследованных программ не была реализована функция поиска в памяти устройства и вывода списка файлов определённых форматов и информации о них. В ходе дипломной работы будет разработана программа для чтения текстовых файлов для ОС Android, в которой будет реализована вышеуказанная функция.

Список литературы

1. Обзор лучших читалок книг на Андроид [электронный ресурс]. – <http://x-up.ru/obzor-luchshix-programm-chitalok-knig-na-android/>
2. ТОП-5 Android-приложений для чтения книг [электронный ресурс]. – http://www.migom.by/news/top-5_android-prilogeniy_dlya_chteniya_knig/
3. Читалки для Android – выбираем лучшую [электронный ресурс]. – <http://xdrv.ru/articles/software/221/full>

УДК 004.522

Речевой интерфейс для интеллектуализации ввода компьютерных программ

Бакаленко В.С., Федяев О.И.

Донецкий национальный технический университет
fedyaev@r5.dgtu.donetsk.ua, valeriy.bakalenko@gmail.com

Бакаленко В.С., Федяев О.И. Речевой интерфейс для интеллектуализации ввода компьютерных программ. В статье описывается система речевого ввода текста программы на языке программирования. Речевой интерфейс представляется программистам более естественный способ набора исходного кода программы. Ядром интерфейса является акустико-лингвистическая модель автоматического распознавания речи.

Введение. В настоящее время набор текста программы на языке программирования выполняется вручную с помощью клавиатуры. Процедура ввода требует хороших навыков работы с клавиатурой, большого внимания и напряжения на зрение. Такой способ ввода для человека является трудоёмким и не совсем удобным [1-3]. Этот недостаток можно устранить путём качественного решения задачи автоматического распознавания речи [4].

Основными проблемами в решении этой задачи являются: большой объём словаря используемых слов; высокая скорость распознавания; качество распознавания. Кроме того, необходимо учитывать многообразие вариантов названий переменных и функций. Скорость распознавания должна быть приемлемой, чтобы программист мог диктовать код и видеть в реальном времени предварительные результаты распознавания [5]. От точности распознавания зависит результат компиляции исходного кода и время исправления ошибок.

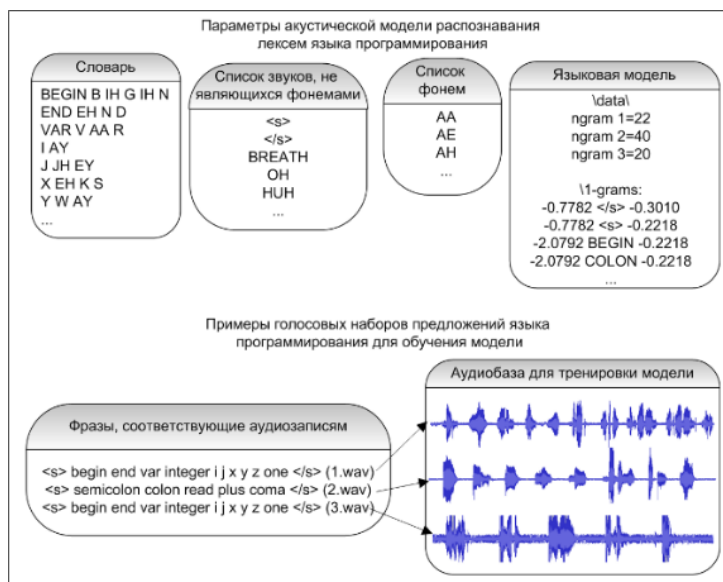
Поэтому цель данной работы состоит в разработке речевого интерфейса, обеспечивающего взаимодействие человека с компьютером более естественным способом. Задача ввода текста программ голосом решается с помощью технологии Sphinx. Она обеспечивает возможность построения акустико-лингвистической модели для любого языка программирования и настройки на особенности голоса диктора.

Инструментальная среда CMU Sphinx4. На сегодняшний день Sphinx [4], является самым популярным и работоспособным из открытых движков. Он разработан в университете Карнеги-Меллона с участием Массачусетского технологического института и Sun Microsystems. CMU Sphinx распространяется на условиях лицензии BSD, т. е. доступен для коммерческого и некоммерческого использования. Одним из достоинств Sphinx является поддержка множества языков и способность создавать свои собственные модели распознавателей речи.

Sphinx-4, как версия из семейства CMU Sphinx [6], состоит из двух компонентов: «тренера» (trainer) и декодера. Тренер создаёт акустическую модель, адаптированную под конкретные потребности, а декодер выполняет собственно распознавание. Архитектура Sphinx-4 на верхнем уровне относительно проста. Как показано на рисунке 1, она включает FrontEnd, клиентскую часть (приложение), декодер и базу знаний.

Блок FrontEnd отвечает за сбор, аннотирование и обработку входных данных. Кроме того, он извлекает объекты из входных данных для чтения с помощью декодера. Аннотации, предусмотренные во FrontEnd, включают в себя начальный и конечный сегменты данных. Операции, выполняемые на данном этапе, реализуют шумоподавление, автоматическую регулировку усиления, анализ Фурье, спектральную фильтрацию Мэла и др.

База знаний содержит информацию необходимую для декодера. Эта информация включает в себя акустическую модель и модель языка. В свою очередь декодер может послать команду базе знаний, требующую от базы знаний динамически изменять себя на основе результатов поиска. Эти модификации могут заключаться в переключении акустических моделей и/или языка модели, а также обновлении некоторых параметров, например, дисперсии преобразования для акустических моделей.



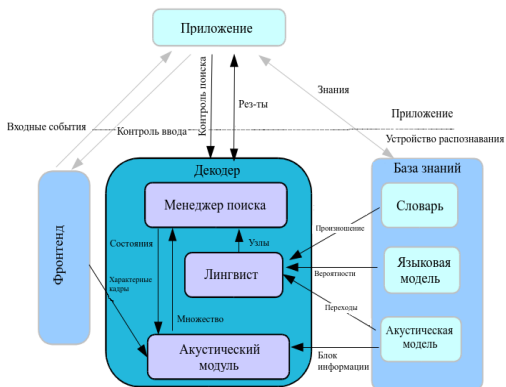


Рис. 1 – Структура инструментария CMU Sphinx4

Декодер выполняет основную часть работы. Он считывает данные с FrontEnd, сопоставляет их с данными из базы знаний и откликом приложения, а также выполняет поиск в пространстве наиболее вероятных последовательностей слов, которые входят в число претендентов на выбор. Термин «пространство поиска» означает описание наиболее вероятных последовательностей слов, которые динамически обновляются с помощью декодера в процессе декодирования.

В отличие от множества других архитектур распознавателей речи Sphinx-4 позволяет приложению контролировать множество функций речевого движка. Во время декодирования приложение может получать данные от декодера. Эти данные позволяют приложению отслеживать, как происходит процесс декодирования, а также влиять на процесс декодирования до его завершения. Кроме того, приложение может обновлять базу знаний в любое время.

Но главным достоинством Sphinx является возможность описания проектируемого распознавателя на уровне формальных моделей, что и послужило основанием для выбора Sphinx-4 в качестве инструментария.

Лингвистическая модель является основой построения приложения распознающего и синтезирующего речь.

Лингвистическая модель состоит из словаря, языковой модели, списка фонем и прочих звуков, обучающего множества. Она служит входными данными для построения акустической модели. В словаре содержится список слов и транскрипции к ним. Транскрипции должны состоять исключительно из фонем, которые присутствуют в списке фонем. Помимо слов есть и другие звуки, не несущие в себе смысловой нагрузки: звуки дыхания, различный шум. Языковая модель – совокупность вероятностей появления слов в речи. В соответствии со всеми вышеперечисленными характеристиками речи записывается аудиобаза и

она оформляется в виде обучающих примеров с правильными ответами. Каждая аудиозапись должна иметь своё текстовое представление. Чем больше материала в аудиобазе, тем лучше качество распознавания. Элементы информационной структуры лингвистической модели показаны на рисунке 2.

Общая схема распознавания и синтеза речи. Взаимодействие человека и компьютера осуществляется с помощью клавиатуры, мыши, монитора, микрофона и колонок. С помощью микрофона и подсистемы отвечающей за распознавание речи пользователь может вводить голосом тексты программ на языке программирования Pascal. Также имеется возможность производить ввод текста программы с помощью клавиатуры. Речевой синтез текстов программ Pascal осуществляется с помощью колонок или наушников и при помощи подсистемы синтеза речи. Кроме того, можно настроить скорость и громкость чтения текстов программ.

Человек отвечает за ввод текстов программ с помощью клавиатуры либо микрофона. Компьютер занимается распознаванием речи, ее синтезом и выводом результатов на формы приложения, отображаемые на мониторе. Во время работы, приложение выводит различную служебную информацию о ходе распознавания речи.

Благодаря разработанному программному продукту пользователь может вводить тексты программ не только с помощью клавиатуры, но и с помощью речи. Чтобы не напрягать зрение и проверять программу, пользователь может просто прослушать написанную им программу.

Функциональная схема разработанного приложения изображена на рисунке 3.



Рис. 3 – Общая схема речевого интерфейса

Разработка речевого интерфейса. Разрабатываемый интерфейс внешне выглядит как приложение-блокнот, сохраняющий возможность ввода текста с помощью клавиатуры, а так же позволяющий вводить тексты программ с помощью речи, а затем сохранять их, озвучивать и отправлять на компиляцию [7].

Интерфейс приложения, изображенный на рисунке 4, состоит из строки заголовка, строки меню, рабочей области, кнопок для активирования режимов синтеза и распознавания речи, а также небольшой области с выводом сообщений о состоянии инструмента распознавания.

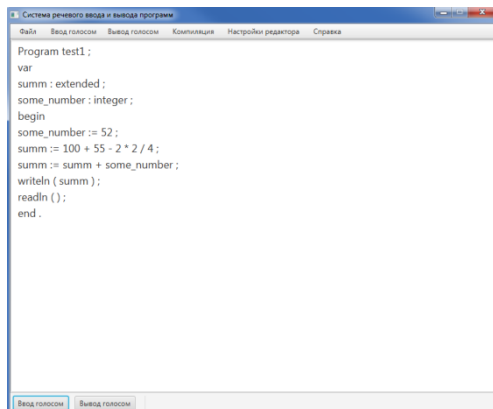


Рис.4– Интерфейс приложения распознавания и синтеза речи

Словарь в созданном приложении состоит из 100 основных слов. Из этих слов и составляются все тестовые предложения, примеры и обучающие множества.

В документации к Sphinx4 сказано, что кроме фонем, используемых для распознавания слов, существуют еще 3 фонемы, которые не являются звуками: <s> - начало предложения, <sil> - тишина в середине предложения, </s> - конец предложения. По сути, все эти фонемы обозначают тишину и отличаются лишь своим положением в сказанной фразе. Система определяет их сама и пользователю совершенно не надо о них беспокоиться.

В виду того, что грамматику Pascal нет возможности приспособить для создания грамматической модели используемой в Sphinx4, из-за использования правых и левых рекурсий, было решено использовать наиболее популярную языковую модель – N-граммную (рис.5) [8-10].

\1-grams:	\2-grams:	\3-grams:
-0.9382 </s> -0.3010	-2.8842 <@> AND -0.2553	-0.3010 <@> AND </@>
-0.9382 <s> -0.1647	-2.5832 <@> ARRAY 0.0000	-0.6021 <@> ARRAY </@>
-2.8224 AND -0.2255	-2.8842 <@> ASM 0.0000	-0.6021 <@> ARRAY LEFT-
-2.8224 ARRAY -	-1.8050 <@> ASSIGN	SQUARE-BRACKET
0.2472	0.0000	-0.3010 <@> ASM </@>
-3.2204 ASM -0.2478	-2.8842 <@> ASTERISK	-1.3802 <@> ASSIGN </@>
-2.2910 ASSIGN -	0.0000	-1.3802 <@> ASSIGN FALSE
0.2068	-2.5832 <@> AT 0.0000	-1.3802 <@> ASSIGN TRUE
-3.5214 ASTERISK -	-1.3802 <@> ASSIGN NOT
0.2478	-0.3010 ASM </@> -0.3010	-1.3802 <@> ASSIGN RANDOM
-3.2204 AT -0.2438	-1.2304 ASSIGN </@> -	SQUARE-ROOT
-3.5214 BACKSLASH	0.3010
-0.2478	-1.5315 ASSIGN FALSE	-0.3010 <@> DO </@>
-2.3173 BEGIN -0.2466	0.0000	-0.6021 <@> DOT </@>
-2.7432 BOOLEAN -	-1.5315 ASSIGN LEFT-	-0.6021 <@> DOT DOT
0.2365	PARENTHESIS -0.0458	-0.6021 <@> DOUBLE </@>
-2.7432 BYTE -0.2365	-1.5315 ASSIGN NIL	-0.6021 <@> DOUBLE
.....	0.0000	SEMICOLON
	-1.5315 ASSIGN NOT -
	0.1249	-0.3010 COLON ARRAY </@>
	-1.5315 ASSIGN RANDOM -	-0.3010 COLON BOOLEAN
	0.2218	SEMICOLON
	-1.5315 ASSIGN SQUARE-	-0.6021 CONSTRUCTOR NAME
	ROOT -0.0969	LEFT-PARENTHESIS
		-0.6021 CONSTRUCTOR NAME
		SEMICOLON

Рис. 5 – N-граммная языковая модель

Полученные результаты. Для тестирования разработанных моделей были написаны различные программы на языке программирования Pascal. Пример распознаваемых аудиофайлов, описывающих речевые произношение операторов исходного текста программы изображен на рисунке 6.

Входные данные для тестирования состоят из записанных фраз и их текстового представления для того, чтобы программный продукт мог их сопоставлять в ходе тестирования.

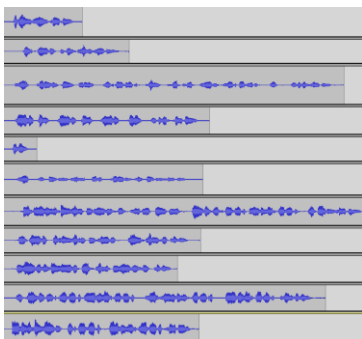


Рис. 6 – Примеры тестовых аудиофайлов

Одним из ключевых моментов является то, что в виде тестового примера приложению необходимо подавать заново записанные примеры, а не те аудиофайлы, на которых производилось обучение системы [13], чтобы не повлиять на качество результатов тестирования.

Суть эксперимента заключалась в определении достаточного количества обучающего материала для хорошего уровня распознавания системы. Грамотно составленная языковая модель сильно влияет на распознавание речи, и она была составлена из списка самых часто встречающихся конструкций языка Pascal. Однако еще более существенный вклад в качество распознавания речи вносит акустическая модель и по важности она занимает первое место. Важно обучить модель так, чтобы она лучше понимала, что говорит человек и как вне зависимости от окружения или говорящего. В данном случае велась разработка однодикторных акустических моделей в спокойном окружении, что значительно повышает шансы на верное распознавание речи.

Всего в тестовой программе было 144 слова, и аудиозапись длилась 123,67 секунд. Каждая из акустических моделей проходила ряд тестов с использованием двух языковых моделей. Первая – модель, построенная на словах, вторая – по часто встречающимся словосочетаниям.

Первый тест проводился с акустической моделью, обученной на 100 аудиофайлах. На этой маленькой модели ясно видно, что при малом количестве обучающего материала, распознавание будет сильно зависеть от качества языковой модели. При плохой языковой модели процент ошибки составил 48,96% [12], а при хорошей – 37,44%.

Второй тест проводился с акустической моделью, обученной на 293 аудиофайлах. На этой модели также видна зависимость качества распознавания от языковой модели. При плохой модели ошибка составила 12,96%, а при хорошей – 2,88%.

Третий тест проводился с акустической моделью, обученной на 500 аудиофайлах и на этом тесте уже не так заметна разница в качестве распознавания в зависимости от различных типов моделей. При плохой модели ошибка составила 7,2 %, а при хорошей – 2,87%. Это может сказать о том, что чем больше имеется обучающего материала, тем меньше распознавание зависит от качества языковой модели.

Все результаты проведенных тестов изображены на рисунках 7 – 10.

Одним из важных аспектов распознавания речи является оценка скорости распознавания речи и количество потребляемой оперативной памяти.

Все значения на диаграмме в секундах и по ней мы можем судить, что с увеличением количества обучающих аудиозаписей увеличивается и время распознавания речи. Любое из значений на диаграмме меньше, чем время, затраченное на произнесение речи, что является хорошим показателем.

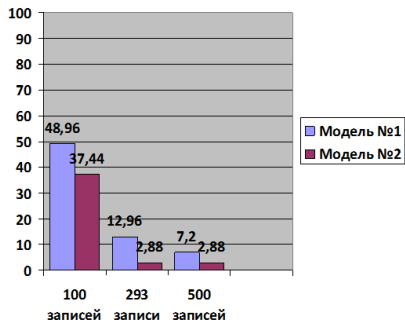


Рис. 7 – Процент возможной ошибки распознавания

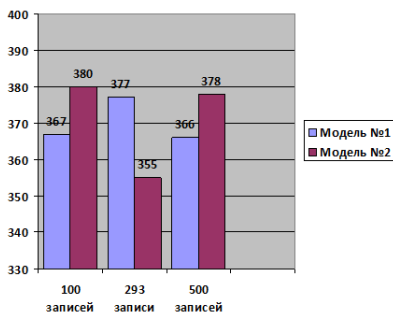


Рис. 8 – Потребляемая приложением память в мегабайтах при распознавании речи

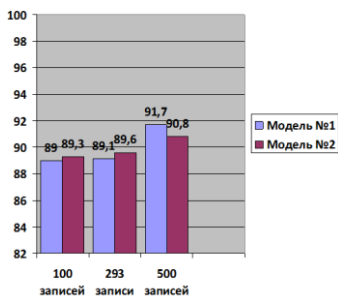


Рис. 9 – Время распознавания моделями аудиозаписи длительностью 123,67 секунды

Количество потребляемой памяти явно не зависит от качества моделей и от количества обучающего материала. Всего приложению было выделено 400мб оперативной памяти и во всех случаях количество потребляемой разрабатываемым программным продуктом памяти приближалось к этому порогу.

Выводы. Средствами системы Sphinx были разработаны языковые и акустические модели для словаря в 101 слово. В работе был выполнен анализ эффективности созданных моделей с помощью ряда тестов системы Sphinx на реальном примере программы. Тестирование показало, что данная инструментальная среда пригодна для создания приложений распознавания речи. Качество распознавания зависит от языковой модели и, главным образом, от акустической модели. Важно внимательно подбирать обучающие данные.

Данный программный продукт представляет собой однодикторное приложение для ввода и вывода текстов программ речью, а также позволяет редактировать их и отправлять на компиляцию. Приложение может послужить хорошим инструментом для облегчения взаимодействия пользователя с компьютером во время написания программ на языке Pascal. Количество потребляемой памяти явно не зависит от качества моделей и от количества обучающего материала. Всего приложению было выделено 400мб оперативной памяти и во всех случаях количество потребляемой разрабатываемым программным.

Список литературы

1. Федяев О.И., Бакаленко В.С., Савкова Д.Г. Речевой интерфейс для интеллектуализации ввода исходного кода программ: Материалы XV международной научной конференции им. Т. А. Таран «Интеллектуальный анализ информации». – Киев: КПИ, 2015. – с.250-256.
2. Федяев О.И., Бакаленко В.С. Intelligent interface for voice input of program source code: Материалы XI международной научно-технической конференции «Интерактивные системы: проблемы человеко-компьютерного взаимодействия». – Ульяновск: УГТУ, 2015. – с.120-126.
3. Бакаленко В.С., Федяев О.И. Разработка речевого распознавателя на основе моделей языка в среде CMU Sphinx: Материалы международной научно-технической конференции студентов, аспирантов и молодых ученых «Компьютерная и программная инженерия - 2015». – Донецк: ДонНТУ, 2015. – с.34-37.
4. Синтез и распознавание речи. Современные решения [Электронный ресурс]. – Режим доступа <http://www.frolov-lib.ru/books/hi/ch05.html>. – Суббота 5 сентября 2015 10:00:00.
5. Dynamic Programming Algorithms in Speech Recognition / Furtuna T.F. // Revista Informatica Economica. – 2008. - №2(46). – p. 94. – Режим доступа: <http://revistaie.ase.ro/content/46/s%20-%20furtuna.pdf>. – Воскресенье 6 сентября 2015 16:00:00.

6. CMU Sphinx Open Source Toolkit For Speech Recognition Evaluation [Electronic resource]. – Режим доступа: <http://cmusphinx.sourceforge.net>. – Четверг 18 июня 2015 12:30:00.

7. Free Pascal Open source Compiler for Pascal and Object Pascal [Электронный ресурс]. – Режим доступа: <http://www.freepascal.org>. – Среда 23 декабря 2015 11:30:00.

8. Скрытая марковская модель [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Скрытая_марковская_модель. – Среда 1 июля 2015 22:00:00.

9. Скрытые марковские модели [Электронный ресурс]. – Режим доступа: https://ru.wikibooks.org/wiki/Скрытые_марковские_модели. – Четверг 2 июля 2015 10:00:00.

10. N-грамм [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/N-грамм>. – Среда 9 сентября 2015 18:00:00.

11. Word Error Rate Calculation [Electronic resource]. – Режим доступа: <http://martin-thoma.com/word-error-rate-calculation>. – Среда 1 июля 2015 20:00:00.

12. Model Adaptation using MAP [Electronic resource]. – Режим доступа: http://www1.icsi.berkeley.edu/Speech/docs/HTKBook/node130_ct.html. – Пятница 16 октября 2015 18:00:00.

13. Training Acoustic Model for CMUSphinx. – Режим доступа: <http://cmusphinx.sourceforge.net/wiki/tutorialam>. – Среда 15 июля 2015 22:00:00.

Секция № 3
Системы
автоматизированного
проектирования



УДК 004.056

Разработка модуля аутентификации PDM-подсистемы в САД-компонентах САПР с открытым кодом

Калинина Е.С. (4 курс, каф. ПИ), Чернышова А.В., ст.преподаватель

Донецкий национальный технический университет

кафедра программной инженерии

lena00777@gmail.com, alla@pmi.dgtu.donetsk.ua

Аннотация

Калинина Е.С., Чернышова А.В., Разработка модуля аутентификации PDM- подсистемы в САД-компонентах САПР с открытым кодом. В тексте данной статьи представлен принцип работы модуля аутентификации, его назначение и возможности его внедрения в САПР системы с открытым кодом на языке Java. Описаны работы алгоритмов регистрации, аутентификации и оценки сложности паролей. Данный модуль позволяет разграничивать права доступа к информации и ограничить доступ к программе незарегистрированным пользователям, что повышает уровень безопасности в системах САПР.

Общая постановка проблемы. Современные методы обработки, передачи и накопления информации способствовали появлению угроз, связанных с возможностью потери, искажения и раскрытия данных, адресованных или принадлежащих конечным пользователям. Поэтому обеспечение информационной безопасности компьютерных систем и сетей является востребованным на сегодняшний день. Актуальность проблемы защиты информационных технологий в современных условиях определяется следующими основными факторами:

- количественным и качественным совершенствованием способов доступа пользователей к информационным ресурсам;
- многообразием видов угроз и возникновением новых возможных каналов несанкционированного доступа к информации;
- увеличением потерь (ущерба) от уничтожения, фальсификации, разглашения или незаконного тиражирования информации (возрастанием уязвимости различных затрагиваемых субъектов);
- концентрацией больших объемов информации различного назначения и принадлежности на электронных носителях [1].

В последнее время актуальным становится необходимость обеспечить эффективную защиту от несанкционированного доступа к проектным данным и документации, а также их нежелательной модификации. Эффективность защиты информации в автоматизированных системах проектирования достигается

применением средств защиты информации (СЗИ). Под средством защиты информации понимается техническое, программное средство или материал, предназначенные или используемые для защиты информации. В настоящее время на рынке представлено большое разнообразие средств защиты информации, которые условно можно разделить на несколько групп:

- средства, обеспечивающие разграничение доступа к информации в автоматизированных системах;
- средства, обеспечивающие защиту информации при передаче ее по каналам связи;
- средства, обеспечивающие защиту от утечки информации по различным физическим полям, возникающим при работе технических средств автоматизированных систем;
- средства, обеспечивающие защиту от воздействия программ-вирусов;
- материалы, обеспечивающие безопасность хранения, транспортировки носителей информации и защиту их от копирования [2].

Основное назначение средств защиты первой группы - разграничение доступа к локальным и сетевым информационным ресурсам автоматизированных систем. СЗИ этой группы обеспечивают:

- идентификацию и аутентификацию пользователей автоматизированных систем;
- разграничение доступа зарегистрированных пользователей к информационным ресурсам;
- регистрацию действий пользователей;
- контроль целостности СЗИ и информационных ресурсов [2].

Таким образом, учитывая актуальность проблемы защиты информации на сегодняшний день, имеет смысл внедрить модуль аутентификации в САПР-систему для последующей защиты разрабатываемых проектов и документов от несанкционированного доступа.

Средства разработки. Модуль будет реализовываться на языке программирования Java и в дальнейшем его можно будет внедрить в САПР-системы с открытым кодом, которые реализованы на языке программирования Java.

Традиционными способами аутентификации в системах САПР являются методы, основанные на знаниях (пароли) и/или обладании ключом (идентификационные карты и т.п.) т.е. парольная аутентификация, которая будет использоваться в ходе реализации программного модуля. Простота реализации и логическая ясность принципов функционирования делают системы парольной аутентификации самыми популярными. И основной задачей при разработке модуля является надёжная аутентификация пользователей.

Описание общего алгоритма работы. На первом этапе использования модуля пользователю сначала необходимо зарегистрироваться, введя логин и пароль в диалоговое окно. При успешной регистрации данные, введённые пользователем,

вносятся в базу данных. При последующем использовании пользователь должен будет войти в систему, т.е. пройти процесс аутентификации. Введённые им логин и пароль, который впоследствии хэшируется, будут сравниваться с теми, которые уже есть в базе данных. В случае совпадения данных пользователь входит в систему, иначе процесс аутентификации придётся повторить.

В настоящее время используются два основных метода аутентификации пользователей [3]: аутентификация по методу «запрос-ответ» и однонаправленная аутентификация. Конкретно для этого модуля будет использоваться аутентификация по методу «запрос-ответ».

Регистрация. Процесс регистрации происходит следующим образом: пользователь открывает форму и переходит на вкладку «Sign in». В появившихся полях он вводит логин и пароль и нажимает кнопку «Sign in». Введённые пользователем данные проверяются, ведётся оценка сложности пароля в процентах, которую можно увидеть, нажав на индикатор. Данные проверяются следующим образом: логин пользователя проверяется на валидность с помощью регулярного выражения (рис.1), т.е. логин должен содержать латинские буквы нижнего регистра, цифры или спецсимволы «-», количество символов от 3 до 15.

USERNAME_PATTERN = "^[a-z0-9_-]{3,15}\$"

Рис.1 – Регулярное выражение для проверки логина

Регистрация завершается успешно, если логин удовлетворяет требованиям и стойкость пароля «Strong» или «Very strong», в этом случае введённые пользователем данные вносятся в базу данных.



Рис.2 – Форма регистрации

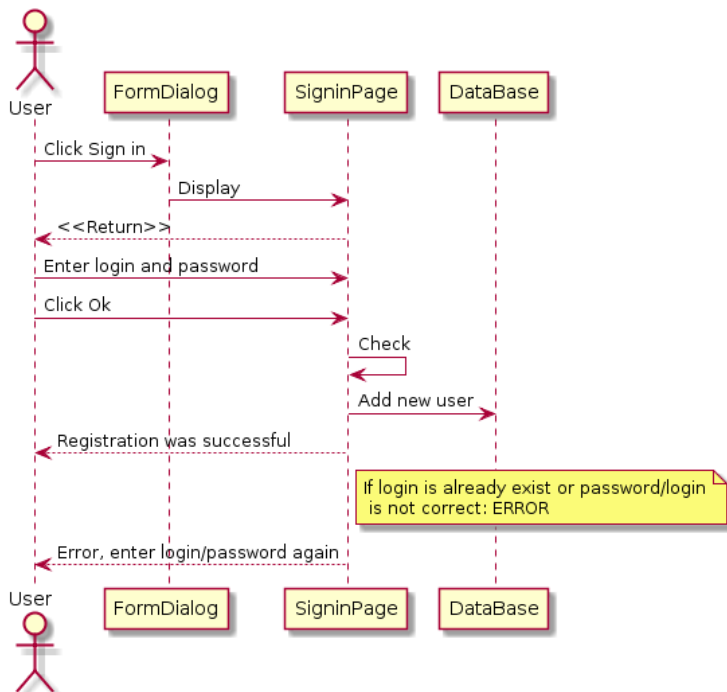


Рис. 3 – Диаграмма последовательности процесса регистрации

Аутентификация. В данном модуле аутентификация осуществляется по методу «запрос-ответ». Пользователь переходит на вкладку «Log in» и в появившихся полях осуществляет вход в систему введя пароль и логин. По нажатию кнопки начинается процесс аутентификации (рис.4). Введённый пользователем пароль хэшируется, на «сервере» генерируется некое псевдослучайное число, которое хэшируется по алгоритму MD5 [4] вместе с хэшем пароля (q). Логин введённый пользователем ищется в базе данных, в случае нахождения такого пользователя от пароля, зарегистрированного в базе, берётся хэш, а после чего он хэшируется по алгоритму MD5 вместе со сгенерированным числом (q'). В том случае, если « $q = q'$ » пользователь входит в систему, иначе повторяет процесс аутентификации заново введя снова логин и пароль.

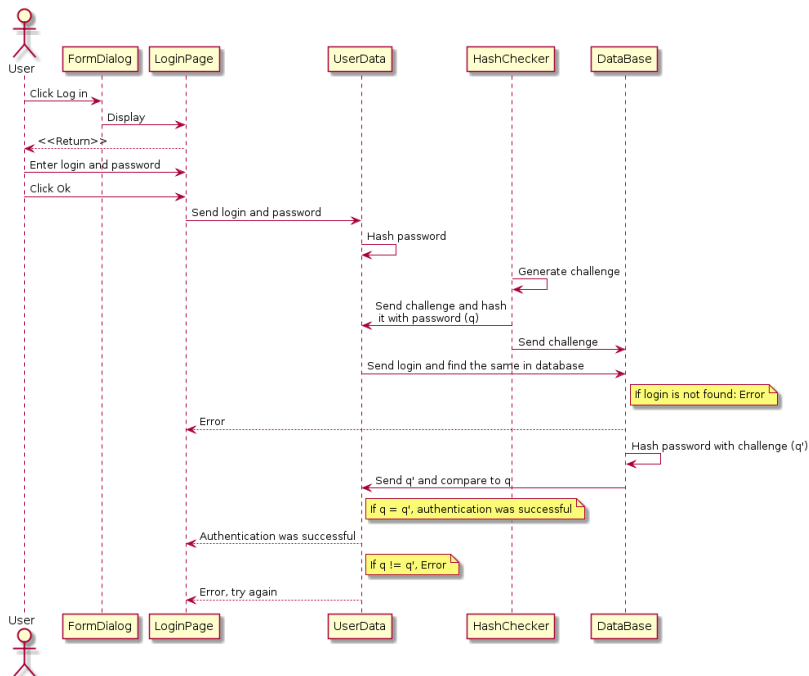


Рис. 4 – Диаграмма последовательности процесса аутентификации

Оценка сложности паролей. В настоящее время парольная аутентификация является наиболее распространенной, прежде всего, благодаря своему единственному достоинству – простоте использования. Однако, парольная аутентификация имеет множество недостатков. В отличие от случайно формируемых криптографических ключей (которые, например, могут содержать уникальный предмет, используемый для аутентификации), пароли пользователей бывает довольно просто подобрать из-за достаточно небрежного отношения большинства пользователей к формированию пароля. Часто встречаются случаи выбора пользователями легко предугадываемых паролей, например [5]:

- пароль эквивалентен идентификатору (имени) пользователя (или имени пользователя, записанному в обратном порядке, или легко формируется из имени пользователя и т.д.);
- паролем является слово или фраза какого-либо языка; такие пароли могут быть подобраны за ограниченное время путем «словарной атаки» - перебора всех слов согласно словарю, содержащему все слова и общеупотребительные фразы используемого языка [6];

– достаточно часто пользователи применяют короткие пароли, которые взламываются методом «грубой силы», т.е. простым перебором всех возможных вариантов;

– существуют и свободно доступны различные утилиты подбора паролей, в том числе, специализированные для конкретных широко распространенных программных средств;

– пароль может быть получен путем применения насилия к его владельцу;

– пароль может быть подсмотрен или перехвачен при вводе [7];

Для предотвращения подобного рода проблемы в модуле будет использоваться оценка сложности паролей по следующему алгоритму.

Стойкость пароля вычисляется по следующему алгоритму:

Изначально стойкости (strength) присваивается значение, равное длине пароля помноженного на 4. После этого происходят проверки на наличие в пароле символов верхнего/нижнего регистра, спецсимволов, цифр, исходя из результата к значению стойкости добавляются баллы или убираются. В том случае, если пароль состоит только из символов, его стойкость не будет достигать значения «Strong», т.е. чем больше значений алфавита задействовано, тем сильнее стойкость пароля. Стойкость (strength) может принимать значения от 0 до 100.

В зависимости от полученного значения паролю присваивается соответствующая характеристика его стойкости (Weak, Strong, Very strong). Результаты представлены в таблице 1.

Таблица 1- Классификация стойкости пароля в зависимости от значения

Значение	Стойкость
0-40	Weak
40-85	Strong
85-100	Very strong

Выводы. Таким образом, внедрив модуль парольной аутентификации пользователей с оценкой сложности пароля в систему САПР с открытым кодом, мы обеспечим разграничение прав доступа к различным уровням информации: группам справочников, отдельным справочникам, классам, объектам и методам расчетов, что позволит предотвратить доступ к среде нежелательных лиц и разрешит вход для легальных пользователей.

Список литературы

1. Актуальность проблемы обеспечения безопасности информационных технологий [электронный ресурс]. – Режим доступа: <http://asher.ru/security/book/itsi/01>
2. Средства защиты информации [электронный ресурс]. – Режим доступа: <http://www.univer.omsk.su/omsk/Edu/infpro/1/zashita/srzashinf.html>
3. Парольная аутентификация [электронный ресурс]. – Режим доступа: <http://www.volpi.ru/umkd/zki/index.php?man=1&page=23>
4. MD5 технологий [электронный ресурс]. – Режим доступа: <http://kriptografea.narod.ru/MD5.html>
5. Сложность пароля [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Сложность_пароля
6. Методы аутентификации [электронный ресурс]. – Режим доступа: <http://www.panasenko.ru/Articles/69/69.html>
7. Какой пароль защитит от взлома, или энтропия на службе секретности [электронный ресурс]. – Режим доступа: <http://samag.ru/uart/more/53>

УДК 622.272

Пространственная визуализация угольной шахты

Балагуров А.В., Скаженик В.Б.

Донецкий национальный технический университет, г. Донецк
 горный факультет, кафедра управления производством
anton.balagurov@gmail.com

Аннотация

Скаженик В.Б., Балагуров А.В. Пространственная визуализация угольной шахты. Рассмотрена проблематика малой интеграции специализированных компьютерных систем на угольных шахтах. Дана краткая характеристика специализированных компьютерных систем для моделирования угольных шахт. Выполнено поэтапное построение визуальной модели.

Ключевые слова: пространственная визуализация, специализированные компьютерные системы, электронные планы горных работ, программный комплекс «Шахта-3D».

Постановка проблемы. На угольных предприятиях Донбасса графическая документация представляется в плоской проекции, что связано в первую очередь с малой интеграцией специализированных компьютерных систем в область планирования горных работ. Такая форма представления информации затрудняет процесс планирования так как на графическом документе в основном

представлен только один из элементов производственного процесса угольного предприятия. При традиционном представлении графической документации инженеру, ответственному за разработку плана ведения горных работ, сложно представить с необходимой точностью пространственное расположение горных выработок и изменение взаимного расположения сети горных выработок во времени.

При традиционном подходе в процессе планирования могут возникать затруднительные ситуации, не позволяющие пересмотреть большое количество вариантов и произвести выбор оптимального решения. Ошибки при принятии решений наиболее вероятны при разработке нескольких сближенных пластов и сложной сети горных выработок.

Цель статьи – анализ возможных подходов к построению визуальной модели угольной шахты и реализация модели в программном комплексе «Шахта-3D».

Электронные планы горных работ. Создание электронных планов горных работ на основе пространственной геоинформационной системы открывает путь к автоматизации решения целого спектра геолого-маркшейдерских и горнотехнических задач [1]. В рамках единой системы возможно решение следующих задач: автоматизированный учет состояния балансовых и промышленных запасов угля в недрах; прогноз динамики напряженного состояния горного массива при ведении очистных работ; автоматизированный контроль проведения подготовительных выработок; прогноз сдвижений и деформаций земной поверхности; текущее и перспективное календарное планирование горных работ.

Специализированные компьютерные системы общего назначения. Программные комплексы хорошо зарекомендовавшими себя при моделировании месторождений, являются Gemcom, MineSched, MineFrame и др. Эти системы позволяют визуализировать горное предприятие и взаимосвязано решать геологические и горные задачи.

Gemcom Minex – программный комплекс, который наиболее широко используется на угольных месторождениях. Система позволяет моделировать пластовые месторождения различных углов падения пласта, а также геологические нарушения. Особенность данного пакета заключается в том, что в отличие от подобных программ Minex работает не с каркасными моделями, а с сеточными поверхностями, такой принцип работы упрощает построение пластов со сложной гипсометрией, а наличие специального функционала позволяет осуществлять их построение автоматически [2].

Программное обеспечение MineSched – это современный инструмент, позволяющий в динамике планировать календарный план открытых и подземных горных работ. MineSched дает возможность горному инженеру очень гибко выбирать время и место будущих работ. Результаты экспортируются в Microsoft Excel и Project для создания таблиц, графиков и презентаций [3].

Программное обеспечение MineFrame. Система комплексно автоматизирует процесс решения геологических, маркшейдерских и технологических задач по средствам компьютерного моделирования объектов горного предприятия. Данная программа применяется на предприятиях специализирующихся на отработке месторождений твердых полезных ископаемых открытым и подземным способом [4].

Этапы визуализации. Для визуализации угольного предприятия разработан программный комплекс «Шахта-3D», позволяющий на основе графической документации угольных шахт построить объемную модель [5].

Визуализация модели осуществляется в три основных этапа: визуализация геологии месторождения; визуализация существующей сети горных выработок; визуализация проектных выработок.

Для создания геологической модели месторождения используются данные разведки, которые представлены в базе данных цифровой информации. На основе этой информации визуализируется сеть разведочных скважин и результаты опробования. Также возможна визуализация геологических нарушений в виде поверхностей.

Построение модели земной поверхности и пластов угля заключается в оцифровке изолиний либо в указании точек поверхности с известными координатами. Посредством триангуляции сети точек формируется объемная модель поверхностей. Прежде всего, в модели фиксируются разрабатываемые, сближенные и балансовые пласты угля. Предусмотрена визуализация запасов с использованием различной цветовой гаммы с целью улучшения восприятия модели.

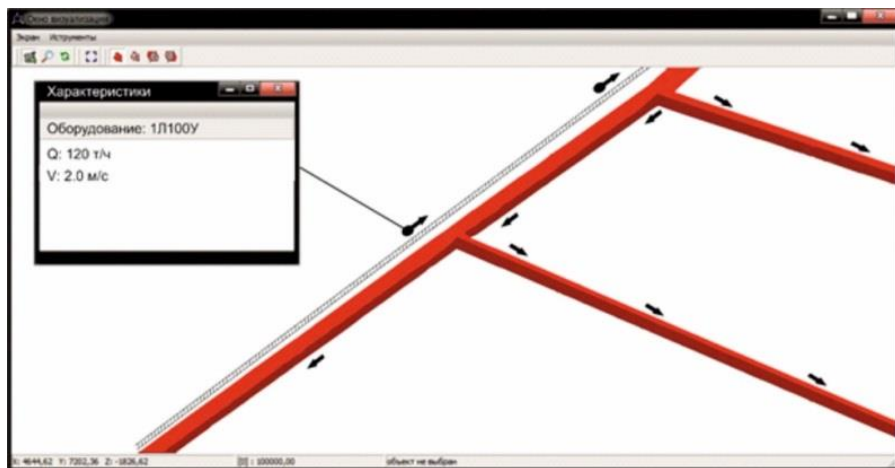


Рис.1 – Фрагмент объемной модели угольной шахты

Существующая сеть горных выработок строится в соответствии с планом горных выработок. Производится оцифровка бортов выработки по почве, а затем в программе визуализации по заданному сечению и форме выработки формируются объемные объекты. Выработки с исходящей струёй воздуха окрашиваются в синий цвет, со свежей - в красный. Вдоль выработки создаются объекты, являющиеся указателями оборудования, находящегося в выработке, ленточные, скребковые конвейера, рельсовый путь и т.д. (рис. 1). Данный способ позволяет объединить воедино план горных выработок, схему вентиляции и транспорта. Выработкам как объемным объектам присваиваются характеристики, такие как грузопоток, скорость и объем проходящего воздуха.

Построение проектных выработок осуществляется в программе визуализации. Так как проектные выработки могут быть пластовые и полевые процедура их построения внутри программы различна. В первую очередь это связано с тем, что траектория пластовых выработок должна соответствовать поверхности пласта и при сложной гипсометрии трасса выработки может иметь сложную форму. Для таких ситуаций реализован режим привязки выработки к поверхности. Для построения выработки требуется задать вектор направления, длину выработки и её сечение. Полевые выработки, как правило, невозможно однозначно привязать к существующим поверхностям, вследствие этого их привязка осуществляется к предварительно задаваемой плоскости.

В результате построения различных объектов формируется визуальная модель угольной шахты, а также дальнейшее развитие сети горных выработок (рис. 2).

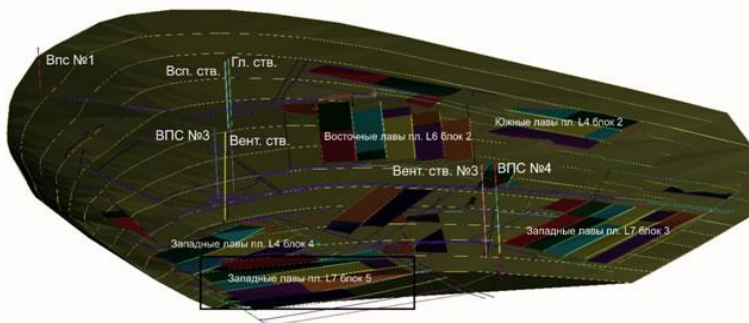


Рис. 2 – Объемная визуализация угольного предприятия

Выводы. Рассмотренная технология построения модели угольной шахты создает предпосылки для качественного изменения подходов к использованию горной графической документации и обеспечивает персонал горнодобывающих предприятий инструментарием для полноценного анализа возможных вариантов развития горных работ.

Список литературы

1. Грищенко, Н.Н. Геоинформационное обеспечение электронных планов горных работ / Н.Н. Грищенко // Наукові праці ДонДТУ. Серія гірничо-геологічна. Випуск 23. - Донецьк, 2001. - С. 133-140.
2. GemcomMinex Geology and Mine Planning for Stratified Deposits [Электронный ресурс]. – Режим доступа: www.gemcomsoftware.com.
3. Geovia MineSched. [Электронный ресурс]. – Режим доступа: www.minesched.com
4. Mineframe. [Электронный ресурс]. – Режим доступа: www.mineframe.ru
5. Скаженик, В.Б. Проектирование разработки месторождений на основе компьютерного моделирования / В.Б. Скаженик // Вісті Донецького гірничого інституту. - 2011. - № 1. - С. 8-14.

УДК 622.272

Системы автоматизированного проектирования (САПР) шахт

Емец А.И.

Донецкий Национальный технический университет

Кафедра программной инженерии

E-mail: a-emets2013@ya.ru

Аннотация

Емец А.И. Системы автоматизированного проектирования (САПР) шахт. Рассмотрена типичная система автоматизированного проектирования САМАРА – комплект программных средств автоматизации камеральных работ геолого-маркшейдерской службы. Технология предполагает создание, пополнение и использование для решения текущих задач геолого-маркшейдерской службы информационно-геометрической модели горных выработок предприятия. Система рассмотрена с точки зрения использования САПР в учебном процессе ДонНТУ.

Общая постановка проблемы. Для Донецкой Народной Республики наиболее актуальной задачей является развитие собственных производств, например, угольной промышленности, традиционно развитой в регионе. Донецкий технический университет (ДонНТУ) должен активно принимать участие в подготовке специалистов для угольной промышленности, а так же - разработчиков и модификаторов существующего программного обеспечения проектирования шахт.

С этой целью можно сделать анализ достоинств и недостатков существующих САПР шахт, как средств обеспечения учебного процесса по

специальности «системы автоматизированного проектирования» (САПР), подготовка которых осуществляется в ДонНТУ.

На сегодняшний день степень использования информационных технологий (ИТ) инженерными службами горнодобывающих предприятий значительно ниже, чем в других отраслях промышленности. Характер информации и технологии ее обработки *геолого-маркшейдерскими* службами предъявляют особые требования к программному обеспечению (ПО) и являются слишком емкими для какого-либо отдельно взятого пакета.

Дадим общую характеристику типичного САПР шахт САМАРА, придерживаясь источников [1-3].

Исследуем САПР САМАРА с точки зрения развитости компонент САПР, включая CAD (графический редактор), САЕ (система инженерных расчетов) и САМ (система автоматизации поддержки производства).

Исследования. САМАРА (аббревиатура от "Система Автоматизации Маркшейдерских Работ") - это комплекс программных средств, приемов и методик его применения, стандартов ведения модели горных работ, схем взаимодействия между предприятием и разработчиком на всех этапах внедрения и функционирования.

Технология предполагает создание, пополнение и использование для решения текущих задач геолого-маркшейдерской службы информационно-геометрической модели горных выработок предприятия. Модель состоит из графического файла (файлов) и внешней базы данных, содержащей дополнительную информацию о геометрии выработок, поверхности, геологических пластах и т.д.

Основой программного обеспечения технологии является универсальный инженерный графический комплекс AutoCAD. Учитывая широчайшее распространение комплекса и развитую его инфраструктуру, такой подход обеспечивает ряд преимуществ, повышающих шансы успешного освоения и закрепления технологии на предприятии. Среди них:

- обилие учебной и методической литературы и других материалов по AutoCAD, позволяющих самостоятельно изучить продукт и постоянно совершенствоваться в его применении;
- открытость AutoCAD, позволяющая дополнять его возможности собственными разработками;
- возможность применения многочисленных (более 5000) приложений и утилит, значительно расширяющих инструментарий;
- возможность создания единой рабочей среды с другими инженерными подразделениями предприятия и прямого обмена с ними графической и связанной информацией.

Базирование технологии на платформе AutoCAD отражает еще один принципиальный момент. Освоение ИТ инженером должно начинаться с

получения навыков инженерной компьютерной графики как таковой. То есть, человек для начала должен научиться делать на компьютере то, что он умеет делать на бумаге с помощью набора чертежных инструментов. И вот тут-то универсальные инженерные графические платформы легкого и среднего классов (на нашей территории это прежде всего AutoCAD, Microstation, реже - CADdy) – вне конкуренции.

Перечень основных функций системы САМАРА:

Расчет, уравнивание, формирование текстового документа и отрисовка:

- теодолитных ходов;
- результатов измерений методом тахеометрической и тригонометрической съемки;
- результатов измерений методом засечек;
- результатов измерений методом ординат;
- результатов нивелирных ходов;
- данных электронных приборов (Topcon, Trimble, ЗТА5, Leica, Nikon, Sokkia, Geodimeter).

Расчет профилей горных выработок с произвольным количеством параметров (почва и кровля выработки, головка рельса, ширина, высота выработки, высота прокладки кабеля, высота конвейера, ширина рабочего прохода и т.д. по желанию оператора).

Предрасчет сбоек, формирование текстового документа и отрисовка результатов.

Построения:

- автоматическое и полуавтоматическое построение плановых проекций горных выработок по опорным точкам с учетом скоб;
- автоматическое построение профилей выработок с произвольным количеством параметров;
- автоматическое построение пространственных (трехмерных) моделей горных выработок;
- автоматическое построение плановых проекций скважин по указанному пласту;
- автоматическое построение литографических колонок скважин в заданных диапазонах отметок и различных вариантах привязок;
- автоматическое построение пространственных моделей осей скважин.

Измерения:

- измерения горизонтальных длин по прямой и вдоль криволинейных траекторий;
- измерение горизонтальных и вертикальных углов;
- измерения наклонных длин и объемов участков горных выработок;
- разметка участков горных выработок на плоскости и в пространстве;
- измерение площадей в контурах произвольной формы.

Подготовка графических документов:

- библиотеки условных обозначений картографических документов, типов линий и штриховок (с возможностью собственного пополнения);
- управление порядком визуализации элементов картографических документов и их автоматическая сортировка;
- полуавтоматическое формирование ряда элементов картографических документов (овраги, строения, железные дороги ...);
- автоматическое формирование зарамочного оформления выкопировок заданного формата, масштаба, расположения и ориентации.

Информационная поддержка:

- функции связывания графических объектов модели с текстово-численной информацией и управления этой информацией;
- функции поиска и выборки моделей по связанным с ними данным.
- Программа содержит в себе обширный спектр возможностей для проектирования систем шахтного и горного делопроизводства, таких как:

Построение модели трубопровода шахтных вод

В данном случае САМАРА реально используется для работы экологических служб Запорожского железорудного комбината (г. Днепроудное).

Исходными данными для этой работы являлись:

- Эскиз съемки трассы трубопровода - несколько листов бумаги, известны координаты некоторых точек (из других источников);
- Проектный и фактический (после постройки) профиль трубопровода – также выполнен на бумаге;

Топографические карты масштаба 1:100000 (сканированные изображения) – для нанесения окружающей ситуации.

Сначала было произведено сканирование эскизов трассы трубопровода, калибровка и сшивка растровых изображений с учетом известных координат.

Затем была произведена векторизация эскиза съемки трассы (без привязки к координатам).

Аналогичная работа была произведена и с фрагментами топографических карт (калибровка шла по линиям сетки и по объектам на эскизе трассы).

После этого произвели наложение векторного плана трубопровода на растровую подложку (карту) с учетом характерных объектов (дороги, проезды, аварийные емкости и т.п.), которые были на карте и на эскизе трассы трубопровода. Получилась линия трубопровода, наложенная на карту местности (см. рис. 1 и 2).



Рис. 1 - Построение профиля штрека, карта местности

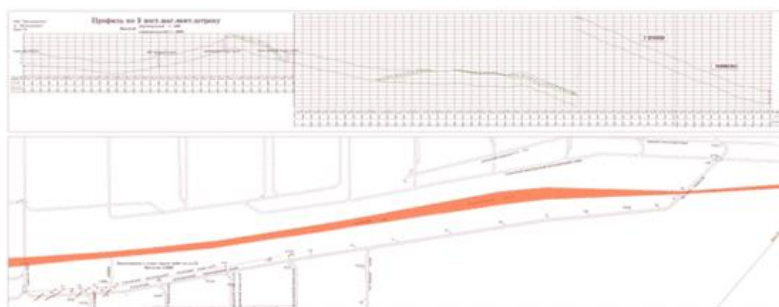


Рис. 2 - Построение профиля штрека, вариант изменения масштаба

Профиль построен автоматически с использованием команды «ПРОФИЛЬ» системы САМАРА с последующим редактированием средствами Автокада (для повышения наглядности профиль сделан без сдвижек по вертикали и нанесены некоторые дополнительные поясняющие надписи).

Эскиз – редактированная копия плана горных работ (убраны ненужные слои и объекты) – см. рис. 3.

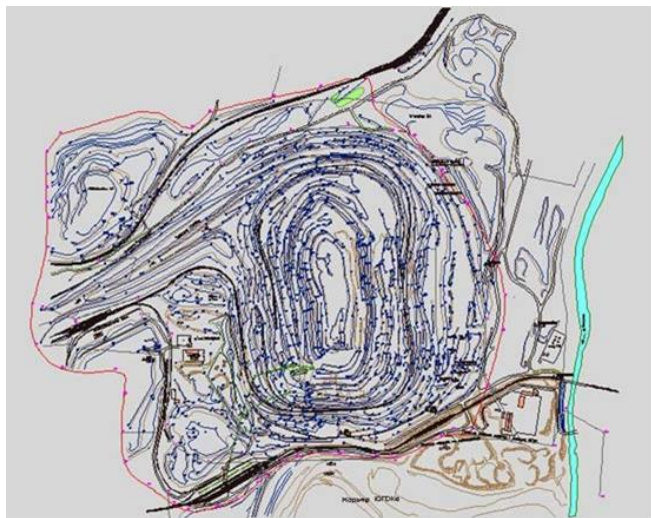


Рис. 3 – 3х-мерная модель карьера

Это рисунок оцифрованного плана карьера (оцифровка производилась с планшетов ведения горных работ). После этого с помощью системы FataMorgana (команда «Диспетчер пластов») построен фантом поверхности карьера. Для этого построения использовались данные по структурным линиям и по отметкам поверхности на планшете (в дальнейшем при пополнении используются данные инструментальной съемки) – см. рис. 4.

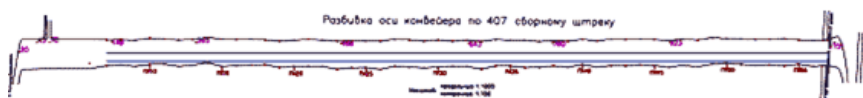


Рис.4 - Разбивка оси конвейера

Данный рисунок показывает пример использования системы САМАРА для разбивки оси конвейера. Борты штреха получены при помощи команды «ОРДИНАТЫ» системы САМАРА, при этом маркшейдерские точки и сопряжения были построены с помощью команд «ТЕОД.ХОД» и «ТАХЕО» системы САМАРА. После этого остается лишь прорисовать конвейер (в масштабе 1:1) и затем измерить привязки для выноса в натуру – см. рис. 5.

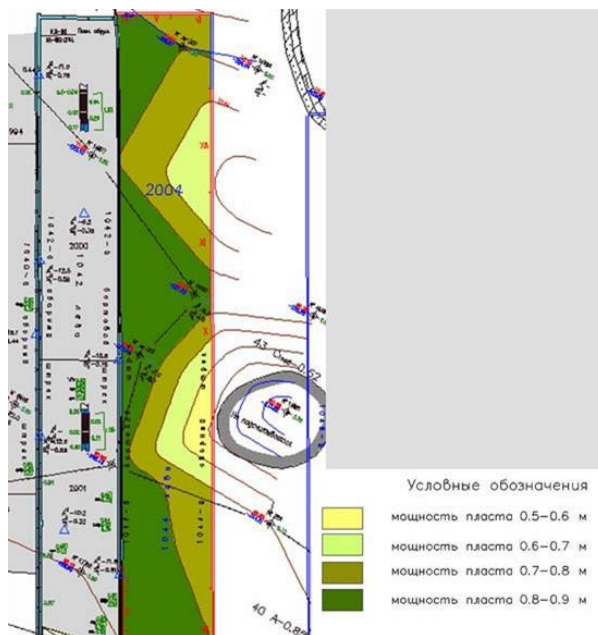


Рисунок 5 – Прорисовка конвейера

Эти рисунки иллюстрируют возможность наглядного представления условий работы. Здесь при планировании новой лавы были отстроены изогипсы мощности пласта по уже созданному фантому (а в данных фантома может быть записано любое количество параметров, не только мощность и отметки почвы, но и влажность, содержание серы, золы, метана, твердость и т.п.).

Выводы. На данный момент САПР САМАРА охватывает огромный комплекс возможностей для проектирования систем горнодобывающей промышленности и отличается от программ конкурентов обширным спектром решений.

Всего к настоящему времени система САМАРА содержит более 100 различных функций, обеспечивающих разработку и пополнение модели горных работ горнодобывающего предприятия, а так же использование модели для решения камеральных задач маркшейдерско-геологической службы.

Можно сказать, что САПР САМАРА имеет достаточно развитую компоненту CAD (графический редактор), САЕ (систему инженерных расчетов) и САМ (систему автоматизации поддержки производства).

Т.о., САПР САМАРА является достаточно развитым типичным представителем своего класса и может являться учебным пособием для разработчиков САПР с выполнением его модификации в рамках учебного процесса.

Список литературы

1. Конструкция САОГ. – Житомирский Г.И., 2008 г.
2. Каталог САПР. Программы и производители – Латышев П.Н., 2006г.
3. <http://ru.wikipedia.org/>

УДК 004.896

Проектирование технических объектов с помощью баз знаний в САД/САЕ системах на примере САТИА

Мёд В.П., Григорьев А.В.

Донецкий национальный технический университет
кафедра программной инженерии
E-mail: crazyleo853@gmail.com

Аннотация

Мёд В.П., Григорьев А.В. Проектирование технических объектов с помощью баз знаний в САД/САЕ системах на примере САТИА. Рассмотрен классический подход к проектированию технических объектов. Рассмотрены методы составления продукционных правил для представления знаний и их свойства. Рассмотрены модули, реализующие подход к проектированию с использованием баз знаний в САД/САЕ системе на примере САТИА. Рассмотрен способ создания экспертных правил средствами системы САТИА.

Общая постановка проблемы. Результатом любого проектирования является «проект — целостная совокупность моделей, свойств или характеристик, описанных в форме, пригодной для реализации системы» [1]. Таким образом, проектирование непосредственно связано с затратами времени и средств и для проектировщика главной целью является оптимизированный подход. Для этого в САПРах предусмотрено создание готовых шаблонов объектов при помощи средств параметризации, экспертных систем и баз знаний. Важным компонентом баз знаний являются продукционные правила, содержащие в себе информацию о параметрах и свойствах объекта, а также зависимостей между компонентами объекта. Задача данной работы определить эффективность такого подхода относительно существующего.

Исследования. В настоящее время существует такая система проектирования технических объектов, при которой на первом этапе разрабатывается концепция проекта. На данном этапе становится понятна сложность проекта, стоимость работ и затраты по времени. На следующем этапе разрабатывается спецификация и проектная документация. При таком подходе необходимо учесть все детали, иначе появятся ошибки, которые могут существенно повлиять на эффективность работы всей проектируемой системы, а

также на ее стоимость и время разработки [2]. Во избежание таких ошибок можно использовать подход, с использованием баз знаний и созданием продукционной модели, с помощью правил «если – то». Это самый популярный формат представления знаний. Существуют разные варианты интерпретации такого правила:

- 1) ЕСЛИ «условие А» - ТО «закключение В»;
- 2) ЕСЛИ «ситуация А» - ТО «действие В»;
- 3) ЕСЛИ «условие А и В выполнены» - ТО «условие С не выполняется» [3].

Такие правила называются продукциями и должны описывать небольшой фрагмент знаний. Также должна быть возможность добавления новых правил, которые не будут зависеть от уже существующих, а также возможность изменять старые [3].

Современные системы автоматизированного проектирования, предоставляют пользователю возможность автоматизировать некоторые этапы проектирования с использованием продукционных правил для преставления знаний об объекте.

Такая возможность реализована в CAD/CAE системе CATIA. Система содержит несколько модулей, с помощью которых возможна реализации базы знаний для объекта:

- Knowledge Advisor позволяет облекать знания конструктора изделия в удобную форму правила. Благодаря инструменту становится возможным придать объекту параметров, зависимых друг от друга посредством формул, правил или проверок. Таким образом, эксперт может проверить, что получится при изменении какого-либо параметра объекта. Так же можно проверить разные варианты исходных данных и что получится в результате их применения [4]. Так же есть возможность взять в качестве исходных брать как геометрические, так и негеометрические параметры такие как, например, материал, из которого изготовлен объект [5].

- Generative Knowledge позволяет создавать скрипты, включающие в себя элементы базы знаний. Интересен модуль тем, что кроме автоматической модернизации геометрических параметров объекта осуществляется также и изменение связанных с ним процессов, например, кинематики [4].

- Knowledge Expert более продвинутый модуль работы с классами объектов баз знаний для составления экспертных правил и проверок для них [4,6].

- С помощью стандартных CAD средств в системе CATIA создается прототип объекта, с последующей параметризацией с помощью приведенных выше модулей. Рассмотрим использование средства работы с базами знаний Knowledge Advisor на примере создания правила для объекта «болт» (рис. 1).

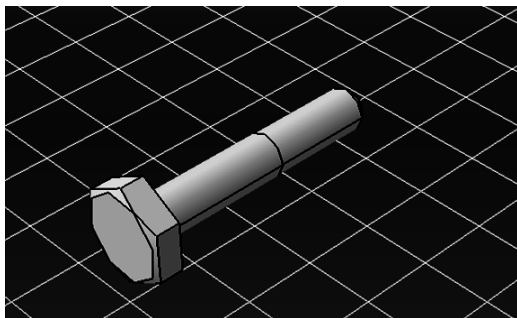


Рис. 1. Модель объекта для создания правила

При взаимодействии с модулем Knowledge Advisor открывается окно Rule Editor, включающий в себя встроенный редактор, словарь объектов и списки параметров объекта [5]. Здесь, с помощью встроенного синтаксиса, можно ввести зависимость параметров или описать производные правила. Введем следующее правило:

```
if `PartBody \ Призма.7 \ Первый предел \ Длина` >= 10mm  
    `PartBody \ Призма.6 \ Первый предел \ Длина` = 3mm
```

(1)

```
if `PartBody \ Призма.7 \ Первый предел \ Длина` <= 10mm  
    `PartBody \ Призма.6 \ Первый предел \ Длина` = 6mm
```

После введения данного правила появляется новая зависимость. Таким образом, при изменении параметров, используемых в правиле, модель объекта автоматически перестраивается (рис. 2).

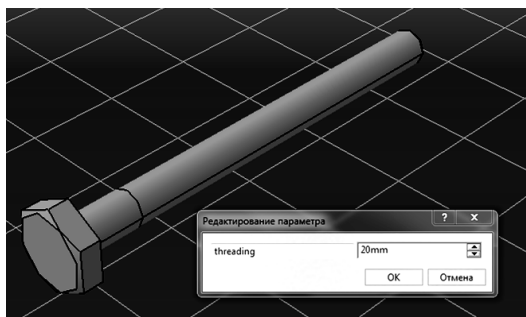


Рис. 2. Модель объекта после изменений параметров

Таким образом, можно реализовать взаимодействие всех параметров системы посредством экспертных правил и создать базу знаний, реализующую

возможность изменять исходные данные и получать новую многовариантную модель технического объекта.

Выводы. Суть представленного в статье подхода в том, что он дает возможность изменения проекта под требования заказчика благодаря созданию экспертных правил. В отличие от общепринятого способа, в нем реализуется возможность автоматизации процесса проектирования, что дает возможность не строить заново всю систему в случае изменения исходных параметров объектов, а также позволяет заранее проверить результаты тех или иных изменений в первоначальных характеристиках.

Список литературы

1. Проектирование [Электронный ресурс] / Интернет-ресурс. - Режим доступа: <https://ru.wikipedia.org/wiki/Проектирование>
2. Проектирование инженерных систем: этапы и стоимость работ [Электронный ресурс] / Интернет-ресурс. - Режим доступа: <http://www.kp.ru/guide/proektirovanie-inzhenernykh-sistem.html>
3. Продукционные правила для представления знаний [Электронный ресурс] / Интернет-ресурс. - Режим доступа: http://www.life-prog.ru/1_15464_produktsionnie-pravila-dlya-predstavleniya-znaniy.html
4. Knowledgeware: как задействовать самые ценные корпоративные активы [Электронный ресурс] / Интернет-ресурс. - Режим доступа: <http://www.sapr.ru/article.aspx?id=7125&iid=291>
5. САПР: оружие производителя XXI века [Электронный ресурс] / Интернет-ресурс. - Режим доступа: <http://www.sapr.ru/article.aspx?id=7101&iid=290#23>
6. Генеративное проектирование в САПР V5 [Электронный ресурс] / Интернет-ресурс. - Режим доступа: http://remmag.ru/admin/upload_data/remmag/08-5/DS.pdf

УДК 004.056

**Подсистема криптографической защиты файлов проектов для САПР
Клокова М.В., Чернышова А.В.**

Донецкий национальный технический университет
кафедра прикладной информатики
marizombie@outlook.com, alla@pmi.dgtu.donetsk.ua

Аннотация

Клокова М.В., Чернышова А.В. Подсистема криптографической защиты файлов проектов для САПР. В данной статье проведен анализ современных методов и способов защиты информации в системах автоматизированного проектирования. Рассмотрены криптографические средства, а также составлен список требований к разрабатываемой подсистеме защиты файлов проектов для поддержания высокого уровня безопасности информации в САПР.

Введение. Появление новых угроз информационной безопасности было вызвано широким внедрением информационных технологий, что не могло не повлиять на все аспекты человеческой жизнедеятельности. Информация, создаваемая, хранимая и обрабатываемая средствами вычислительной техники, стала определять действия большей части людей и технических систем, что привело к резкому увеличению возможностей нанесения ущерба, связанных с хищением информации, так как воздействовать на любую систему (социальную, биологическую или техническую) с целью ее уничтожения, снижения эффективности функционирования или воровства ее ресурсов (денег, товаров, оборудования) возможно только в том случае, когда известна информация о ее структуре и принципах функционирования. [1]

Различают несколько типов угроз для объектов защиты информации:

- угрозы конфиденциальности данных и программ;
- угрозы целостности данных, программ, аппаратуры;
- угрозы доступности данных.

Объекты защиты информации – это информация, носитель информации или информационный процесс, которые нуждаются в защите от несанкционированного доступа, изменения и копирования третьими лицами.

Носителем защищаемой информации может быть как физическое лицо, так и материальный объект или физическое поле. Информация содержится в виде символов, сигналов и образов, технических процессов и решений, количественных характеристик и физических величин.

Существует такое понятие как объект информатизации [2], который так же нуждается в защите. Защищаемый объект информатизации, как и

информационная система, предназначена для обработки защищаемой информации.

Основные объекты защиты информации:

- информационные ресурсы, содержащие конфиденциальную информацию;
- системы и средства, обрабатывающие конфиденциальную информацию (технические средства приема, обработки, хранения и передачи информации (ТСПИ));

- ТСПИ размещенные в помещениях обработки секретной и конфиденциальной информации. Общепринятая аббревиатура - ВТСС (вспомогательные технические средства и системы). К ВТСС относятся технические средства открытой телефонной связи, системы сигнализации, радиотрансляции и т.д., а также помещения, которые предназначены для обработки информации с ограниченного использования. [3]

Иными словами, объектами защиты информации являются источники информации, переносчики информации и получатели информации. Однако главная цель – это защита самой информации.

Постановка задачи. Анализировать существующие средства защиты в современных САПР [4] на примере САПР Autodesk AutoCAD [5], изучить криптографические алгоритмы шифрования и алгоритмы электронно-цифровой подписи, спроектировать подсистемы защиты файлов проектов для САПР, выбрать средства реализации, изучить возможности использования криптографических библиотек в Visual Studio 2012, разработать программную систему защиты файлов проектов для САПР с интуитивно понятным интерфейсом, внедрить подсистему защиты в САПР-систему.

Изложение материалов исследования. Встраиваемая подсистема криптографической защиты файлов проектов разрабатывается для САПР Autodesk AutoCAD 2012 и более новых версий. Выбор был сделан на основании возможности встраивания подпрограмм, написанных на C# с использованием .NET фреймворка.

САПР представляет собой комплекс технических средств, размещённый на нескольких автоматизированных рабочих местах, соединённых в локальную вычислительную сеть. На одном АРМ возможно совмещение нескольких подсистем, что уменьшает количество используемой вычислительной техники.

В САПР Autodesk AutoCAD с недавних пор есть возможность использовать программные модули для защиты файлов проектов. Однако в данной работе будет описана возможность создания собственной подсистемы защиты файлов проектов для САПР с использованием криптографических алгоритмов шифрования. Характерной особенностью является то, что подсистема защиты файлов проектов, написанная на языке C#, может быть внедрена в любую САПР, которая позволяет подключать свои программные модули.

Начиная с версии AutoCAD 2016 года, возможность добавлять пароли к чертежам исчезла. Вместо этого была представлена система, соответствующая

всем текущим стандартам защиты. Для симметричного шифрования эксперты рекомендуют 256-битный AES. Для асимметричного RSA рекомендуют 2048-битный ключ, как минимальную защиту, и версию 4096 бит для осуществления интернет-транзакций.

На данный момент в AutoCAD 2016 работают такие команды, связанные с параметрами безопасности:

- DIGITALSIGN – добавляет чертежу цифровую подпись, которая исчезает при неавторизованных изменениях;
- SAVEAS – в опциях диалога сохранения файла можно увидеть различные настройки DWG и DWF файлов, а также еще один вариант добавления цифровой подписи;
- SECURITYOPTIONS – контролируют ограничения безопасности для исполнимых файлов AutoCAD;
- SIGVALIDATE – отображает информацию о прикрепленной к чертежу цифровой подписи, в диалоговом окне – проверка подлинности цифровой подписи.

Кроме этого, в среде AutoCAD присутствует системная переменная SIGWARN, которая также отвечает за цифровую подпись. Если системная переменная включена и открыт файл с подлинной подписью, статус подписи будет отображаться. Если же она выключена и открыт файл, статус подписи будет отображен только в случае, если подпись неверна. Управление этой переменной осуществляется из диалогового окна сохранения файлов [6].

Разрабатываемая система обладает схожими возможностями: симметричное шифрование (AES) файлов проектов, асимметричное шифрование ключей AES [7] алгоритмом RSA [8] (для их защиты), использование однонаправленных функций (хеш [9] от файла проекта) и его (хеша) шифрование по RSA для ЭЦП [9] документа файла проекта. Однако сочетает всё это в простом интерфейсе, а также без надобности открывать проект чертежа для его шифрования. По большей части эта разработка уникальна тем, что в дальнейшем возможно встраивание данной подсистемы и в другие САПР, поддерживающие работу с .NET.

В качестве средства разработки была выбрана Visual Studio 2012.

В ходе разработки подсистемы криптографической защиты файлов проектов для решения задач шифрования использовалось пространство имен System.Security.Cryptography [10], которое предоставляет криптографические службы, включающие безопасное кодирование и декодирование данных, а также целый ряд других функций, таких как хеширование, генерация случайных чисел и т.д.. Данное пространство имен включает достаточное количество классов, которое способно предоставить нам все необходимые методы решения нашей задачи.

Например, одним из них является класс Aes – System.Security.Cryptography.Aes, вложенный в абстрактный класс симметрических алгоритмов System.Security.Cryptography.SymmetricAlgorithm [11]. В абстрактном

классе ассиметричных алгоритмов System.Security.Cryptography.AsymmetricAlgorithm расположен класс, используемый нами для цифровой подписи – System.Security.Cryptography.RSA [12].

Данная программа обеспечивает защиту данных пользователя следующим образом. После создания чертежа и сохранения его в каталоге на главной панели AutoCAD выбрать кнопку AutoSecure (название разработанной подсистемы). В главном окне с помощью кнопки Take from через диалоговое окно открытия файла подгрузить желаемый чертеж (см. рис. 1). После этого можно выбрать необходимый режим шифрования.

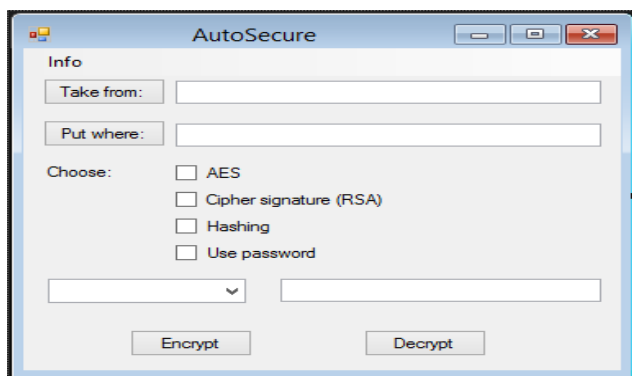


Рис. 1. Вид главного окна подпрограммы AutoSecure

Кроме того, для маломасштабных проектов на форме присутствует возможность введения ключа пользователем или автоматическая генерация ключа для минимальной защиты файла проекта. После выбора параметров пользователь нажимает кнопку Encrypt и шифруемый файл сохраняется в указанном каталоге (выбирается в поле Put where). Аналогичным образом можно дешифровать файлы проектов, только в конце нажимать кнопку Decrypt.

Одной из пока нереализованных задач является сохранение ключей в отдельных файлах. Тогда пользователь сможет хранить ключи к криптографическим алгоритмам не в директории, прописанной по умолчанию в подсистеме, а в любом каталоге диска.

Подсистема AutoSecure дает возможность использовать симметричное шифрование файлов проектов, ЭЦП для проверки целостности данных и защиты авторского права, использование простой парольной защиты для файлов проектов на основе хэш-функций.

Выводы. При проектировании подсистемы САПР криптографической защиты файлов были выполнены следующие задачи:

- изучены симметричные криптографические алгоритмы (AES);

- изучены асимметричные алгоритмы шифрования (RSA);
- изучен механизм использования электронно-цифровой подписи для контроля целостности и защиты авторства пользователя-создателя файлов проектов в САПР;
- разработано программное обеспечение для защиты файлов проектов САПР с использованием криптографических алгоритмов шифрования средствами Visual Studio 2012 с использованием криптографических библиотек;
- разработан простой, интуитивно понятный интерфейс программной системы.

Список литературы

1. Технология защиты информации: Информационные технологии, информация, информатика [Электронный ресурс]. – Режим доступа к ресурсу: http://technologies.su/tehnologiya_zachity_informacii.
2. Объект информатизации — WikiSec [Электронный ресурс]. – Режим доступа к ресурсу: www.wikisec.ru/index.php?title=Объект_информатизации.
3. Основные объекты защиты информации [Электронный ресурс]. – Режим доступа к ресурсу: ab-solut.net/ru/articles/osnovnye_obekty_zawity_informacii.
4. Система автоматизированного проектирования — Википедия [Электронный ресурс]. – Режим доступа к ресурсу: https://ru.wikipedia.org/wiki/Система_автоматизированного_проектирования
5. AutoCAD для Mac и Windows | САПР | Autodesk [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.autodesk.ru/products/autocad/overview>.
6. About Passwords and Encryption [Электронный ресурс]. – Режим доступа к ресурсу: <https://knowledge.autodesk.com/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/AutoCAD-Core/files/GUID-504C12C5-F17B-4B02-AC9B-5F7560261997-htm.html>.
7. Как устроен AES / Хабрахабр [Электронный ресурс]. – Режим доступа к ресурсу: <https://habrahabr.ru/post/112733>.
8. Алгоритм шифрования RSA [Электронный ресурс]. – Режим доступа к ресурсу: www.e-nigma.ru/stat/rsa.
9. Функции хеширования и электронно-цифровая подпись [Электронный ресурс]. – Режим доступа к ресурсу: studopedia.org/1-29126.html.
10. System.Security.Cryptography [Электронный ресурс]. – Режим доступа к ресурсу: [https://msdn.microsoft.com/en-us/library/system.security.cryptography\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.security.cryptography(v=vs.110).aspx).
11. Aes класс [Электронный ресурс]. – Режим доступа к ресурсу: [https://msdn.microsoft.com/ru-ru/library/system.security.cryptography.aes\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.security.cryptography.aes(v=vs.110).aspx).
12. System.Security.Cryptography [Электронный ресурс]. – Режим доступа к ресурсу: [https://msdn.microsoft.com/en-us/library/system.security.cryptography.rsa\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.security.cryptography.rsa(v=vs.110).aspx).

УДК 004.896

Разработка программных средств, повышающих эффективность решения задачи параметризации в САПР AUTOCAD

Роговец Е.А., Григорьев А.В.

Донецкий национальный технический университет г.Донецк

Кафедра прикладной математики и информатики

E-mail: pro100puls@mail.ru

Аннотация

Роговец Е.А., Григорьев А.В. Разработка программных средств, повышающих эффективность решения задачи параметризации в САПР AUTOCAD. Эта основана на методе параметризации и использовании математической модели. В работе предлагается решение задачи параметризации, использован вариативный метод создания параметрической модели.

Общая постановка проблемы. Параметрические операции — это ключевые методы построения геометрических фигур в процессе их моделирования. Основным элементом для большинства операций есть 3D-профиль, при движении которого осуществляется формообразование.

Изменение исходного профиля и создания связи между элементами модели является одним из главных способов создания параметрической модели. После создания операции пользователь может изменить форму и размеры первоначального контура, причем все эти изменения соответственно отразятся в модификации 3D-профиля и 3D-операции.

Такой подход реализован фактически во всех параметрических системах трехмерного моделирования. В свою очередь возможности проектировщика в модифицировании исходной модели благодаря мощному механизму геометрической параметризации в 2D-черчении при использовании ассоциативных построений с наложением взаимосвязей и объектных привязок не имеют предела.

Параметрическое моделирование существенно отличается от обычного двухмерного черчения или трехмерного моделирования. Конструктор, в случае параметрического проектирования, создает математическую модель объектов с параметрами, при изменении которых происходят изменения конфигурации детали, взаимные перемещения деталей в сборке и т.п.

Исследования. Все связи, именуемые параметрическими, можно условно разделить на категории:

- геометрические (параллельность, перпендикулярность, взаимоотношения и т.д.);
- топологические;

- математические (связь параметров разных элементов, блоков и уравнений);
- размеры.

Предлагаемая вариационная параметризация базируется на построении макета или эскиза, а также накладывании ограничений, зависимостей параметров и связей на объекты.[1]

Алгоритм построения параметрической модели вариационного вида можно свести к нескольким шагам:

- Создание эскиза для нужного действия, впоследствии на который и накладывают параметрические связи;
- Далее уточняются параметры, размеры и значения. Определяются переменные и задаются зависимости;
- Производится выбранное действие, в котором значения параметров и операций уже имеет значение и влияние;
- В случае необходимости создания сборки взаимное положение компонентов сборки задаётся путём указания сопряжений между ними.[2]

Вариационная параметризация позволяет легко изменять форму эскиза или величину параметров операций, что позволяет удобно модифицировать трёхмерную модель.

Предлагаемая вычислительная модель $M=(X=(X^1, X^i, X^0), R)$ состоит из ряда параметров:

X – множество параметров, значения которых могут принимать различный вид:

X^1 – определяет одно точное значение;

X^i – интервальное значение параметра;

X^0 – обозначает не определенное значение;

R множество отношений между параметрами. [3, 4]

В среде AutoCAD параметризация осуществляется с помощью функции автоналожения зависимостей. Она автоматически накладывает зависимости на контур изделия. Помимо принудительной параметризации существует также подразумеваемая зависимость, позволяющая накладывать параметризацию автоматически в процессе рисования. Так же важной особенностью является возможность параметризации и накладывания ограничений и зависимостей внутри динамического блока.

Применить параметризацию возможно с помощью обычных, стандартных инструментов которые находятся во вкладке «Параметризация» (см. рис. 1).

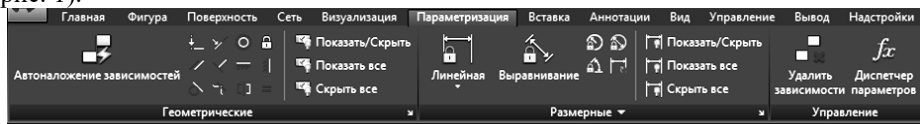


Рис. 1 – Меню параметризации

В блоке доступен диспетчер блоков, в котором перечислены параметры зависимостей и пользовательские параметры. В диспетчере можно указать порядок параметров и их значения.

Выводы. Представленная идея вариационной параметрической модели позволит создавать проекты, которые будут иметь взаимосвязанные параметры. Возможность параметризовать трехмерную модель может значительно упростить создание объектов и оптимизировать работу над существующими проектами. Математическая модель будет изменять параметры автоматически подстраиваясь под внесенное редактирование уже готового объекта. Такой подход предлагается реализовать на языке LISP и интегрировать в среду САПР AUTOCAD.

Список литературы

1. Параметрическое моделирование [Электронный ресурс]. - Режим доступа: https://ru.wikipedia.org/wiki/Параметрическое_моделирование
2. Параметризация [Электронный ресурс]. - Режим доступа: <http://www.sapr.ru/article.aspx?id=8064&iid=325>
3. Нариньяни А.С., Телерман В.В., Ушаков Д.М., Швецов И.Е. Программирование в ограничениях и недоопределенные модели // Информационные технологии №7, 1998. М., Издательство “Машиностроение”. - С. 13-22.
4. Григорьев А.В. Методы построения функций в специализированной оболочке для создания интеллектуальных САПР // Искусственный интеллект. – Донецк, 2001 – №3 – С. 40–53.

УДК 004.896

Графическая база данных в AutoCAD, назначение и структура Жусупов Е.В., Жусупов М.В., Григорьев А.В.

Аннотация

Жусупов Е.В., Жусупов М.В., Григорьев А.В. Графическая база данных в AutoCAD, назначение и структура. Рассмотрены методы создания графической базы данных в AutoCAD, средствами языка программирования AutoLisp. Определена структура, требование и назначения графических баз данных.

Общая постановка проблемы. В последние годы AutoCAD стал неотъемлемой частью для выполнения процесса проектирования. Существующие графические базы данных в AutoCAD имеют ряд недостатков, что делает

актуальной задачу создания новой графической базы данных, способной облегчить работу с чертежом.

Цель данной публикации – изложение общих требований к такой базе данных.

Исследования. Основу ИО (Информационное обеспечение) САПР составляет совокупность данных, которые необходимы для выполнения процесса проектирования. Совокупность данных, используемых всеми элементами САПР, называется информационным фондом. В этом фонде выделяют БД и архивы. Архивами пользуются редко и их помещают в долговременные ЗУ. Но в определенные моменты времени содержимое архивов помещают в БД. Информационное обеспечение представляет собой совокупность информационного фонда и средств его ведения. Основное назначение ИО состоит в создании, поддержке и организации доступа к данным. Ядром ИО является БД, которая в САПР играет роль инструмента, объединяющего отдельные элементы.

Цель предлагаемой работы – дать анализ графических баз данных, их назначения и структуры.

Базой данных называется структурированная совокупность связанных данных конкретной предметной области разнообразного назначения, в которой отражаются состояния объектов, их свойства и взаимоотношения.

Данные несут информацию об объектах, которые могут быть материальными (схема, плата) и нематериальными (методы оптимизации). Каждый объект характеризуется атрибутами. Среди элементов данных выделяют ключевые, по которым можно определить другие атрибуты объекта. Упорядоченную совокупность записей данных называют файлом данных или набором данных.

Требования к БД:

- целость данных;
- организация БД должна обеспечивать согласование времени выборки данных прикладными программами с частотами их использования прикладными программами САПР;
- универсальность;
- открытость БД;
- наличие языков высокого уровня взаимодействия пользователей с БД;
- секретность;
- оптимизация БД.

Базы данных. Виды моделей данных. База данных представляет собой дальнейшее развитие и интегрирование идей и методов информационно-поисковых систем.

Эта особенность организации базы данных обеспечивает использование одних и тех же данных во многих различных приложениях, а также позволяет

решать самые разнообразные задачи планирования, проектирования, управления и исследования. Как правило, в базах данных обеспечивается систематическое, централизованное создание, обновление и наращивание массива данных.

Важной особенностью БД является независимость данных от того или иного ИПЯ. Вместе с тем сохраняется возможность создания ИПЯ (информационно-планового языка) в такой форме, которая не влечет за собой необходимость изменять особенности хранения данных.

Для успешного функционирования САПР безусловно необходимо решение задачи интеграции графической информации в базе данных САПР. Как показало изучение состояния исследований в области БД САПР, актуальной научной задачей, имеющей существенное значение для САПР, является разработка инструментальных средств (как языковых, так и модельных), связанных с интегрированным представлением графической информации.

На основании проведенного анализа состояния вопроса в области реализации графических аспектов в БД САПР в настоящей работе поставлена цель - разработать языковые средства, методы и программные средства оптимального представления графической информации в базах данных САПР.

Для реализации этой цели необходимо:

- разработать языковые средства графического представления данных в рамках модели данных, адекватно удовлетворяющей требованиям САПР;
- разработать методы и алгоритмы оптимального представления графической информации в БД САПР;
- создать пакет прикладных программ (в рамках комплекса инструментальных средств, ориентированных на поддержку модели данных, адекватно удовлетворяющей требованиям САПР), реализующий разработанные языковые средства и алгоритмы оптимального представления графической информации.

Методика исследования основана на результатах теории баз данных, математической лингвистики, теоретико-графового подхода.

Сформулированы концепции разработки языковых средств графической интерпретации данных для объектно-ориентированных САПР:

- интеграция графической информации с информацией моделирования;
- независимость от устройств ввода/вывода;
- переносимость;
- простота описания;
- компактность при создании версий изображений.

Рассмотрим для примера организацию графической базы знаний в среде AutoCAD.

Рассмотрим общие сведения.

Чертеж Автокада с точки зрения объектной модели представляет собой базу данных. Как любая база данных, база данных Автокада содержит таблицы. Наиболее часто разработчик использует следующие таблицы:

- таблицу слоев.
- таблицу блоков.
- таблицу типов линий.
- таблицу текстовых слоев.
- словари.

Ключевое значение имеет таблица блоков.

Упрощенная схема базы данных Автокада показана на рис 1.

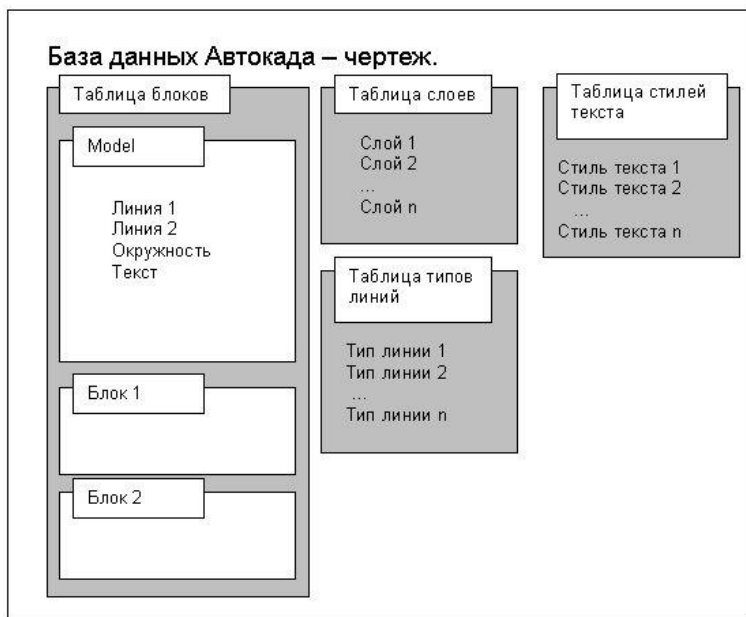


Рис. 1 - Упрощенная схема базы данных Автокада

На схеме показаны наиболее часто используемые таблицы базы данных Автокада. Вы видите, что примитивы, которые наблюдаются в области построения Model (в нашем примере – 2 линии, окружность и текст), лежат в таблице блоков, а точнее в ее записи, которая соответствует модели. На схеме не показаны взаимосвязи между элементами таблиц, так, например, целесообразно было бы провести стрелки от примитивов к записям таблицы слоев, однако материала по взаимосвязям элементов базы данных Автокада, требующих отдельного рассмотрения.

База данных Автокада характеризуется целым набором свойств, таких, как лимиты области построения, состояние режима объектной привязки, состояние тонирования и многих других.

Выводы. В предлагаемой работе рассмотрены основные принципы организации графических баз данных, определённые их функции и типичная структура. Рассмотрен пример организации графической базы данных в AutoCAD. Указаны, каким требованиям должна соответствовать любая графическая база данных, создаваемая под специфические условия применения.

Список литературы

1. В.В.Сидорков. «САПРы и подсистемы проектирования баз данных ». СПбГТУ, 2010 г. –203 с.
2. Камиссаров М.Ю. «Инструментальные средства интеграции и оптимизации представления графической информации в базах данных САПР». Москва 2011 г.

Секция № 4
Информационные
управляющие системы и
технологии



УДК 004.9

**Технологии электронного обучения: опыт МГУ имени М.В.Ломоносова
Главацкий С.Т., Михалев А.В.**

Московский государственный университет имени М.В.Ломоносова,
г. Москва, Россия,

Механико-математический факультет, кафедра теоретической информатики
E-mail: glavatsky_st@mail.ru

Аннотация

Главацкий С.Т., Михалев А.В. Технологии электронного обучения: опыт МГУ имени М.В.Ломоносова. Рассматриваются современные тенденции развития электронного обучения в высшей школе, в частности, в МГУ имени М.В.Ломоносова, а также принципы разработки открытого электронного ресурса.

Современная ситуация в России и в мире, тенденции развития. В настоящее время открытое и дистанционное (или, в современном понимании, электронное) обучение становится широко распространенным и даже обязательным элементом традиционных образовательных систем.

Развитие систем дополнительного образования в последние десятилетия связано с глубинными изменениями в образовательной системе, с ее ориентацией на образование в течение всей жизни человека. Если раньше человек мог получить среднее или высшее образование, которого бы ему хватило для успешной деятельности на всю жизнь, то теперь это все больше оказывается нереальным – нужно постоянно повышать свою квалификацию. Это обусловлено тем, что в современном постиндустриальном обществе информация и знания занимают ключевые позиции во всех сферах общественной жизни. Знание превращается в основную социальную ценность. Новая роль знаний в общественном развитии принципиально меняет место образования в жизни и всего общества, и каждого человека.

Электронное обучение (в англоязычной терминологии – eLearning) занимает в современной системе обучения особое место, органично дополняя другие формы обучения. Электронное обучение активно используется как в учебных заведениях, так и в других организациях. Ведущие мировые аналитические компании предсказывают ему большое будущее и утверждают, что мировой рынок систем дистанционного обучения является источником больших возможностей для инвесторов.

Открытое и электронное обучение предназначено для решения широчайшего круга задач:

- для реализации программ школьного или внешкольного обучения и в начальных, и в средних общеобразовательных учебных заведениях;

- для обучения педагогов в педагогических институтах и училищах, в целях повышения квалификации преподавательских кадров для дальнейшего непрерывного совершенствования их профессионального мастерства;
- как одна из форм профессионально-технического и непрерывного образования;
- для обучения в общественно-просветительской деятельности и других направлениях неформализованного образования;
- для высшего образования.

Немаловажным преимуществом дистанционного обучения является его большая эффективность. По оценкам специалистов, в процессе eLearning значительно сокращается время обучения, а скорость запоминания материала возрастает. Правда, данное преимущество, возможно, не всегда зависит от изучаемого материала и метода его подачи. Например, проблематично выработать правильное произношение, изучая иностранные языки только дистанционно, не имея достаточной разговорной практики.

Электронное обучение позволяет повысить качество образования за счет широкого использования мировых образовательных ресурсов и увеличения доли самостоятельного освоения материала, причем последнее особенно важно, поскольку постепенно обеспечивает выработку таких качеств, как самостоятельность, ответственность, организованность и умение реально оценивать свои силы и принимать взвешенные решения, без чего немислима успешная карьера. По данным Европейского центра развития профессионального образования Cedefop, 47% студентов считают, что мотивация и вовлеченность в процесс обучения — основные факторы успеха при получении электронного образования. Немаловажно также качество технического обеспечения и длительность обучения.

Дистанционное обучение, хотя отличается от традиционных форм обучения, но, по сути, представляет собой гибрид очной и заочной формы. В отличие от группового заочного обучения, оно является более индивидуальным, и так же, как при очных занятиях, предполагает наличие личного учебного плана и постоянное взаимодействие с тьютором, задача которого — курировать обучение, консультировать по сложным темам и вопросам, проверять контрольные работы и тесты, помогать готовиться к экзаменам.

Несмотря на то, что прогнозы относительно скорой электронной революции в системе образования всех стран пока не оправдываются, и традиционная форма обучения не уступит в ближайшем будущем свои позиции альтернативному — дистанционному — его варианту, электронное обучение уже перешло из статуса экспериментального в обычный режим, и активно используется как учебными заведениями, так и предприятиями, заинтересованными в постоянном повышении уровня знаний и навыков своих сотрудников.

Необходимо отметить, что проблемы развития eLearning стали рассматриваться гораздо шире, чем первоначально. Если сначала речь шла о дистанционном обучении как о возможной форме получения образования, то сейчас те или иные элементы электронного обучения достаточно широко применяются в вузах и в традиционных формах получения базового и дополнительного образования. С учетом интенсивного развития информационных технологий в мире меняется и сам образовательный рынок и образовательная среда, в рамках которой реализуется процесс обучения. Мировые тенденции все ярче показывают, что будущее за гибкими электронными моделями образовательного процесса, в котором активно используются различные средства, методы и технологии, в том числе и дистанционные. В настоящее время по уровню развития современных электронных средств обучения ряд вузов России (в том числе – экономических и гуманитарных) находится в одном ряду с ведущими университетами индустриально развитых стран Западной Европы и Америки.

Сложилась представления и о минимально необходимых требованиях к техническому и информационному оснащению дистанционного процесса обучения. Вошли в действие новые нормативные документы, в значительной мере легализующие методы дистанционного обучения.

Система дистанционного обучения в МГУ. В Московском университете создана Информационная среда дистанционного обучения (ИСДО) МГУ и система дистанционного обучения (СДО) МГУ (<http://sdo.msu.ru>).

Информационная среда дистанционного обучения (ИСДО) МГУ предназначена для обеспечения коммуникационной и информационной поддержки всех процессов дистанционного обучения МГУ. Основная цель системы – предоставить широкому кругу желающих доступ к методическим разработкам, учебному материалу и опыту преподавателей МГУ имени М.В.Ломоносова.

При создании ИСДО МГУ перед командой проектировщиков и разработчиков стоял ряд задач, решение которых необходимо для успешной реализации проекта:

- добиться того, чтобы требования, предъявляемые к рабочему месту студента, были минимальны (таким образом, система становится доступной для более широкого круга пользователей);
- максимально снизить требования к пропускной способности канала Интернет со стороны студента, что позволяет добиться максимально широкой географии использования системы (заметим, что определенные ограничения, конечно, присутствуют, но касаются, как правило, таких аспектов, как использование аудио- и видео-составляющих в учебных материалах, и, к сожалению, в этом случае не представляется возможным кардинально снизить нагрузку на канал, но разработчики надеются, что при современных темпах

развития коммуникационных каналов это ограничение не будет играть роли уже в ближайшем будущем);

- добиться максимальной отказоустойчивости и надежности системы, что является важным фактором для поддержания непрерывности учебного процесса;

- добиться широкой масштабируемости системы (данный аспект очень важен для дальнейшего развития, так как расширяемость системы только лишь за счет аппаратных ресурсов позволяет при увеличении аудитории в кратчайшие сроки увеличивать ее производительность, не снижая доступности сервиса для пользователей);

- обеспечить самую современную информационную защиту системы (этому аспекту нельзя уделять недостаточно внимания, в свете важности и конфиденциальности определенной информации, хранимой на сервере, а также резко растущего количества угроз информационной безопасности в сетях открытого доступа и, в частности, в сети Интернет).

- Разработанная СДО в течение 8-ми лет успешно поддерживает процесс обучения по нескольким десяткам дистанционных учебных курсов. Разработчиками системы с учетом анализа отечественного и зарубежного опыта электронного обучения проводится постоянная ее модификация и развитие в целях удовлетворения растущих потребностей пользователей.

Разработка электронного ресурса. В настоящее время в МГУ на базе нескольких узлов (как СДО, так и других) поддержки среды электронного обучения реализуется сотни различных дистанционных учебных курсов. Наиболее популярные из них – курсы подготовки поступающих в МГУ, а также ряд курсов повышения квалификации специалистов. В то же время на многих факультетах ряд очных курсов для студентов переведен на технологии eLearning: курсы не только обеспечиваются выкладываемыми на учебные сайты методическими материалами, но и поддерживаются широким спектром онлайн-сервисов, таких, как форумы, подсистемы тестирования и контрольных заданий, внутренней электронной почтой.

Следует учитывать и современные тенденции в выборе технологий обучения. Так, Тони Бейтс (Tony Bates), признанный эксперт в области методологии дистанционного обучения (ДО), предложил удачную модель, помогающую в принятии сложных решений при выборе технологий ДО. Он сформулировал семь критериев, помогающих выбрать или отвергнуть конкретную технологию, и соединил их в мнемоническом акрониме «ACTIONS». Критерии располагаются иерархически в порядке уменьшения степени важности:

- «Access (Доступность)» – это наиболее важный критерий. Если у обучающихся нет доступа к выбранной технологии, либо эта технология потребует от него дополнительных расходов, то она может стать неприемлемой;

- «Cost (стоимость)» – различные технологии требуют различных расходов как от обучающихся, так и от поставщика услуг ДО. Принимая решения

о выборе конкретных технологий, разработчики должны учитывать их экономическую эффективность;

- «Teaching and Learning (Преподавание и обучение)» – цели обучения могут быть достигнуты при правильном выборе соответствующих технологий. Но при должном подходе к разработке курсов обучающиеся могут успешно достигать целей обучения путем использования самых разнообразных технологий;

- «Interactivity (Интерактивность)» – наличие интерактивности и дружественного пользовательского интерфейса. Но следует учесть, что интерактивность – это в большей степени вопрос качественной разработки курсов, нежели неотъемлемое свойство конкретной технологии;

- «Organizational issues (Организационные вопросы)» – возможности (или сложности) эффективного внедрения избранной технологии в систему подачи материала;

- «Novelty (Новизна)» – внедрение новых технологий обеспечивает потенциальный выигрыш. Однако нужно быть уверенным в том, что технология останется экономически эффективной, когда ее «новизна» пройдет;

- «Speed (Скорость)» – скорость внедрения. Этот критерий связан с некоторым риском. Если ждать появления более совершенного технологического решения, то можно не успеть развить навыки и умения, требуемые в системе ДО.

В дальнейшем Бейтс предложил еще одну модель, обозначенную им акронимом «SECTIONS». Она может быть использована для анализа сравнительных достоинств того или иного вида мультимедийных средств.

- «Students (Студенты)»: подходит ли это вашим студентам?
- «Ease of Use (Простота в использовании)»: являются ли интерфейс и дизайн системы надежными и интуитивно понятными для ученика?

- «Cost Structure (Структура затрат)»: разумна ли стоимость?

- «Teaching & Learning (Преподавание и обучение)»: приведет ли это к повышению качества обучения?

- «Interactivity (Интерактивность)»: выводит ли система ДО учащегося за рамки простого чтения, просмотра и прослушивания?

- «Organization (Организация)»: является ли организационная система достаточно эффективной для гарантии того, чтобы предлагаемые конкретные средства общения или технологии были пригодными и практичными?

- «Novelty (Новизна)»: предлагается ли доброжелательная инструктивная помощь, которая в значительной мере опирается на текстовые материалы?

- «Speed (Скорость)»: возможно ли быстро ли реализовать и обновлять систему ДО?

Сам Бейтс при обсуждении предложенных критериев пояснял, что нужно просто взять их как «заголовки» для оценки технологий, но отмечал ограниченность такого подхода – «не должно быть иллюзий, что существуют

простые решения для выбора и использования технологий в очном или дистанционном образовании».

Выбирая те или иные решения в области организации eLearning в Московском университете, мы, прежде всего, ставим перед собой задачу создания открытого электронного ресурса, свободный доступ к которому могут получить все желающие, и именно их выбор позволит оценить правильность наших решений при создании учебного контента.

Предлагаемые в МГУ дистанционные учебные математические курсы разрабатываются с применением различных технологий, исходя из потребностей и возможностей обучающихся:

- либо это курсы, реализованные в рамках системы дистанционного обучения, где обучающимся и преподавателям предоставлен весь спектр традиционных возможностей для поддержки полноценного учебного процесса;
- либо это – электронная библиотека видео- и аудиозаписей лекций ведущих ученых МГУ (наподобие курсов, реализуемых в проекте “Coursera”).

Заключение. В заключение отметим, что технологии eLearning в современном мире развиваются очень бурными темпами.

Набирает популярность обучение с использованием мобильных устройств – mLearning, развиваются новые методы представления и передачи учебной информации.

Современными трендами развития систем управления обучением (LMS) являются:

- с одной стороны, отказ от использования LMS в качестве хранилища обучающих записей и средства доставки контента – в этой области на смену LMS приходит новый тип систем хранения данных – Learning Record Store (LRS), предполагающий, что контент может быть расположен повсюду и может включать в себя даже большие потоки обновляемых данных;
- с другой стороны, мы видим всё более тесную интеграцию LMS-систем с системами управления талантами – Talent Management System (TMS).

Мы живем в столь быстро меняющемся мире, что сложно представить себе, как будут выглядеть системы дистанционного обучения завтра. Тем не менее, основные принципы электронного обучения, о которых говорилось выше, остаются неизбылемыми. И его результаты внесут свою лепту в строительство здания образования и науки будущего.

Список литературы

1. Открытое и дистанционное обучение: тенденции, политика и стратегии. – М.: Изд. ИНТ, 2004. – 139 с.
2. Концепция развития математического образования в Российской Федерации: распоряжение Правительства Российской Федерации от 24 декабря

2013 г. № 2506-р г. Москва // Собрание законодательства Российской Федерации от 2014 г. , № 2 , ст. 148 (часть I).

3. Главацкий С. Компьютерные технологии в системе дистанционного обучения МГУ / Главацкий С.Т., Бурькин И.Г. – Saarbrücken: LAP LAMBERT Academic Publishing, 2013. – 156 с..

4. Использование современных информационных технологий для дистанционного обучения слушателей математическим дисциплинам / А.В. Михалёв, С.Т. Главацкий, Н.М. Адрианов, И.Г. Бурькин, А.А. Одинцов. // Современные проблемы математики, механики и их приложений. Материалы международной конференции, посвященной 70-летию ректора МГУ академика В.А.Садовниченко. — "Университетская книга" М, 2009. — С. 341–342.

5. Использование инновационных технологий электронного обучения для программы “МГУ-школе” / Главацкий С., Адрианов Н., Бурькин И., Иванов А., Одинцов А. // Международная конференция ИИТО-2012 “ИКТ в образовании: педагогика, образовательные ресурсы и обеспечение качества”, 13-14 ноября 2012 г., г. Москва, Россия. – М: Институт ЮНЕСКО по информационным технологиям в образовании, 2012. – С. 109-112.

6. Bates T. Technology, e-learning and distance education, 2nd ed. London; New York, Routledge, 2005 (<http://books.google.ru/books?id=SIsoTQV275QC>).

УДК 004.054

Метрики оценки качества web-систем

Давыденко Д.П., Губенко Н.Е.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
darinochka.davidenko@yandex.ru

Аннотация

Давыденко Д.П., Губенко Н.Е. Метрики оценки качества веб-систем.

Выполнен краткий анализ основных метрик оценки качества web-систем. Определены ключевые параметры метрик и методы оценки web-проектов по их параметрам.

Ключевые слова: метрика, web-проект, тестирование.

Постановка проблемы. Известно, что существует достаточно большое количество способов тестирования web-проектов[1]. Качество web-проекта определяется качеством самого продукта по степени соответствия требованиям пользователей и качеством процесса разработки. Одним из эффективных инструментов контроля качества является тестирование с помощью метрик.

Согласно международному стандарту ISO 14598 метрика - это количественный масштаб и метод, который может использоваться для измерения. Измерения помогают оценить как продукт, так и сам процесс его разработки. В результате измерений определяется количественная характеристика какого-либо свойства объекта.

Цель статьи - обзор и классификация метрик оценки качества *web*-систем. Кроме того, существует целый ряд классификаций, ранжирующих метрики по разным критериям. В виду сложности и трудоемкости процесса тестирования предполагается разработать специальное программное обеспечение для этих целей.

Метрики *web*-проектов – это количественные показатели, отражающие их отдельные характеристики. При правильном применении метрики являются полезным инструментом оценки работы *web*-системы. По ним можно отследить целый ряд важных параметров:

- снижения качества разработки;
- определить причины снижения качества;
- выявить самые сложные участки в системе и т.п.

Кроме того, числовые показатели способны помочь в планировании затрат ресурсов на дальнейшее развитие разрабатываемого продукта. Поэтому введение и использование метрик полезно для улучшения контроля над процессом разработки и тестирования *web*-систем. Необходимую для контроля информацию собирают (как вручную, так и автоматически) и используют для оценки прогресса выполнения запланированных работ и освоения бюджета.

Метрики количества - это те характеристики, которые можно измерить. Примеры таких метрик: количество дефектов, покрытие требований, количество дефектов, найденных пользователями, отношение количества открытых дефектов к закрытым и т.д.

Метрики качества отражают качество работы по проекту. Например, качество отчетов об ошибках, качество разработанных тест кейсов. Метрики качества называют еще субъективными оценками. Как правило, метрики качества должны подтверждаться метриками количества. Например, часто определяются оценка качества заносимых дефектов. Это субъективная оценка участников проекта. Для ее подтверждения необходимо проанализировать количество дефектов с резолюцией «Не является дефектом» или количество запросов программистов на уточнение шагов для воспроизведения инцидента. Субъективные метрики собирают в виде опросников в Wiki или Word, там же можно хранить обобщённые результаты опросов[2]. Для большей наглядности сгруппируем метрики по типам в таблицу 1.

Таблица 1 – Классификация метрик

Метрики по тест кейсам	
Passed/Failed Test Cases	Метрика показывает результаты прохождения тест кейсов - отношение количества удачно пройденных к завершившимся с ошибками. К концу проекта количество провальных тестов должно стремиться к нулю.
Not Run Test Cases	Метрика показывает количество тест кейсов, которые еще необходимо выполнить в данной фазе тестирования. Из полученной информации путем анализа можно выявить причины, по которым тесты не были проведены.
Метрики по багам	
Open/Closed Bugs	Метрика показывает отношение количества открытых багов к закрытым (исправленным и перепроверенным)
Reopened/Closed Bugs	Метрика показывает отношение количества переоткрытых багов к закрытым (исправленным и перепроверенным)
Rejected/Opened Bugs	Метрика показывает отношение количества отклоненных багов к открытым
Bugs by Severity	Количество багов по серьезности
Bugs by Priority	Количество багов по приоритету
Метрики по задачам	
Deployment tasks	Метрика показывает количество и результаты установок приложения. Если количество отклоненных командой тестирования версий будет критически высоким, рекомендуется срочно проанализировать и выявить причины, а также в кратчайшие сроки решить имеющуюся проблему.
Still Opened Tasks	Метрика показывает количество все еще открытых задач. К окончанию проекта все задачи должны быть закрыты. Задачи – это такие виды работ: написание документации (архитектура, требования,

	планы), введение новых модулей или изменение существующих по запросам на изменения, работы по настройке стендов, различные исследования и т.д.
Метрики по сложности	
Цикломатическая сложность	Метрика предназначена для оценивания сложности потока управления программы (control flow graph), вычисляется на основе ориентированного графа, где вычислительные операторы или выражения представляются в виде узлов, а передача управления между узлами – в виде дуг. Формула вычисления цикломатической сложности: $C = e - n + p$, где e – число ребер, n – число узлов, p – число соединенных компонент графа управляющей логики. Упрощенно формулу можно рассматривать как количество ветвлений, которые может проходить программа, увеличенное на единицу.
Метрики по назначению	
Прогнозирующие	Метрики нужны, чтобы спрогнозировать те проблемы, с которыми может столкнуться проект.
Диагностические	Метрики служат для мониторинга текущего состояния проекта.
Ретроспективные	Метрики помогают избежать в будущем тех ошибок, с которыми команда разработчиков сталкивалась в текущих или уже завершенных проектах.

В начале таблицы рассмотрены метрики по типам сущностей, участвующих в обеспечении качества и тестировании программного обеспечения, а именно: метрики по тест кейсам (Test Cases), метрики по багам / дефектам, метрики по задачам. Метрики "Open/Closed Bugs", "Bugs by Severity" и "Bugs by Priority" хорошо визуализируют степень приближения продукта к достижению критериев качества по багам. После каждой итерации тестирования количество открытых багов сравнивает их с реальными данными. Метрики "Reopened/Closed Bugs" и "Rejected/Opened Bugs" направлены на отслеживание работы отдельных

участников групп разработки и тестирования. Также интересна метрика по времени выполнения задач и многие другие.

Далее в таблице приведена еще одна распространенная группа метрик оценки сложности программных проектов. Эти метрики могут применяться и на ранних стадиях работы при осуществлении проектирования. Основное их предназначение – выявить наиболее критичные участки программного проекта, которые являются потенциальными источниками ошибок и повышенных рисков на всех стадиях его жизненного цикла. Вычисление метрик в ходе пошаговой реализации проекта позволяет своевременно определить наиболее сложные, сопровождающиеся высокими рисками, структурные единицы и принять меры по устранению рисков за счет внесения коррективов.

Более-менее универсальной классификацией метрик может считаться их разделение на: метрики оценки программного обеспечения и метрики оценки условий разработки программного обеспечения [3].

В каждой из этих категорий могут использоваться метрики с разными шкалами оценки свойств программного обеспечения: номинальной, порядковой, интервальной и относительной. Номинальная шкала показывает наличие либо отсутствие у программного обеспечения какого-либо свойства. Порядковой шкале соответствуют метрики, служащие для ранжирования определенных характеристик с помощью сравнения с некоторыми опорными значениями. Измерения по этой шкале по факту показывают взаиморасположение конкретных программ. Интервальная шкала показывает уже не только взаиморасположение программ, но и то, насколько далеко они отстоят друг от друга. Относительная шкала позволяет не только разместить программы определенным образом и оценить их размещение относительно друг друга, но также может показать, насколько далеко оценки отстоят от точки, начиная с которой характеристика вообще может быть измерена.

Таким образом, использование метрик позволяет эффективно управлять проектом: своевременно обнаруживать недостатки, устранять их и при этом видеть, насколько хорошо работают найденные методы решения проблем.

Выводы. Завершая обзор метрик оценки качества web-систем, можно утверждать, что в настоящий момент существует весьма большое количество отдельных показателей, с помощью которых можно эффективно тестировать разработку web-проекта. Использование целого комплекса необходимых метрик и графиков на всех этапах жизненного цикла проекта позволяет улучшить не только процесс тестирования, но и разработку проекта в целом. Исходя из проделанного анализа метрик качества web-систем, можно утверждать, что ни один из рассмотренных методов не является совершенным и для разработки программного обеспечения оценки качества web-систем целесообразно разработать интегрированное программное обеспечение, объединяющее большинство из рассмотренных методов.

Список литературы

1. Foundations of Software Testing: ISTQB Certification / D. Graham, E. van Veenendaal, I. Evans
2. et al. // CENGAGE Lrng Business Press. – 2006. – 258 p.
3. Зачем использовать метрики в тестировании [электронный ресурс]
URL:<http://bugscatcher.net/archives/3560>
4. Метрики в управлении проектами [электронный ресурс] // URL:<http://www.issoft.by/metriki-v-upravlenii-proektami/>

УДК 004.92

Политика защиты систем дистанционного обучения

Штанько А.А., Губенко Н.Е.

Донецкий национальный технический университет, г. Донецк

Кафедра «Компьютерное моделирование и дизайн»

E-mail: shtanko.andrei@mail.ru

Аннотация

Штанько А.А., Губенко Н.Е. Политика защиты систем дистанционного обучения. *Выполнен анализ сути политики защиты информации. Рассмотрены возможные угрозы информационной безопасности, выявлены места, которые больше всего подвержены вмешательству нарушителей, предложены возможные пути обеспечения безопасности*

Ключевые слова: *политика защиты информации, дистанционное обучение, информационная безопасность, защита систем.*

Постановка проблемы. В последнее время дистанционное обучение становится все более популярным, это связано с демократизацией процесса обучения, концепцией «пожизненного» обучения и следования парадигме «знания – к обучаемому». Все больше университетов записывают свои учебные материалы и лекции на видео либо представляют в иных компьютерных формах для поддержки эффективного удаленного учебного процесса. И в этом случае крайне важно защитить как личные данные студентов, так и сами материалы обучения, ведь среди них могут быть и дорогостоящие курсы и лицензированное программное обеспечение. Поэтому защита системы дистанционного обучения является сегодня сложной и важной задачей, требующей разработки особой политики ее реализации.

Цель статьи — проанализировать суть политики защиты информации, возможные угрозы информационной безопасности, рассмотреть что такое политика защиты, в чем ее суть, проанализировать возможные угрозы информационной безопасности, рассмотреть структуру системы дистанционного

обучения, выявить места, которые больше всего подвержены вмешательству нарушителей. Предложить возможные пути обеспечения безопасности.

Дистанционное обучение – это обучение с помощью средств телекоммуникаций, при котором субъекты обучения (ученики, педагоги, тьюторы и др.), имея пространственную или временную удаленность, осуществляют общий учебный процесс, направленный на создание ими внешних образовательных продуктов и соответствующих внутренних изменений (приращений) субъектов образования [1]. Информационно такие системы очень насыщены, а структурно достаточно сложны, имеют высокую стоимостную оценку, поэтому требуют особого внимания к вопросам безопасности.

Под информационной безопасностью понимают защищенность информации от незаконного ознакомления, преобразования и уничтожения, а также защищенность информационных ресурсов от воздействий, направленных на нарушение их работоспособности. Природа этих воздействий может быть самой разнообразной. Это и попытки проникновения злоумышленников, и ошибки персонала, и выход из строя аппаратных и программных средств, стихийные бедствия (землетрясение, ураган, пожар и т.п.) [2]. Под политикой безопасности понимают совокупность документированных управленческих решений, направленных на защиту информации и ассоциированных с ней ресурсов [3].

Сегодня принято выделять пять основных угроз информационной безопасности [2].

Существует пять основных типов угроз информационной безопасности:

- уровень доступа к информации в АС;
- угроза раскрытия параметров системы;
- угроза нарушения конфиденциальности;
- угроза нарушения целостности;
- угроза отказа служб (отказа доступа к информации).

Для разработки политики информационной безопасности необходимо провести анализ структуры системы дистанционного обучения (см. рис. 1). Которая поддерживает следующие функции: выдача необходимого для обучения материала, создания условий для удаленного тестирования и сдачи экзаменов, обеспечение способов обмена информацией между преподавателем и студентом, оценка отправленных студентом работ, обеспечение поддержки учащихся.

Из приведенной схемы на рисунке 2, что есть некие репозитории, которые содержат в себе материалы по обучению (видеофайлы, лекции, аудио уроки), а также сервисы (программы для работы, рекомендации, и т.д.) для удаленных студентов благодаря компьютерным сетям. По итогам своей работы студент может отправить результаты обучения преподавателю через интернет.

Для начала, студент должен зарегистрироваться, только после этого он имеет доступ ко всем материалам представленным в его курсе. Следующим шагом является само обучение, сюда входит получение необходимого материала,

его изучение и прохождения тестирования, для контроля качества образования. Обучаемый имеет право на какое-то время заморозить обучение, чтобы потом снова вернуться к нему. В некотором случае (к примеру если за отведенный период времени обучаемый не завершил один из этапов своего обучения, хотя бы с минимальной оценкой) обучение будет прекращено.



Рис. 1 – Структура системы дистанционного обучения.

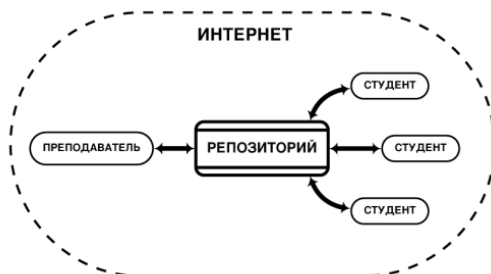


Рис. 2 – Репозитории сценария обучения.

Анализ приведенной структуры системы ДО позволяет сделать вывод, что для повышения безопасности наибольшее внимание нужно уделить следующим моментам

- удаленная аутентификация студента;
- контроль доступа;
- обнаружение вторжения;
- защита сетевых коммуникаций;
- гарантированность доставки;
- защита хранилища данных.

Аутентификация – процедура проверки подлинности заявленного пользователя, процесса или устройства. Эта проверка позволяет достоверно убедиться, что пользователь (процесс или устройство) является именно тем, кем себя объявляет. При проведении аутентификации проверяющая сторона убеждается в подлинности проверяемой стороны, при этом проверяемая сторона тоже активно участвует в процессе обмена информацией. Обычно пользователь подтверждает свою идентификацию, вводя в систему уникальную, неизвестную другим пользователям информацию о себе (например, пароль или сертификат) [2].

Но есть и более совершенные способы авторизации, они используют технологию смарт-карт, для разработки более сильных авторизационных систем с простым интерфейсом. Главное достоинство такого способа это безопасная обработка и хранение секретной информации – то есть конфиденциальные данные могут находиться только на смарт-карте. Единственная слабая сторона смарт-карт в сравнении с системами основанными на паролях – это увеличенная стоимость.

Целесообразно, чтобы аутентификационные механизмы являлись одинаковыми для всех частей образовательной системы. Это необходимо для простоты пользования системой.

После входа в систему, доступ к данным должен контролироваться посредством электронных сертификатов, эти сертификаты должны свидетельствовать о прохождении студентом обусловленного этапа обучения, а так же возможный переход к дальнейшему заданию. Эти методы должны быть использованы как дополнение к тем средствам контроля доступа, которые уже существуют.

В нашей системе ДО целесообразно выявление постороннего вмешательства с помощью мониторинговой системе в режиме онлайн, которая будет сравнивать действия авторизованного студента с историей его активности целью которого является определить достоверность пользователя. Эти записи состоят из целого ряда пунктов, среди них может быть время входа в систему, использование тех или иных сервисов, а так же данных, к которым доступ запрещен. Такой подход может быть полностью осуществлен программными средствами и не нуждаться в дополнительных затратах. Недостатком этого подхода является вероятность отказа входа в систему авторизованного пользователя, а так же возможное недовольство студента тем, что программа записывает о нем данные.

Сетевые коммуникации можно защитить с помощью кодирования данных. При выборе гибридной системы, для засекречивания данных будет использоваться криптография (конфиденциальные ключи для самой системы и для пользователя), так же есть асимметричная криптография, она необходима для упорядочивания сессионных ключей и для гарантии передачи (на основе электронно-цифровой подписи).

Требования для гарантированности доставки возникают как на стороне студента, так и на стороне системы и служат для предотвращения отказа от:

- отправки сообщения (подтверждение того, что студент – автор работы);
- приема сообщения (подтверждение того, что работа была принята системой);
- содержания сообщения (подтверждение того, что сообщение не было модифицировано в процессе доставки) [4].

Для достоверности передачи и приема сведений целесообразно использовать цифровую подпись, если коммуникации "заверены" отправителем при помощи ключа. Чтобы подтвердить верность содержания необходимо использовать код распознавания сообщения (message authentication code - MAC), который является итогом обработки сведений. Процедура обработки сведений сделана таким образом, что малейшая смена сути сообщения служит причиной изменения в MAC. Таким образом можно контролировать полноту сообщения.

Для решения проблем бесперебойности целесообразно: создание резервных копий и восстановление, защита организации, которые обеспечивают дистанционное обучение и так далее. Надежность, а так же доступность системы в любое время очень важна. Ведь у студента должна быть возможность подключиться к системе в любое удобное для него время, поэтому система должна работать безотказно всегда.

Выводы. Рассмотрена структура и способы организации дистанционного обучения, реализация возможных угроз информационной безопасности, а так же способы обеспечения защищенности такие как аутентификация, обеспечение авторизации при помощи смарт-карт, ознакомление со способами защиты при помощи мониторинговых систем, определение положительных и отрицательных аспектов, защита кодированием данных.

Список литературы

1. Андреев А.А. Определимся в понятиях // Высшее образование в России. 1998. – № 4. – С. 44–48.
2. Шаньгин В. Ф.: Ш12 Защита компьютерной информации. Эффективные методы и средства / Шаньгин В. Ф. - М. ДМК Пресс, 2010. - 544 с. ил.
3. Шаньгин В.Ф.: Информационная безопасность компьютерных систем и сетей: учеб, пособие. – М.: ИД "ФОРУМ": ИНФРА-М, 2008. - 416 с.: ил. (Профессиональное образование).
4. Титарев Д.Л., Егоров А.А. Защита информации в образовательных системах и средах. // Открытое Образование, 2003, № 5.
5. Маховиков С. А., Домнина Е. Г. Тенденции внедрения информационных систем в дистанционное обучение // Молодой ученый. — 2013. — №6. — С. 207-209.

УДК 004.65

**Применение современных web-технологий к автоматизации учета потребления энергоресурсов бюджетной сферы города Донецка
Цесаренко Д.С., Харитонов А.Ю.**

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
E-mail: danil.tse1265@gmail.com

Аннотация

Цесаренко Д.С., Харитонов А.Ю. Применение современных web-технологий к автоматизации учета потребления энергоресурсов бюджетной сферы города Донецка. Произведено сравнение между существующий ресурсом на базе MS Access и создаваемым web-сайтом, определены функциональные особенности сайта.

Ключевые слова: энергоресурсы, Access, PHP, MySQL, база данных, web-технологии.

Постановка проблемы. В настоящее время имеется огромная проблема энергосбережения. Человечество потребляет топливные ресурсы намного быстрее, чем их успевает вырабатывать природа. Возрастающие с каждым годом выработка и потребление энергии в мире создают необходимые условия для ускорения научно-технического прогресса. Но вместе с тем возрастающие объемы потребления энергии требуют все больших и больших объемов углеводородного сырья, запасы которого не безграничны. Главным фактором, обуславливающим необходимость энергосбережения, является истощаемость запасов органического топлива. По оценкам экспертов, при современном уровне добычи мировых запасов угля хватит на 600 – 1000 лет, нефти 150 – 250 лет и газа 120 – 300 лет. Как следствие, в будущем можно ожидать постоянного роста цен на нефть и газ. Отдельное внимание следует уделить энергосбережению зданий бюджетной сферы, а именно: школы, детские сады, больницы и прочее. Один из возможных путей решения проблем энергосбережения – это глобальный учет использования энергии. Однако это возможно только путем создания автоматизированных систем сбора, обработки и анализа информации об использовании энергии.

Цель статьи – сравнить уже существующий продукт на базе MS Access и последующий web-ресурс, выявить преимущества новой версии, определить функциональные особенности создаваемого ресурса.

Сравнение автоматизированных систем. Система, созданная на базе MS Access это своего рода база данных, заполняемая завхозами городских школ. Пользователи вводят данные счетчиков в базу и отправляют по электронной почте на указанный адрес. Письма автоматически считываются и данные вводятся в главную базу. После этих процедур появляется возможность составления графиков и отчетностей, на основе которых можно провести анализ потребления энергии.

Сейчас система требует глобального обновления и усовершенствования, т.к. она была создана в 2008 году, а за пройденное время технологии развились и вышли на новый уровень, что позволяет гораздо упростить администрирование и учет потребления энергоресурсов. Сравнение существующего продукта и разрабатываемого web-ресурса представлено в табл. 1.

Таблица 1 – Сравнение систем

Продукт на базе MS Access	Разрабатываемый web-ресурс
Для ввода данных пользователю необходимо наличие специального программного обеспечения.	Пользователю необходимо только подключения к сети интернет.
Привязка пользователя к определенному компьютеру, т.к. для ввода данных необходимо соответствующее ПО.	Пользователь может вводить данные в базу с любого компьютера в любой точке мира.
После того как база заполнена, ее необходимо отправить по электронной почте на сервер.	База заполняется автоматически после ввода данных пользователем.
Заполнение базы требует от пользователя базовых знаний в обращении с программой MS Access.	Интуитивно понятный интерфейс исключает возможность трудностей при вводе и отправке данных.

Функциональные особенности сайта. Разрабатывая систему учета потребления энергоресурсов предполагалось что школы, детские сады, больницы и т.д. используют одни и те же виды энергии в относительно одинаковом количестве. Исходя из этого анализ данных производился только в зданиях школы. В ходе работы мы увидели, что школы потребляют газ, холодную и горячую воду, электричество, а также в зданиях имеются по несколько подводов. Созданная система имеет интуитивно понятный интерфейс для пользователя, и процесс ввода данных очень прост. При регистрации субъект выбирает из предлагаемого списка здания, для которых в дальнейшем будет заполнять базу данных. Далее, после авторизации, пользователю предлагается ввести показания счетчиков за каждый календарный день. После создания базы данных, программа составляет графики ресурсопотребления что позволит выявить отклонения и позволит проанализировать их причины.

Сам сайт выполнен в стиле минимализма и не загружает компьютер пользователя, что сказывается на быстродействии. Среда, с которой непосредственно взаимодействует пользователь написана на современных HTML5 и CSS3. База данных создана на MySQL 5.1, а функционал сайта обеспечил язык программирования PHP версии 5.3. Все данные вводимые пользователем, хранятся в базе данных на сервере кафедры КМД.

Доступ к базе данных осуществляется с помощью таблиц «Энергопотребляющие организации», «Информация по организации», «Информация о здании», «Потребление», доступ к котрым осуществляется иерархически, от верхнего уровня к нижнему. Принцип иерархии показан на рис. 2

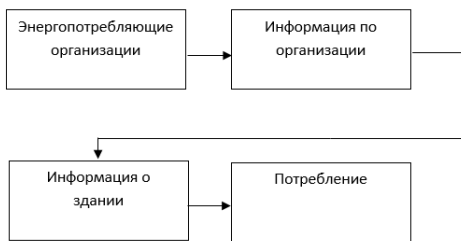


Рис. 1 – Принцип иерархии доступа

Структура данных в MySQL выглядит аналогично с Access, и изображена на рисунке 2.

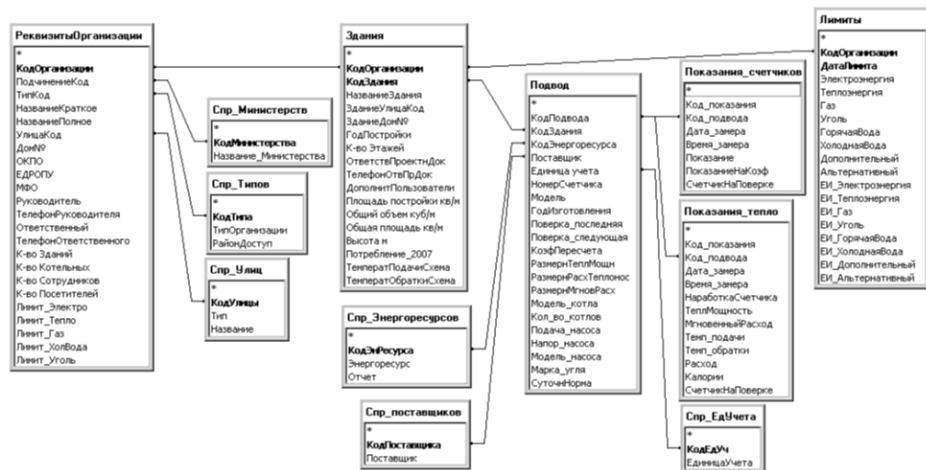


Рис. 2 – Структура данных

Выводы. Имеющийся ресурс позволяет анализировать потребление энергии более рационально. Это отражается, в первую очередь, на экономической составляющей систем энергоснабжения и делает бюджетную сферу города более экономичной. Но также несет в себе и экологический характер. Оказывает позитивное влияние одновременно на решение экологических проблем по сокращению выбросов парниковых газов, снижению вероятности и интенсивности кислотных дождей, минимизации угрозы здоровью людей.

Данная система зарекомендовала себя как действующее решение проблемы учета и анализа энергоресурсов. Однако, в настоящее время она устарела и требует глобального обновления с применением современных технологий, что упростит выполнения задачи ее конечных пользователей. Новый web-сайт позволит выполнять все те же операции в наименьшее количество действий, не будет требовать знаний специальных программ от пользователя и исключает привязку к устройству с которого осуществляется ввод данных

Список литературы

1. Практическое руководство по повышению энергоэффективности муниципальных систем / Под ред. А.С. Копеца и Р.В. Кишканя. – Донецк, 2007. – 204 с.
2. Отчет о научно-исследовательской работе «Автоматизация учета потребления энергоресурсов бюджетной сферы г.Донецка» / Под рук. С.Сафьяц - Донецк, 2008. – 163с.
3. ГОСТ 34.601-90. ИТ. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

УДК 004.42

Архитектура распределённой системы управления целеориентированными проектами

Двойкин А.А., Привалов М.В.

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
94dvoikin@gmail.com

Аннотация

Двойкин А.А., Привалов М.В. Архитектура распределённой системы управления целеориентированными проектами. Выполнен анализ и описаны особенности целеориентированных систем, их актуальность, описаны функции систем управления целеориентированными проектами, необходимость использования распределённой архитектуры.

Ключевые слова: целеориентированные системы, системы управления проектами, распределённая архитектура, микросервисы.

Цель статьи – провести анализ и описать особенности целеориентированных систем, описать их актуальность и преимущества, описать функции систем управления целеориентированными проектами, описать необходимость использования распределённой архитектуры.

Введение. С 90-х годов по настоящее время используют процесс-ориентированный подход в управлении предприятиями, который основывается на понятии процесса. Процесс в данном случае – это набор функций, выполнение которых в определенной последовательности должно обеспечить результат. Однако процессы имеют проблемы с масштабированием (тяжело вставить дополнительную функцию в процесс, не нарушив при этом процесса), гибкостью (при необходимости если одну функцию вытащить из одного процесса и вставить в другой, то это может привести к нарушению обоих процессов или вовсе невозможно т.к. все процессы выполняются по определенным технологиям), прозрачностью (в процессах мы можем видеть только последовательность функций их выполнения и никаких других зависимостей функций друг от друга). Целеориентированный подход может решить данные проблемы.

Целеориентирование означает, что все процессы на предприятии ориентируются на достижение целей, т.е. достижение желаемых результатов, поставленных руководством. Цели предприятия делятся на подцели и так до тех пор, пока подцель на определенном уровне декомпозиции не станет реальной для выполнения сотрудниками предприятия. Таким образом, строится иерархия целей предприятия, где четко видна зависимость целей, что решает проблему с прозрачностью. При необходимости любую цель можно дополнительно разбить

на подцели либо для определенной подцели назначить другую родительскую цель, что решает проблемы с гибкостью и масштабированием.

Целеориентированный проект, как и любой другой, имеет следующие признаки:

- Цель – получить конечный продукт или результат;
- Установленные сроки начала и завершения – дата начала проекта и дата его окончания;
- Определенные ресурсы – трудовые, финансовые, оборудование и информация.

Управление проектом – это процесс руководства всеми работами по проекту от начала до завершения. Управление проектом предполагает три основных вида деятельности: планирование, организация, управление.

В процессе управления целеориентированными проектами менеджеры часто сталкиваются с трудностями в:

- представлении информации о списке задач и целей, об ответственности сотрудников, о сроках выполнения задач и целей
- составлении иерархии целей, идентификации крупных вех проекта и их декомпозиции;
- презентации графика работ.

Для того, чтобы упростить работу менеджеров, необходимо автоматизировать процессы, в которых возникают трудности. А именно, разработать инструменты для:

- создания и редактирования информации о задачах и целях проекта;
- назначения ответственных за выполнение задач и целей;
- просмотра иерархии целей и перераспределения ответственности в случаях возникновения проблем.

Таким образом, разработка системы управления целеориентированными проектами является актуальной задачей. Построенная система должна, помимо удовлетворения вышеописанных потребностей менеджеров, быть гибкой, доступной, масштабируемой и хорошо поддерживаемой. Такие особенности систем, как правило, отражаются в их архитектуре. Определим архитектуру нашей системы. Для этого выполним анализ существующих систем управления проектами.

Анализ архитектуры существующих систем. Самый простой в реализации вариант системы управления проектами – это локальные системы управления. Локальная система – это система, которая находится непосредственно на рабочем месте и не имеет серверной части. Такая система не имеет связи с другими системами, пока входная информация не будет передана в неё вручную или импортирована с какого-либо носителя. Рассмотрев применимость такой архитектуры к системе управления целеориентированными проектами, можно сделать вывод, что для каждой такой системы нужно отдельно

формировать входные данные: задачи, цели, что является очень неудобным, особенно в случае р2р-взаимодействия.

Клиент-сервер — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами[3]. Физически клиент и сервер — это программное обеспечение. Обычно они взаимодействуют через компьютерную сеть посредством сетевых протоколов и находятся на разных вычислительных машинах, но могут выполняться также и на одной машине. Программы — сервера, ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных. Данный тип архитектуры подходит для целеориентированных систем управления проектами, как таковых, но имеет не самую лучшую в данном случае гибкость и масштабируемость. Данные системы подразумеваются постоянно расширять или изменять их структуру в зависимости от сферы деятельности предприятия, для которого осуществляется управление проектами, а в клиент-серверное приложение с нераспределенной архитектурой тяжело вносить изменения при динамичной трансформации структуры предприятия.

Многозвенная архитектура является аналогом клиент-сервера, но с возможностью распределения серверного приложения на несколько разных машин[4]. То есть многозвенная архитектура обладает теми же недостатками по отношению к системам управления целеориентированными проектами.

Сделав анализ существующих архитектур, возможно сделать вывод, что для разработки систем управления целеориентированными проектами необходимо использовать распределенную архитектуру, способную обеспечить как веб-доступ к системе, так и любой другой. Распределенные архитектуры, основанные на веб-службах или микросервисах, обладают этими свойствами, при этом обеспечивая лёгкое расширение и простое управление.

Разработка архитектуры. Целеориентированные системы управления имеют следующие особенности:

- Целостность. Системный взгляд на организацию и окружение. Целеориентированная система использует целостный / системный подход для рассмотрения организации;

- Принятие нечеткого и иррационального поведения людей. По определению целеориентированная система должна покрывать человеческую составляющую организации;

- Широкий взгляд на процессы управления. Целеориентированная система не рассматривает “управление” только как работу, выполняемую исключительно людьми с титулом “менеджер” (“начальник”). Вместо этого, под управлением понимается любой вид целенаправленной деятельности [1];

- Широкий взгляд на активные элементы. В целеориентированной системе активным элементом (элементом принимающим решения), который

выполняет функции управления, может быть человек, группа людей, система ИИ, робот или примитивный микроконтроллер;

– Широкий взгляд на роли элементов. Много концепций, используемых в автоматизации в настоящее время, используют довольно жесткий подход. Они определяют объекты управления и управляющие объекты. И со временем это разделение не меняется. Элементы в целеориентированной системе способны принимать различные роли в процессах управления в различное время, в зависимости от контекста.

– Одним из основных принципов целеориентированных систем является принцип непрерывного управления, основанный на концепции НОРД циклов[1]. НОРД цикл состоит из 4 определенных состояний:

- Наблюдение: сбор исходных данных.
- Ориентация: анализ и синтез данных для формирования текущей модели.
- Решение: определение последовательности действий на основе текущей модели.
- Действие: исполнение принятых решений.

Следовательно, возможно сделать вывод, что является необходимым разработать распределённую систему управления, основной сферой применения которой являются циклические процессы, также при анализе архитектур существующих систем был сделан вывод о необходимости применения распределённой архитектуры.

Опираясь на вышесказанное, становится возможным при разработке серверной части применить микросервисную архитектуру. Преимущества микросервисной архитектуры:

- Писать и поддерживать небольшие сервисы всегда проще, чем большие. Чем меньше кода, тем легче уместить его в голове;
- Для каждого сервиса можно выбрать язык и библиотеки, подходящие конкретно для решаемой этим сервисом задачи;
- Получаем горизонтально масштабируемый и отказоустойчивый код;
- Модульность;
- Нетрудно отследить зависимости между сервисами.

На рис. 2 приведена схема архитектуры системы, построенная с учётом высказанных выше соображений.

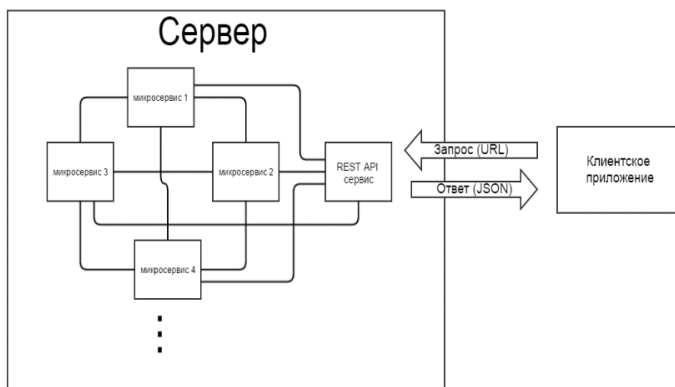


Рис. 2 – Архитектура системы

В качестве технологий возможно использовать программные платформы, которые предоставляют возможность создания приложений с микросервисной архитектурой. Одной из таких технологий является Node.js. Node.js имеет ряд преимуществ:

- REST / JSON API. Node.js блестяще справляется с построением легковесных REST / JSON интерфейсов. Неблокирующий ввод-вывод и JavaScript делают Node отличным вариантом для написания обертки вокруг базы данных или веб-сервиса, которая общается с клиентом в формате JSON;
- односторонние приложения. Для современных веб-приложений, делающих большую часть работы в браузере, отлично подходит сервер, который может одновременно обрабатывать тысячи запросов и имеет низкое время отклика;
- использование инструментов командной строки. Является выгодным использовать огромный выбор уже существующих инструментов командной строки;
- потоковая обработка данных. Традиционные стеки веб-приложений обычно трактуют запросы и ответы HTTP, как атомарные события. Но на самом деле это потоки, и приложения Node.js могут воспользоваться этим фактом;
- системы реального времени.

В качестве формата обмена данными предлагается использовать JSON. В первую очередь, если останавливать свой выбор на Node.js, то, как сказано выше, JSON отлично подходит в качестве формата при построении легковесных REST / JSON интерфейсов на Node.js. При этом JSON имеет ряд преимуществ над XML:

- занимает меньше места;
- прямо интерпретируется с помощью Javascript в объекты;

– за счет своей лаконичности, формат JSON больше подходит для сериализации сложных структур.

Для хранения данных предлагается использовать документоориентированную систему управления базами данных MongoDB. Основными преимуществами такой СУБД являются:

– документоориентированное хранение (JSON-подобная схема данных). Очень выгодно и удобно при разработке легковесных REST / JSON интерфейсов;

– Javascript как язык для формирования запросов. Очень выгодно и удобно, если в качестве технологии, используемой в серверном приложении, выбирать Node.js;

– динамические запросы;

– поддержка индексов;

– эффективное хранение двоичных данных больших объемов, например, фото и видео.

Выводы. В результате анализа подходов и систем управления проектами показано, что целеориентированный подход решает проблемы с прозрачностью, гибкостью и масштабированием при управлении проектами. В ходе анализа архитектур и целеориентирования выяснено, что необходимая система управления целеориентированными проектами должна иметь распределённую архитектуру. При этом архитектура серверной части может быть реализована на принципе сервис-ориентирования с применением микросервисов, что обеспечит ряд преимуществ, таких как: модульность, масштабируемость, отказоустойчивость и доступность. Выполненный анализ современных средств разработки позволил предложить один из вариантов стека технологий для реализации такой системы. Направлением дальнейших исследований является создание и исследование прототипа системы управления целеориентированными проектами с дальнейшей разработкой коммерческого варианта.

Список литературы

1. Целеориентированная автоматизация систем управления [электронный источник] // URL: www.c2ways.ru

2. Преимущества и недостатки микросервисной архитектуры [электронный ресурс] // URL: <http://eax.me/micro-service-architecture/>

3. Клиент-сервер [электронный ресурс] // URL: https://ru.wikipedia.org/wiki/Клиент_сервер

4. Трехзвенная (многозвенная) архитектура "клиент – сервер" [электронный ресурс] // Интуит – открытый университет: [сайт]. [2014]. URL: <http://www.intuit.ru/studies/courses/508/364/lecture/8643>

УДК 004.62:004.652:004.77

Распределённая информационная система совместной поддержки единой базы граждан

Толстых В.К., Киселева Л.Н.

Донецкий национальный университет, кафедра компьютерных технологий
г. Донецк,

Аннотация

Толстых В.К., Киселева Л.Н. Распределённая информационная система совместной поддержки единой базы граждан. Рассмотрен способ создания, потребления и сопровождения личных дел граждан посредством единой регистрационной базы данных (ЕРБД), расположенной в Internet. Предложена распределённая сервис-ориентированная архитектура информационных систем организаций (ИСО) с общим центром в виде ЕРБД. Описано взаимодействие граждан и ИСО с ЕРБД, позволяющее гражданам экономить личное время и контролировать свои данные, а организациям сокращать стоимость разработки и поддержки своих информационных систем. Описаны требования к Web-сайту и операциям Web-сервисов ЕРБД.

***Ключевые слова:** регистрация, база данных, личное дело, сервис-ориентированная архитектура, информационная система, Web-сервис.*

Введение. Регистрационные данные о гражданах в большей или меньшей степени собираются и поддерживаются всеми организациями, учреждениями. Это – их персонал, клиенты, пациенты, студенты и прочие категории контингента. Обычно регистрационными данными являются паспортные данные, документы об образовании, о состоянии здоровья, контактная информация и т. п. Сегодня, при обращении каждого гражданина в новую организацию, на него заводят новое личное дело и, если это современная организация с автоматизированной системой управления, то данные из личного дела вводят в информационную систему организации (ИСО).

Очевидно, что такая концепция множественного локального создания и хранения личных дел граждан не соответствует тенденциям развития современного информационного общества. Многократный ручной ввод и хранение одних и тех же данных, причём не всегда актуальных, – это расточительство для государства и неудобство для его граждан. Такие корпоративные базы ИСО, не взаимодействуют ни друг с другом, ни с гражданами, очевидно, – это «тупик» для автоматизации государственных процессов в целом. Некоторые продвижения в этом направлении уже имеются, например, [1,2] и др.

В работах [3-7] предлагается решать проблему сбора регистрационных данных и их поддержку в актуальном состоянии с помощью единой регистрационной базы данных (ЕРБД). Наличие ЕРБД в Internet-пространстве открывает новые возможности для построения ИСО. Архитектура ИСО теперь может быть распределённой сервис-ориентированной архитектурой (СОА) [8-10]. Здесь элементы ИСО, для доступа к данным контингента, трансформируются в операции Web-сервисов ЕРБД. Такая архитектура ИСО позволяет организациям хранить, модифицировать и автоматизировано потреблять в реальном времени из единого места готовые актуальные регистрационные данные граждан. При этом сами граждане смогут через Internet контролировать корректность существующих своих данных в ЕРБД и вносить новые данные, оставляя для организаций только функции проверки и подтверждения новых данных. Сразу же отметим, что ИСО может получить доступ к данным личности ЕРБД только после обращения гражданина в эту организацию и открытия им доступа к необходимой части своих данных для этой организации.

Использование СОА в информационных системах организаций позволит организациям отказаться от разработки и поддержки некоторых частей собственного программного обеспечения и баз данных. Достаточно подключиться к необходимым операциям Web-сервиса ЕРБД. В такой архитектуре ИСО приобретает виртуальный распределённый характер. Логика доступа к данным своего контингента строится на основе вызова внешних операций, а внешние операции обеспечивают доставку актуальных регистрационных данных контингента. В такой архитектуре ИСО сокращаются сроки и стоимость разработки своей информационной системы, повышается её масштабируемость, улучшается управляемость создаваемой системы и снижаются издержки на поддержку данных в актуальном состоянии.

Таким образом, ЕРБД позволит экономить не только финансовые, человеческие и временные ресурсы, но и откроет принципиально новые возможности автоматизации управления контингентом граждан в различных сферах деятельности всего государства, позволит быстро и с минимальными затратами проводить статистические и социологические исследования на основе накапливающихся данных ЕРБД.

Естественно, ЕРБД не должна заменять ведомственные базы, в которых содержится специализированная информация о гражданах (медицинская, коммерческая...), она должна быть их ядром, через которое ведомства, как и все другие организации, могут пополнять и синхронизировать основные данные своих баз. При этом ведомства могут даже отказаться от хранения дублирующей информации.

Особенно актуально внедрение ЕРБД в сфере образования от школ до вузов. Именно в школах создаются массовые, объёмные регистрационные данные (личные дела учеников), которые, к сожалению, в большинстве случаев хранятся

в бумажном виде и, естественно, не участвуют в процессах автоматизации ни школ, ни отделов, управлений образования, ни всего образования в целом.

Распределённая архитектура взаимодействия граждан и ИСО через ЕРБД. На рис.1. изображена распределённая архитектура взаимодействия ИСО с личными делами граждан через ЕРБД. Сам узел ЕРБД состоит из трёх частей: база регистрационных данных; Web-сайт; Web-сервис. Web-сайт обеспечивает неавтоматизированный (не машинный) доступ к данным всем категориям пользователей ЕРБД для поиска, просмотра и контроля данных, для создания и подтверждения новых данных граждан, а также для формирования аналитических отчетов по имеющимся данным.

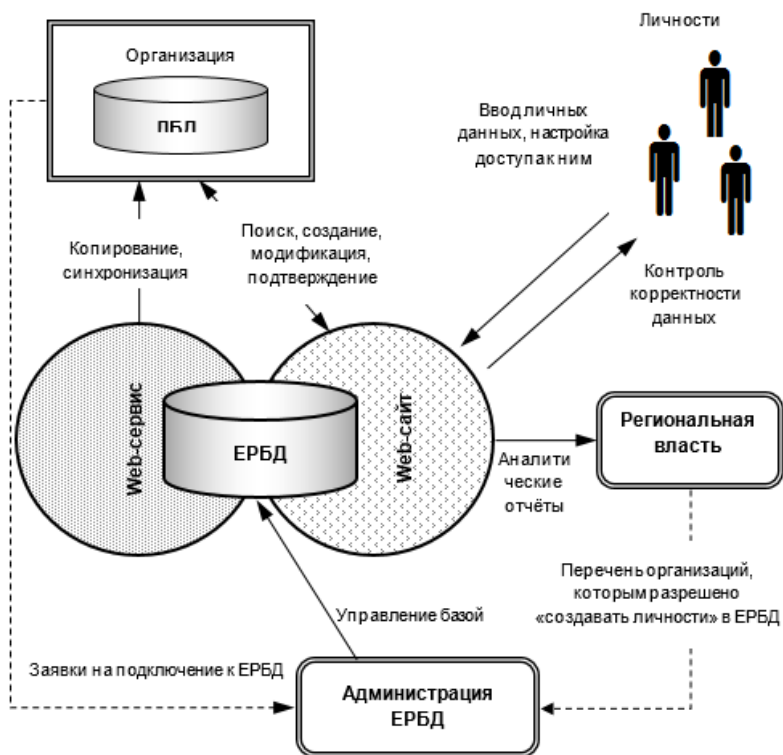


Рис. 1. Распределённая архитектура взаимодействия ИСО с личными делами граждан через ЕРБД

Кроме того, посредством Web-сайта граждане могут удалённо регистрироваться в организациях, вводить новые данные о себе (временно не подтвержденные), делать настройки доступа к своим данным для различных

организаций. Web-сервис ЕРБД предоставляет набор операций для автоматизированного доступа на копирование и синхронизацию данных ЕРБД с локальными базами данных (ЛБД) ИСО.

Организации, принимающие участие в создании, поддержке и потреблении данных отличаются друг от друга по функциональным возможностям работы с данными ЕРБД. Региональной властью определяется перечень организаций (ведомств, объединений...) имеющих право на создание, модификацию и подтверждение регистрационных данных граждан. Организации не из перечня региональной власти могут получить доступ только на потребление регистрационных данных. Для работы с ЕРБД организациям необходимо подать заявку администратору ЕРБД на авторизированное подключение.

Первая регистрация граждан в ЕРБД осуществляется при обращении их в организации, наделённые правами создания записей в базе. Регистрация осуществляется на основании документа, удостоверяющего личность. Регистратор (ответственное лицо за ввод, подтверждение и модификацию данных) такой организации создает в ЕРБД новую запись о личности с начальными данными (ФИО личности, ее дата и место рождения, пол, данные предъявленного документа, а также контактные данные). Регистратор так же имеет возможность создания других записей, касающихся личности (данные различных документов, информацию, необходимую для внутренней работы организации, данные о родителях, детях и т.д.).

Зарегистрированный гражданин (личность) получает логин и пароль для авторизированного подключения к Web-сайту ЕРБД, где он может просматривать и контролировать корректность своих данных, при необходимости, модифицировать контактные данные, пароль, добавлять новые данные, временно не подтвержденные регистратором организации, уполномоченной региональными властями.

Гражданам, зарегистрированным в ЕРБД, при обращении в какие-либо организации, подключенные к ЕРБД, рекомендуется заранее произвести настройки прав доступа этих организаций к своим данным. Регистраторы таких организаций через браузер подключаются к Web-сайту ЕРБД, авторизуются, находят требуемую личность и производят регистрацию личности в своей ИСО посредством соответствующей операции копирования Web-сервиса ЕРБД.

Создание и подтверждение ранее созданных личностью данных, осуществляется регистратором организации (при наличии таковых полномочий) на основании предъявленных личностью документов. При этом доступ к данным личности получают все необходимые сотрудники организации, в объеме, который установила личность для этой организации. При необходимости, организация осуществляет синхронизацию данных своего контингента в ЛБД ИСО на основе данных ЕРБД с помощью соответствующих операций синхронизации Web-сервиса ЕРБД.

Сервисы ЕРБД для доступа ИСО. Главные задачи, которые должны решаться ИСО при взаимодействии с ЕРБД, – это копирование новых регистрационных данных нужных личностей из ЕРБД в ЛБД и, в дальнейшем, – синхронизация обновлённых данных ЕРБД в ЛБД.

Копирование может реализовываться двумя способами: копирование данных каждой личности отдельно и копирование данных группы личностей.

Первый случай необходимо использовать при регистрации в ИСО новых личностей из ЕРБД. Регистратор, после просмотра данных личности на Web-сайте ЕРБД, копирует её данные в ЛБД. При этом в ЕРБД делается отметка о регистрации текущей личности в организации и запрашивается соответствующее Web-приложение ИСО. Этому приложению по протоколу HTTPS передаются параметры, идентифицирующие регистратора и выбранную личность: логин регистратора из ЕРБД, пароль регистратора, а также идентификатор личности ID из ЕРБД.

Передаваемые учётные данные регистратора можно использовать для реализации корпоративных алгоритмов авторизации. Web-приложение ИСО, принимающее эти данные, в ответ должно осуществить подключение к Web-сервису (или WCF-сервису) ЕРБД для запроса на копирование данных личности в авторизованном сеансе. Для этого целесообразно вызывать первой отдельную операцию авторизации и создания сеанса с сервисом ЕРБД. Далее можно вызывать необходимые операции для получения требуемых данных о личности и последующей записи в ЛБД. Обращаем внимание, что в ЛБД обязательно должен записываться ID личности для возможности дальнейшей синхронизации контингента с ЕРБД.

Второй случай копирования необходимо применять для доступа к дополнительным данным ЕРБД личностей, ранее зарегистрированных в ЛБД и имеющих ID-идентификаторы. Этот способ целесообразно использовать для небольших компаний или мобильных устройств, когда нецелесообразно локально хранить редко используемые и часто изменяемые регистрационные данные. Можно хранить в ЛБД только ID и может быть ещё – ФИО контингента. В этом случае для доступа к данным какой-либо личности в сеансе сервиса копирования, необходимо передавать идентификатор личности. Таким образом, можно в одном сеансе работать с целой группой личностей ЕРБД и в on-line режиме получать любые необходимые данные контингента. Напомним, что основное время сетевых взаимодействий тратится на установление связи клиента с сервером, т.е. на создание сеанса. В то время как на передачу данных в открытом сеансе затрачивается незначительные сетевые ресурсы.

Очевидно, что для построения операций синхронизации необходимо использовать уже имеющиеся операции копирования, которые должны вызываться для получения данных личности, требующих синхронизацию. Отличие синхронизации от копирования будет заключаться в том, что операции синхронизации должны работать

с дополнительными параметрами – датами запрашиваемых данных ЛБД и ЕРБД. Именно на основании этих дат принимается решение либо о передачи обновлённых данных для ЛБД, либо о сообщении отсутствия необходимости синхронизации каких-либо данных для текущей личности.

Выводы. Рассмотренная распределённая СОА ИСО с единым местом хранения регистрационных данных граждан в ЕРБД позволит организациям (кроме первой организации-регистратора) избавиться от необходимости рутинной ручной регистрации граждан, которые к ним поступают, что особенно актуально при массовых регистрациях абитуриентов в учебных заведениях. Такая архитектура позволит осуществлять многосторонний контроль данных ЕРБД как организациями, так и самими гражданами. Распределённая ИСО позволит организациям производить удалённую регистрацию граждан (копирование личных дел в свои локальные базы) с их согласия даже без непосредственной встречи с ними и предъявления ими документов, поскольку все документы ЕРБД, доступные для копирования, являются уже подтверждёнными уполномоченными организациями-представителями региональной власти.

Личности смогут самостоятельно однократно вводить и контролировать свои данные, смогут по своему усмотрению открывать или ограничивать для различных организаций доступ к личным данным, смогут самостоятельно поддерживать контактные данные в актуальном состоянии. Кроме того, граждане смогут удалённо регистрироваться в различных организациях, экономя личное время, эмоциональные и физические силы.

Такая архитектура позволит организациям относительно быстро и дёшево создавать, потреблять и поддерживать регистрационные данные своего контингента в актуальном состоянии. Организации, оказывающие услуги гражданам в реальном времени, могут вообще отказаться от создания и поддержки собственных регистрационных баз данных. В этом случае их данные будут полностью находиться в ЕРБД. Достаточно в нужное время осуществить Internet-подключение к ЕРБД.

ЕРБД позволит экономить не только человеческие и временные ресурсы, но открывает принципиально новые возможности автоматизации управления в различных сферах деятельности всего государства, позволит быстро и с минимальными затратами проводить статистические и социологические исследования. ЕРБД даст возможность автоматизировать перепись населения, избирательные компании, приемные компании учебных заведений, развивать системы контроля успеваемости школьников и студентов, работу центров тестирования, налоговых служб, пенсионных фондов, повысить качество государственного планирования, организовать массовое адресное информирование граждан и многое другое.

Список литературы

1. Сетевой город. Образование: построение единой информационно-образовательной среды муниципального образования. — Режим доступа: <http://www.irtech.ru>.
2. *Пююа Маргус*. Опыт развития электронных услуг в Эстонии. — Режим доступа: <http://www.gosbook.ru/document/5532>.
3. *Толстых В.К.* О единой регистрационной базе данных граждан. — Системы обработки информации, № 3, Т. 2. — 2012. — С. 120-123.
4. *Толстых В. К., Киселёва Л. Н.* Концепция единой регистрационной базы данных граждан. — Наука та інновації, №4, Т. 8. — 2012. — С. 67-74.
5. *Толстых В. К., Киселёва Л. Н.* О единой регистрационной базе граждан и структуре данных личности в этой базе. — Реєстрація, зберігання і обробка даних, №3, Т. 13. — 2011. — С. 81-91.
6. Патент 60436 Україна, МПК G06Q 90/00, G06F 17/40, G06F 17/30. Спосіб формування єдиної реєстраційної бази даних громадян / *Толстих В.К., Кисельова Л.Н.*; заявники та власники Донецький національний ун-т, Толстих В.К., Кисельова Л.Н. — № u201011884; заявл. 07.10.2010; опубл. 25.06.2011, Бюл. № 12/2011.
7. Патент 60435 Україна, МПК G06Q 90/00, G06F 17/40. Спосіб контролю та споживання даних єдиної реєстраційної бази даних громадян / *Толстих В.К., Кисельова Л.Н.*; заявники та власники Донецький національний ун-т, Толстих В.К., Кисельова Л.Н. — № u201011883; заявл. 07.10.2010; опубл. 25.06.2011, Бюл. № 12/2011.
8. *Данилов И.Г.* Сервис-ориентированная архитектура как основа для построения современных распределённых систем // Известия ЮФУ. Технические науки. — 2010. — №7 — С.168-173.
9. *Богданов А.В., Станкова Е.Н., Мареев В.В.* Сервис-ориентированная архитектура: новые возможности в свете развития Grid технологий // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", 2008. — 32 с.
10. *Сатунина А.Е., Сысоев А.С.* Сервис-ориентированный подход к построению и функционированию корпоративной информационной системы [Электронный ресурс] // Современные проблемы науки и образования. — 2009. — № 6-1. — С. 33 — Режим доступа: URL: <http://www.science-education.ru/ru/article/view?id=1295>.

УДК 007.51

Проектирование компьютеризированной подсистемы учета и планирования выполнения заказов для малого предприятия по изготовлению печатных плат

Лысенко В.С., Землянская С.Ю

Донецкий Национальный Технический университет

кафедра автоматизированных систем управления

Vadik44444444@yandex.ua

Аннотация

Лысенко В.С., Землянская С.Ю. Компьютеризированная подсистема учета и планирования выполнения заказов для малого предприятия по изготовлению печатных плат. Рассмотрены основные отличия малого предприятия от большого, которые являются базисом для проектируемой подсистемы. Определены первичные, вторичные, цели третьего уровня, а также конечные цели подсистемы.

Ключевые слова: печатные платы, малые предприятия, планирование заказов, учет заказов, подсистема.

Актуальность. Любая деятельность человека в современном мире направлена на производство некоего выходного продукта. Скорость производства этого продукта, в свою очередь, зависит от производительности объекта. Соответственно, количество производимого продукта, а, следовательно, прибыль, прямо пропорциональна производительности предприятия.

Существенным моментом является, то, что группа предприятий «малые», делится, в свою очередь, еще на 3 разновидности, а именно:

- Микро предприятия
- Малые предприятия
- Средние предприятия

Каждая из этих разновидностей имеет свои особенности, существенно влияющие на подходы к автоматизации.

Также ключевым моментом является то, что численность представленной выше группы предприятий по состоянию на 2015 год составила 69,4%, что значительно превосходит количество крупных предприятий – всего 30,6%.

На основании приведенных данных можно утверждать, что данная подсистема имеет многочисленную широкую целевую аудиторию, состоящую из 3 частей, что лишь еще раз подчеркивает ее актуальность в рамках нынешних реалий.

Анализируя основные отличия большого предприятия от малого и среднего, можно сделать следующие выводы:

– На малом предприятии количество работников очень ограничено, в отличие от больших предприятий с сотнями сотрудников. Рассмотрим конкретные цифры.

– На малом или среднем предприятии количество работников, в лучшем случае, достигает от 15 до 250 человек, в случае большого – от 250 человек.

– Годовой денежный оборот также значительно уступает денежному обороту больших предприятий. Рассмотрим конкретные цифры. На малом предприятии денежный оборот достигает, в лучшем случае, (средние предприятия) 2 млрд рублей, в то время как на большом – более 2 млрд рублей, в среднем более 10 млрд рублей.

– Также есть различие в форме налогообложения. В отличие от большого предприятия малое или среднее предприятие платит «Единый налог», который выплачивается частным лицом и является ежемесячным платежом с фиксированной суммой, не зависящей от объемов продаж, в отличие от больших предприятий

– Для предприятий разных масштабов давно ведутся разработки, связанные с автоматизацией различных сфер деятельности этих предприятий, таких как управление финансовой деятельностью, кадрами, производственными процессами и прочими

Аналоги разрабатываемой подсистемы. Наиболее популярными из них являются ПО 1С: Предприятие, а также проект usu.kz, занимающийся разработкой подобных программ, в том числе и на заказ. В связи с тем, что данные продукты имеют высокую цену, выплатить которую небольшие предприятия за подобное ПО не в состоянии, разрабатываемая подсистема создается и адаптируется именно для таких предприятий, и будет финансово доступна для них.

Цели подсистемы. При разработке подсистемы должны быть достигнуты цели разных уровней. Рассмотрим их детальнее.

1. Первичные:
 - a. Систематизация процесса учета и планирования заказов
 - b. Уменьшение вероятности ошибки при учете и планировании заказов
 - c. Увеличение производительности предприятия
2. Вторичные:
 - a. Повышение репутации предприятия
 - b. Увеличение скорости производства предприятия
 - c. Повышение конкурентноспособности предприятия
3. Цели третьего уровня:
 - a. Расширение рынка сбыта
 - b. Привлечение квалифицированных кадров
4. Конечная цель:
 - a. Увеличение абсолютной прибыли

Структура подсистемы

Представленная подсистема проектируется с помощью программы *Enterprise Architect* на унифицированном языке моделирования.

Действующие лица в разрабатываемой системе:

- Директор
- Заказчик
- Главный инженер
- Заведующий складом
- Плановый отдел

Заказчик производит заказ, который попадает к директору на рассмотрение. Директор как руководитель передает информацию, полученную от заказчика, главному инженеру, для того, чтобы главный инженер смог произвести комплексный анализ этой информации и предоставить результаты директору, из которых будет следовать, сможет ли данное предприятие выполнить заказ.

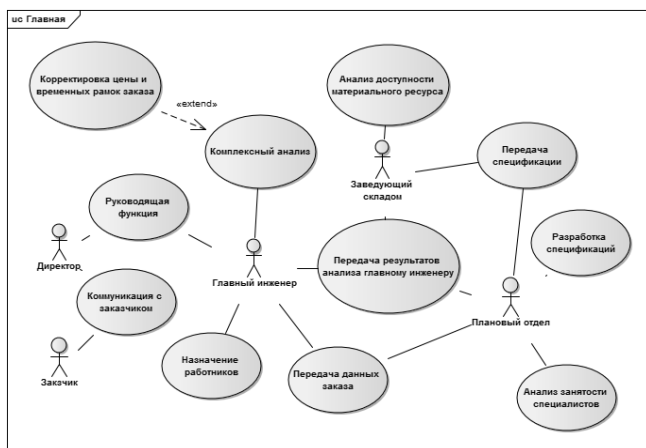


Рис.1. Общая диаграмма деятельности

Рассмотрим декомпозицию руководящей функции в рамках данной подсистемы:

Директор осуществляет коммуникацию с подчиненными по разнообразным производственным вопросам, в том числе и раздаче заданий, а также производит прием отчетов и результатов выполненной работы, а при необходимости вносит корректировки в вышеперечисленные документы.

После этого главный инженер передает полученную информацию плановому отделу для последующей работы, необходимой для осуществления комплексного анализа.

Плановый отдел занимается *разработкой спецификации* для заказанной продукции. Рассмотрим детальнее процесс *разработки спецификации*. Он состоит из следующих действий:

- Составление списка необходимых для выполнения заказа специалистов
- Составление списка необходимых для выполнения заказа работ
- Составление списка необходимых для выполнения заказа материальных ресурсов
- Расчет себестоимости продукта

Кроме составления списка необходимых для выполнения заказа специалистов, плановый отдел также анализирует занятость специалистов, что покажет, есть ли свободные люди для выполнения данного заказа.

Рассмотрим детальнее, из чего состоит расчет себестоимости продукта. Для того, чтобы рассчитать себестоимость продукта, плановому отделу необходимо будет произвести следующие расчеты:

- Стоимости работ, производимых специалистами
- Стоимости затраченных расходных материалов
- Командировочных расходов
- Общепроизводственных расходов
- Стоимости топлива и энергии, которые будут затрачены для выполнения данного заказа
- Отчислений во внебюджетные фонды

После разработки спецификации плановый отдел передает спецификацию заведующему складом который, в свою очередь, производит анализ доступности материального ресурса, который покажет, хватает ли материалов для выполнения заказа по заявленному списку необходимых материальных ресурсов в спецификации, разработанной плановым отделом.

После всех перечисленных действий, заведующий складом и плановый отдел передают результаты своей работы главному инженеру, который производит комплексный анализ, необходимый для окончательного решения, сможет ли предприятие выполнить заказ в указанные сроки, а также назначает свободных работников на выполнение заказа. По своему усмотрению, главный инженер корректирует цену и временные рамки заказа, если предприятие не может выполнить заказ, например, если анализ доступности специалистов показал, что какие-то специалисты заняты в заданные временные рамки, после чего передает отчет директору.

База данных подсистемы. Для хранения информации, необходимой для функционирования подсистемы, была спроектирована следующая база данных (рис.2).

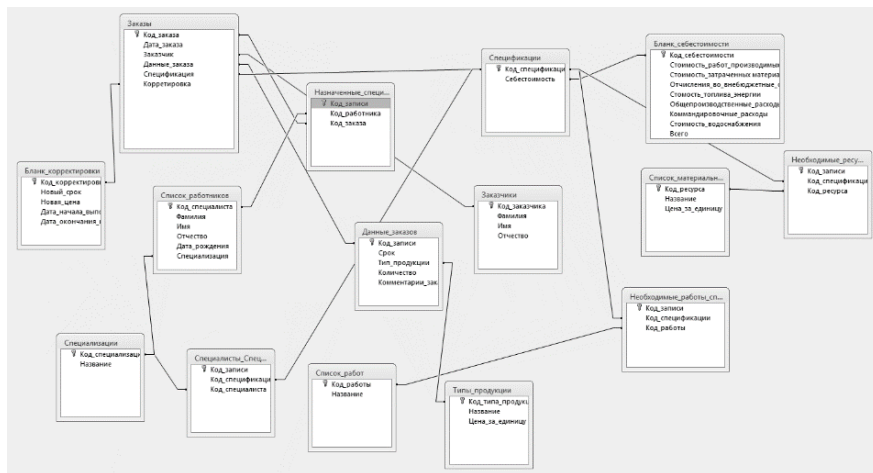


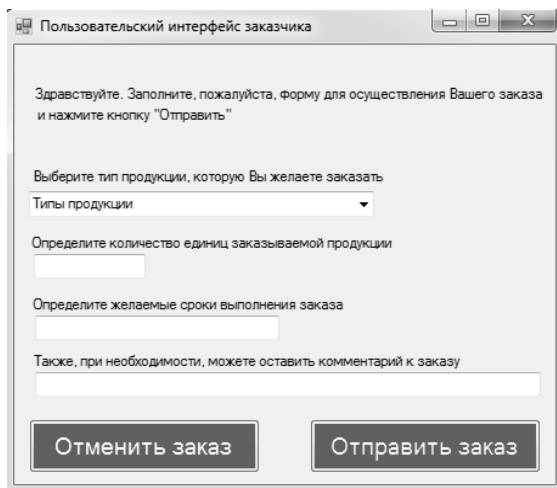
Рис.2. Логическая схема базы данных

Стоит также отметить, что системой предусмотрен заказ **однотипной** продукции в одном заказе, если нужно заказать разную продукцию, производится несколько заказов.

Структура экранной формы. Экранные формы для создаваемой подсистемы разрабатываются с адаптацией под максимально возможную целевую аудиторию, обладающую различными уровнями владения компьютерными навыками. Поэтому создается интуитивно понятный пользовательский интерфейс. Пример структуры экранной формы можно увидеть на рис.3

Подсистема позиционируется как многопользовательская, из чего следует, что в ней планируется реализовать множество пользовательских интерфейсов, а именно для каждого действующего лица подсистемы:

- Директор
- Заказчик
- Главный инженер
- Заведующий складом
- Плановый отдел



Пользовательский интерфейс заказчика

Здравствуйте. Заполните, пожалуйста, форму для осуществления Вашего заказа и нажмите кнопку "Отправить"

Выберите тип продукции, которую Вы желаете заказать

Типы продукции

Определите количество единиц заказываемой продукции

Определите желаемые сроки выполнения заказа

Также, при необходимости, можете оставить комментарий к заказу

Рис.3. Структура пользовательского интерфейса заказчика

Естественно, каждый интерфейс будет иметь свой функционал, совпадающий с обязанностями и возможностями каждого действующего лица подсистемы.

Подсистема учета и планирования выполнения заказов для малого предприятия по изготовлению печатных плат должна стать финансово и функционально подходящим выходом из затруднительного положения для малых предприятий, что должно помочь им увеличить скорость своего производства, расширить рынок сбыта, а, следовательно, и увеличить прибыль.

Список литературы

1. Кузнецов В.В. Экономические методы управления предпринимательской деятельностью. Деловая игра [Текст]/ Кузнецов В.В., Кондратьева М.Н., Алашеева О.В - Ульяновск: УлГТУ, 2001. - 20 с.
2. Лешек А. Мацяшек, Анализ требований и проектирование систем. Разработка информационных систем с использованием UML [Текст]/ Лешек А. Мацяшек – Вильямс 2002-432 с
3. Антонов А.В. Системный анализ. [Текст]/ Антонов А.В. - М.: Высшая школа, 2004. - 454 с.
4. Роберт Дж. Мюллер, Базы данных и UML [Текст]/ Роберт Дж. Мюллер – Лори, 2002 – 420 с.

УДК 004.65

**Спецификации номенклатуры – хранение и обработка в различных
ERP-системах
Харитонов Ю.Е.**

Донецкий национальный технический университет, г. Донецк
кафедра экономической кибернетики
u-e-c-h-a-r@yandex.ru

Аннотация

Харитонов Ю.Е. Спецификации номенклатуры – хранение и обработка в различных ERP-системах. *Выполнен обзор методов хранения и обработки конструкторских спецификаций в различных системах управления предприятием. Предложен реляционно-программный подход к процедуре разузлования.*

Ключевые слова: *спецификации номенклатуры, разузлование, иерархические базы данных, реляционно-программный подход.*

Постановка проблемы. Автоматизация производственного учета характеризуется переходом рынка внедрений от стадии «первоначального насыщения» к стадии «рационального использования». Задача обеспечения простоты первоначального развертывания во многом зависит от понимания механизмов работы с основными исходными данными, к которым несомненно относятся спецификации номенклатуры.

Цель статьи – обзор методов хранения и обработки спецификаций номенклатуры в наиболее рейтинговых ERP-системах.

Анализ методов хранения спецификаций. В основе всех существующих систем управления промышленным предприятием (PDM/PLM/ERP) лежит информация о номенклатуре выпускаемой продукции. Она включает как данные о составе изделий (конструкторские спецификации, Bill of Material), так и данные о технологии изготовления изделий (маршрутные карты, Bill of Operations).

Процедуру определения связей между различными уровнями изделия называют разузлованием (explosion). Схема разузлования позволяет представить сложное изделие как совокупность элементарных деталей, входящих в его состав.

По оценкам аналитиков Центра Tadviser [1], в России по количеству проектов ERP абсолютным лидером является компания «1С». Только на платформе «1С:Предприятие 8.0» реализовано более 11,6 тыс. комплексных проектов (т.е. внедрен полный функционал). По состоянию на июль 2015 г. за всю историю наблюдений Tadviser (с 2005 г.) наиболее часто внедряемыми платформами можно назвать помимо «1С:Предприятие 8.0» также и «Галактика

ERP», Microsoft Dynamics AX, Microsoft Dynamics NAV, SAP ERP). Рассмотрим подробно методы хранения спецификаций для каждой из перечисленных систем.

В состав системы IC:PDM входит конструкторско-технологический справочник (КТС). Справочник интегрирован со справочником общей номенклатуры предприятия и основан на механизмах PDM системы, что позволяет заимствовать его элементы непосредственно в технологические процессы или составы изделий.

КТС является средством для ведения нормативно-справочной информации по материалам, стандартным и прочим изделиям. КТС организован в виде иерархической структуры, образованной папками. Конечным элементом иерархии является понятие ресурса. Ресурс не может содержать подчиненные папки, но может хранить внутри себя элементы определенного вида. Ресурс создается отдельно от папки иерархии и для него указывается набор свойств, присущих хранящимся в ресурсе элементам. Для различных ресурсов (даже хранящих один вид элемента) набор свойств может быть различным.

IC:УПП. Состав исходных комплектующих спецификации можно сформировать с помощью специальной обработки «Конструктор спецификаций» (Справочники – Номенклатура – Конструктор спецификаций). Конструктор спецификаций позволяет увидеть полный перечень исходных комплектующих до самых начальных этапов изготовления изделия. Таким образом можно получить результат разузлования для каждого передела – законченной части технологического процесса.

ГАЛАКТИКА-ERP. Первоисточником конструкторско-технологической информации (данные о составе и технологии изготовления изделий, узлов и ДСЕ) является комплекс «INTERMECH».

Конструкторский состав изделий из комплекса «INTERMECH» переносится в систему «ГАЛАКТИКА-ERP» в виде множества объектов типа «Конструкторская спецификация». Это связано с тем, что в «ГАЛАКТИКА-ERP» конструкторские спецификации являются одноуровневыми документами, т.е. для каждой сборочной единицы в соответствующей ей конструкторской спецификации указываются только те узлы, детали, стандартные изделия, материалы, документация, комплекты, комплексы, программные средства, которые непосредственно входят в состав этой сборочной единицы.

Редактор IPS AVS (Intermech Professional Solutions) является модулем расширения системы IPS Search и позволяет редактировать состав сборочных единиц в виде конструкторской спецификации. В системе IPS спецификация – это форма представления состава изделия, поэтому в IPS невозможно добиться рассогласования состава и соответствующей ему конструкторской спецификации.

Microsoft Dynamics AX. Компоненты спецификации на каждом уровне настраиваются в строках спецификации. Критерии строк спецификации:

- в строках спецификации содержатся различные компоненты (или номенклатуры), связанные с данной спецификацией;
- каждая строка спецификации содержит назначенный номер, и для большинства целей может иметь неограниченное число строк (номенклатур), которые составляют спецификацию;
- в каждой строке спецификации в целях планирования, расчета и потребления должно быть указано, существуют ли для каждой номенклатуры какие-либо конфигурации.

Конфигурации спецификаций используются, когда для спецификации имеются несколько возможных комбинаций. Любая номенклатура в спецификации с конфигурацией должна присоединяться к конфигурационной группе в строке спецификации. Последовательность конфигурирования (конфигурационный маршрут) для спецификации и все правила (правила конфигурации) настраиваются в спецификации. Конфигурационные группы используются для упорядочения компонентов в спецификации, чтобы они могли изготавливаться вместе.

Анализ в Microsoft Dynamics AX основан на технологии аналитической обработки в реальном времени (OLAP), которая ускоряет многомерный анализ благодаря использованию служб аналитики MS SQL Server. В Microsoft Dynamics AX функция OLAP и система ERP интегрированы таким образом, что все сопоставления происходят непосредственно в Microsoft Dynamics AX, что позволяет в полной мере воспользоваться преимуществами бизнес-информации и создавать большинство отношений между таблицами, расширенными типами данных и метками перечисляемых элементов.

ERP-система R/3 компании SAP AG обладает всеми необходимыми свойствами для успешного решения задач поддержки основной производственной деятельности корпорации и управления бизнес-процессами на основе единого информационного пространства. Развития PDM-технологии является ее четвертое поколение, так называемое cPDM (collaborative Product Definition management – совместное управление определением продукта), представляющее собой Web-ориентированные системы PDM. Для систем категории cPDM важна полная координация и связь этих процессов.

Подходы к обработке спецификаций. Иерархическая модель хранения спецификаций представляет собой связный неориентированный граф древовидной структуры, объединяющий сегменты – узлы нижнего уровня. Иерархическая БД состоит из упорядоченного набора деревьев.

В 1С данные хранятся в виде объектов 1С – набора реквизитов и табличных частей плюс методы (функции). Объекты 1С бывают разных видов (справочники, документы и т.п.), хранятся как одна строка в таблице SQL (для шапки) и несколько строк в других таблицах SQL (для каждой табличной части, если таковые есть).

Как было указано выше «ГАЛАКТИКА-ERP» хранит конструкторские спецификации в виде одноуровневых документов для каждой сборочной единицы – объектов соответствующего типа.

Microsoft Dynamics AX выполняет аналитическую обработку данных в реальном времени (OLAP). Эта технология ускоряет многомерный анализ благодаря использованию служб аналитики MS SQL Server.

ERP-система R/3 является Web-ориентированной системой, основанной на комплексе аналитических приложений Business Intelligence, который поддерживает XML, XMLA, ODBO, Common Warehouse Metadata Interchange (CWM), интерфейс программирования бизнес-приложений BAPI.

Вернемся к платформе 1С, в которой все прикладные решения поставляются в исходных кодах, и рассмотрим механизм разувязания номенклатуры : определения по спецификации изделия состава и количества входящих в него составляющих. Процедура «РазувязаниеНоменклатуры» в 1С УПП используется в обработках «Отчет производства за смену» и «Помощник планирования» и применяет рекурсивный алгоритм обхода графа спецификаций [2]. Там же приведен пример последовательного поуровневого обхода графа – принципиально "нерекурсивный" метод решения.

Для создания реляционной модели БД спецификаций номенклатуры предлагается матрицу смежности представить в виде отношения, где один атрибут представляет родительский узел, а другие – дочерние узлы. Можно создать отношение с количеством атрибутов, равным уровням иерархии N-мерного дерева, а глубину вхождения узла или детали в узел верхнего уровня ограничить признаком, например ненулевым количеством соответствующих деталей/узлов в родительском узле. В результате каждый кортеж будет отражать «генетически» полную информацию о содержащихся в изделии узлах/деталях с ненулевым количеством, что расширяет диапазон использования необходимых SQL-запросов и упрощает программирование процедур обработки данных. При расчете расхода материалов на изделие необходимо выполнить процедуру сборки количеств деталей по таблице вхождений.

Список литературы

1. Системы управления предприятием (рынок_России) // Аналитический центр TADVISER: [сайт]. [2015]. URL: www.tadviser.ru/index.php/ Статья: Системы управления предприятием (рынок России)
2. Запрос против рекурсии или разувязание номенклатуры// Информационно-аналитический центр по автоматизации учета и управления: [сайт]. [2010-2011]. URL: <http://infostart.ru/public/78285/>

УДК 004.41

Разработка веб-ориентированной подсистемы доступа к медицинским изображениям

Симоненко А.А., Привалов М.В.

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
E-mail: sunnyday.styling@gmail.com

Аннотация

Симоненко А.А., Привалов М.В. Разработка веб-ориентированной подсистемы доступа к медицинским изображениям. Выполнен анализ процесса удаленного доступа к медицинским данным в PACS. Разработана функциональная структура веб-ориентированной подсистемы авторизированного доступа к DICOM-данным.

Ключевые слова: медицинские данные, изображения, DICOM, PACS, Internet

Постановка проблемы. В современной медицине используется масса разнообразной техники (КТ, МРТ, КР и т.д.) необходимой для изучения внутреннего строения человека, диагностирования заболеваний или повреждений внутренних органов, так же данные такого рода используются для обучения молодых специалистов и научных исследований: разработки новых методов лечения и диагностирования. Это приводит нас к тому, что такие данные должны быть мобильны в рамках медицинских учреждений и за их пределами.

Цель статьи – провести анализ процесса удаленного доступа к медицинским данным в PACS и разработать функциональную структуру веб-ориентированной подсистемы авторизированного доступа к DICOM-данным.

Архитектура рассматриваемой информационной системы. Стандартом DICOM (Digital Imaging and Communications in Medicine)[1] описаны основные принципы, работы с медицинскими изображениями, но возникает вопрос хранения с учетом значительных размеров кадров и серий. Решение задачи хранения привело к возникновению целого класса систем *Picture Archiving and Communication System*, далее PACS.

PACS – это системы передачи и архивации DICOM изображений[2], предполагают создание специальных удаленных архивов на DICOM Server-ax, где весьма объемный архив может длительное время существовать в «горячем» виде и быть быстро доступным для поиска и просмотра интересующей информации по DICOM сети.

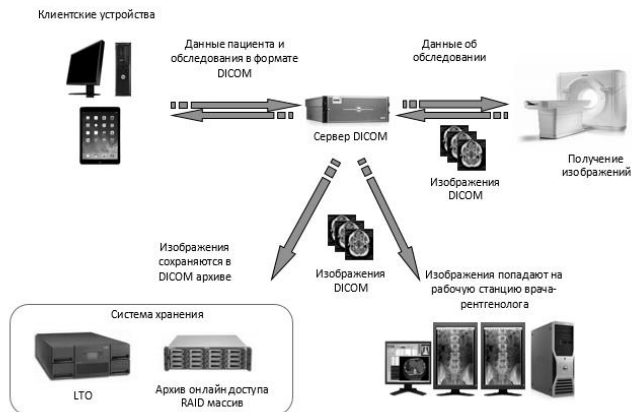


Рис. 3 – Архитектура PACS

Анализ существующих средств удаленного доступа в PACS. На данный момент существует множество решений по созданию PACS с модулями веб-ориентированного доступа, рассмотрим наиболее известные и широко используемые.

Компания Philips и её решение IntelliSpace PACS Anywhere. Данный программный продукт является свободным, но используется совместно с системой IntelliSpace PACS Enterprise. IntelliSpace PACS Anywhere предоставляет полный доступ к данным (DICOM – формата) пациентов практически с любого мобильного устройства. Данная система предназначена только для просмотра уже существующих данных и имеет некоторые ограничения:

Разрешающая способность устройства должна быть не менее 1024x768.

Предполагается наличие высокоскоростной сети, между клиентом и сервером.

Компания Siemens и её решение Syngo.Via. Данный программный продукт схож с разработкой компании Philips, существенным отличием стал более жесткий контроль безопасности данных пациентов. Персональные данные не хранятся локально, а могут быть получены только при наличии доступа к ним у каждого конкретного врача. Данный функционал реализуется путем выдачи каждому персонального логина и пароля, и необходимости их ввода для получения данных.

Оценка недостатков существующих решений. Исходя из рассмотренных решения, мы сталкиваемся с 2 основными проблемами:

- Анонимность используемых данных
- Пересылка больших объемов данных через сеть (особенно актуально в сетях с низкой пропускной способностью).

Первая проблема связана с тем, что базы данных с DICOM-файлами должны быть «частично» открытыми, единственным ограничением является распределенный доступ к персональным данным пациентов. Данное ограничение связано с врачебной тайной, именно поэтому только лечащий врач должен иметь полный доступ к личным данным пациентов. Остальные пользователи системы, могут рассматривать лишь непосредственные дата-сетсы.

Вторая проблема связана с достаточно большим размером дата-сетов. Они могут колебаться в пределах от 5 мегабайт до нескольких гигабайт. Отсюда следует, что для пересылки их с сервера на клиентское приложение требуются сети с большой пропускной способностью или же, затраты значительного количества времени, что является недопустимым во многих случаях.

Решение описанных проблем предполагает внесение изменений в архитектуру PACS на участке взаимодействия клиентских устройств и DICOM сервера. Перейдем к подробному анализу логики работы каждого модуля (рис.1) и подсистемы, а так же рассмотрению технологий для их реализации.

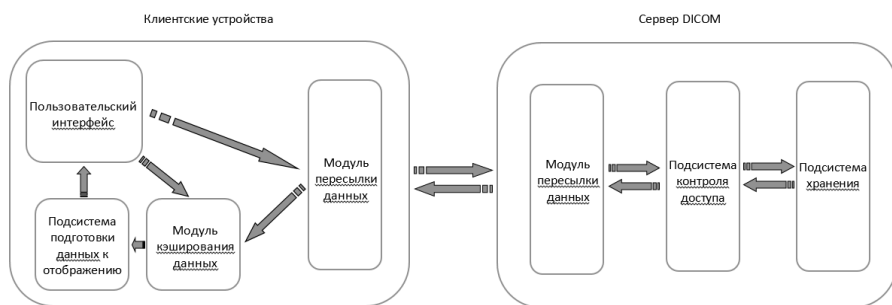


Рис. 2 – Предлагаемое архитектурное решение

1. Пользовательский интерфейс необходим для удобства взаимодействия пользователя с данными. Для его реализации предполагается использование современных веб-технологий таких как HTML5, CSS, JS, AngularJS. Таким образом интерфейс предоставит пользователю все необходимые инструменты для просмотра медицинских изображений, так же UI будет использоваться для авторизации, при необходимости получения персональных данных пациентов.

2. Модуль пересылки данных присутствует, как на клиентской так и на серверной стороне. Технология JSON, используемая в данном модуле, обладает рядом преимуществ. Она позволяет выполнять частичную загрузку данных, а так же асинхронную (фоновую) загрузку, в процессе работы приложения. Использование JSON позволит добиться максимальной «видимой» производительности приложения-клиента, в сетях с низкой пропускной способностью. Но существует, так же проблема связанная с тем, что по

технологии JSON данные пересылаются в открытом виде, и не шифруются. Решением может стать, использование защищенного протокола (https) и/или использование дополнительного шифрования/дешифрования на сервере и клиентских приложениях.

3. Модуль кэширования данных на клиентской стороне необходим для уменьшения нагрузки на сеть, а так же для облегчения повторного доступа к данным. Поскольку в данный момент, существует проблема с доступом к жесткому диску из браузера на прямую стоит обратить внимание на функционал File API. При помощи данной API веб-приложение может создавать, читать, просматривать и записывать файлы находящиеся в области пользовательской «песочницы», размер которой может указать сам пользователь.

4. Подсистема подготовки данных к отображению выполняет подготовку изображений к отображению на пользовательском интерфейсе. В случае двумерного изображения это формирование картинки для отображения на HTML5 Canvas. В случае работы с трехмерными изображениями используется функционал библиотеки WebGL, для работы с 3D-графикой. Данная библиотека использует низкоуровневые средства поддержки OpenGL, которые позволяют ей выполнять расчеты на видеоадаптере, что в значительной мере увеличит производительность при построении трехмерных изображений. Сформированные трехмерные сцены проецируются на плоскость и отображаются на HTML5 Canvas.

5. Подсистема контроля доступа реализуется на основе избирательного управления доступом, то есть создание соответствий пользователей и ролей, с предоставлением каждой роли тех или иных данных.

Выводы. Произведен анализ процесса удаленного доступа к медицинским данным в PACS. Определены основные проблемы связанные с удаленным доступом к данным и разработана функциональная структура веб-ориентированной подсистемы авторизованного доступа к DICOM-данным. Так же предложено ряд технологических решений, по построению такой системы и внедрению её в архитектуру PACS.

Список литературы

1. DICOM [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2015]. URL: <https://ru.wikipedia.org/wiki/DICOM>
2. PACS [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2016]. URL: <https://ru.wikipedia.org/wiki/PACS>

УДК 004.41

**Разработка веб ориентированной подсистемы планирования и учёта
индивидуальной нагрузки преподавателя ВУЗа
Васильев В.В., Привалов М.В., Пряхин В.В.**

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
E-mail: vavlvl@mail.ru

Аннотация

Васильев В.В., Привалов М.В., Пряхин В.В. Разработка веб ориентированной подсистемы планирования и учёта индивидуальной нагрузки преподавателя ВУЗа. Выполнена постановка задачи для создания подсистемы планирования и учёта индивидуальной нагрузки преподавателя. Разработана функциональная структура подсистемы, обеспечивающей информатизацию данной предметной области.

Ключевые слова: индивидуальная нагрузка преподавателя, индивидуальный план, журнал учёта работы преподавателя, функциональная структура подсистемы.

Постановка проблемы. В силу различных причин уровень информатизации в высших учебных заведениях нередко оказывается не на самом высоком уровне. В связи с чем анализ, проектирование и разработка подсистемы для планирования и учёта индивидуальной нагрузки преподавателя ВУЗа является актуальной задачей. Подобная информационная подсистема позволит уменьшить временные затраты преподавателей на формирование и ведение индивидуальных планов и журналов учёта работы, а также обеспечит оперативный доступ к ним.

Цель статьи – провести анализ процесса планирования и учёта индивидуальной нагрузки преподавателя ВУЗа и разработать функциональную структуру подсистемы для автоматизации процессов формирования и ведения индивидуальных планов и журналов учёта работы преподавателя.

Описание работы существующей информационной системы. Индивидуальный план является основным документом, определяющим объем и содержание работы профессорско-преподавательского состава. Индивидуальные планы рассматриваются и утверждаются на заседании кафедры [1].

Индивидуальный план (планы) преподавателя составляется на основании документа УП4, который содержит информацию о распределении учебной работы преподавателей кафедры в часах на весь учебный год. В течение года в индивидуальный план могут быть внесены изменения на основе решения кафедры с согласия преподавателя.

По окончании каждого учебного семестра преподавателем заполняется графа выполнения учебной нагрузки за данный семестр. По окончании учебного года заполняются графы «Выполнение» по каждому из разделов индивидуального плана работы, и преподаватель отчитывается по всем видам деятельности на заседании кафедры.

Журнал учёта работы преподавателя является основным документом учета часов и ведется в течение учебного года. В журнале ведётся подробный учёт выполнения учебной работы преподавателя, который содержит информацию о каждом проведённом занятии, с указанием даты, темы, количества прочитанных часов и шифра академической группы, для которой было проведено занятие.

В конце журнала содержится отчёт о выполнении плана учебной работы, который содержит итоговую информацию о проведенных занятиях за каждый месяц, за семестр и за весь учебный год.

Разработка функциональной структуры подсистемы.

Подсистема должна реализовывать следующие функции:

- формирование отчётности: как индивидуальной, так и по кафедре;
- учёт выполнения учебной работы;
- формирование индивидуального плана;
- утверждение индивидуального плана;
- корректировка индивидуального плана.
- Входными документами подсистемы являются:
- форма УП4, содержащая информацию об учебной нагрузке преподавателей кафедры на учебный год с разделением на семестры;
- расписание занятий;
- график защит дипломных работ;
- служебные записки, на основании которых заведующий кафедрой вносит изменения в индивидуальный план преподавателя.
- Выходными документами подсистемы являются:
- индивидуальные планы и информация о их фактическом выполнении;
- журналы учёта работы преподавателей;
- отчётность по кафедре.

Веб-ориентированность подсистемы обусловлена тем, что такой подход обеспечивает независимость пользователей от операционной системы, а единственным требованием к программному обеспечению пользователя является наличие современного веб-браузера [2].

Исходя из анализа работы существующей информационной системы в подсистеме были выделены следующие функции:

- формирование индивидуального плана – выполняется каждым преподавателем не позже чем за месяц до начала действия индивидуального плана. Входной информацией для формирования плана являются данные формы

УП4, содержащей нагрузку преподавателей по семестрово. Каждый сформированный план утверждается заведующим кафедрой. Следует отметить, что данная функция выполняет оптимальное разбиение имеющейся учебной нагрузки преподавателя на два индивидуальных плана в случае, когда нагрузка превышает максимально допустимый размер;

- корректировка индивидуального плана – данная функция используется для внесения изменений в нагрузку преподавателя; осуществляется заведующим кафедрой на основании служебной записки преподавателя;

- формирование журнала учёта индивидуальной работы преподавателя – создаёт пустой электронный журнал. Входной информацией для данной функции является расписание занятий, которое в дальнейшем используется для автоматизированного заполнения журнала;

- текущее ведение журнала учёта работы преподавателя – выполняется автоматизировано на протяжении всего учебного года. Преподаватель осуществляет контролирующую роль, а также вносит информацию об изменениях в работе отклоняющихся от расписания;

- заполнение фактического выполнения индивидуального плана (-ов) – данная функция осуществляет подсчёт итогового количества часов, вычитанных преподавателем в каждом месяце, семестре и учебном году;

- формирование индивидуальной отчётности;

- формирование отчётности по кафедре.

Описание функции «Формирование индивидуального плана».

Данная функция является одной из основных функций разрабатываемой подсистемы. Входным документом является форма УП4, которая содержит информацию об учебной нагрузке преподавателей кафедры на учебный год с разделением на семестры. Результатом работы функции является индивидуальный план(-ы) преподавателя. Управляющими воздействиями на данную функцию являются закон об образовании, кодекс законов о труде и внутренние распоряжения ВУЗа, которые регулируют учебную нагрузку преподавателей, максимально допустимое превышение нормы часов и т.п. Декомпозиция функции представлена на рис. 2.

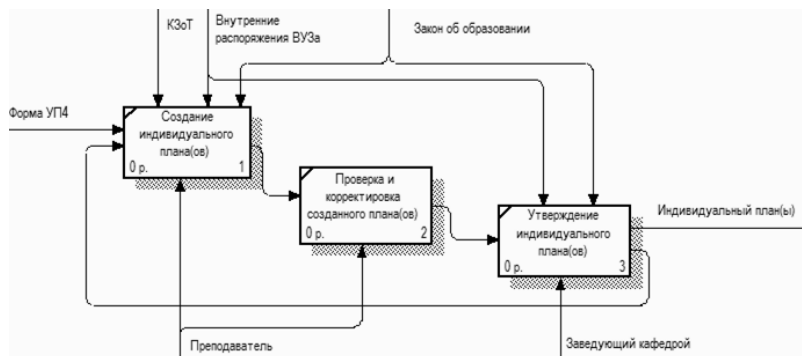


Рис. 2 – Декомпозиция функции формирования индивидуального плана

Формирование индивидуальных планов происходит следующим образом: преподаватель импортирует данные об учебной нагрузке из формы УП4, исходя из которых, подсистема выполняет формирование индивидуального плана либо оптимальным образом выполняет разбиение нагрузки на два индивидуальных плана, в случае если имеющаяся нагрузка превышает установленную норму нагрузки на ставку. Данные действия выполняются в соответствии с установленными ограничениями, такими как норма нагрузки и максимально допустимое превышение нормы. Преподаватель проверяет сформированный план(-ы) и вносит корректировки в случае необходимости. После отправляется уведомление заведующему кафедрой о том, что существуют индивидуальные планы требующие утверждения. Результат утверждения отправляется преподавателю в виде уведомления. В случае успешного утверждения индивидуальный план(-ы) распечатывается в двух экземплярах, а также сохраняется в электронной базе данных и изменению не подлежит.

Описание функции «Текущее ведение журнала учёта работы преподавателя». Назначение данной функции состоит в ведении журнала учёта работы преподавателя. Согласно установленному регламенту текущее ведение журнала учёта работы преподавателя выполняется ежедневно. В журнал вносятся записи о проведённых занятиях с указанием даты, группы, учебной дисциплины, темы и вида занятия, а также количества прочитанных часов.

Входным документом также, как и выходным является журнал учёта работы преподавателя, частью которого является расписание занятий преподавателя.

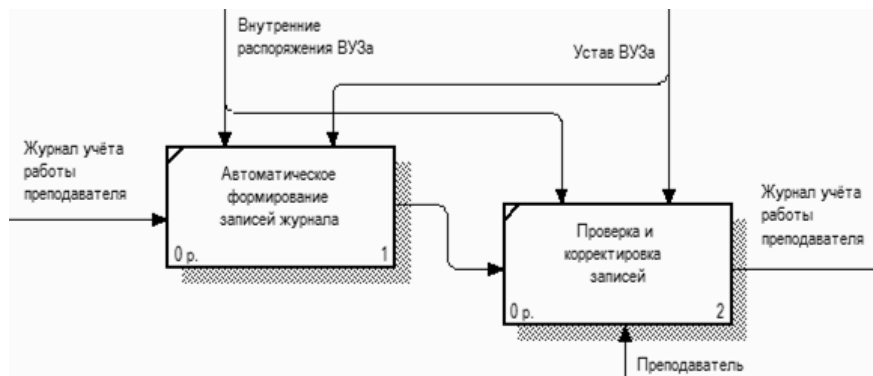


Рис. 3 – Декомпозиция функции текущего ведения журнала учёта работы преподавателя

При вызове данной функции подсистема генерирует записи журнала на основании имеющегося расписания занятий преподавателя. Происходит это следующим образом: выполняется поиск последней записи в журнале, и со следующего по текущий рабочий день выполняется генерация записей. Далее преподаватель выполняет проверку сформированных записей и вносит правки в случае необходимости.

Выводы. Проведён анализ процесса планирования и учёта индивидуальной нагрузки преподавателя ВУЗа и разработана функциональная структура подсистемы для автоматизации процессов формирования и ведения индивидуальных планов и журналов учёта работы преподавателя.

Список литературы

1. Инструкция по оформлению индивидуального плана работы преподавателя государственного образовательного учреждения высшего профессионального образования «Донецкий Национальный Технический Университет».
2. Д.С. Негурица. Модель пользователя в адаптивных интерфейсах программного обеспечения Web-ориентированных систем. / Д.С. Негурица, Е.В. Соколова // *Радиоелектронні І Комп'ютерні Системи*. – 2014. – №1. – С. 104-111.

Секция № 5
Математическое
и компьютерное
моделирование



УДК 658.5.011

Имитационное моделирование в процессе проектирования станков и станочных систем

Москалюк Т.Г., Харитонов А.Ю.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерных систем мониторинга
E-mail: tarasusthelight@gmail.com

Аннотация

Москалюк Т.Г., Харитонов А.Ю. Имитационное моделирование в процессе проектирования станков и станочных систем. Выполнен анализ основных концепций имитационного моделирования станков и станочных комплексов. Разработан программный продукт, реализующий расчёт производительности промышленного комплекса для его оптимизации.

***Ключевые слова:** имитационное моделирование, проектирование программного продукта, NILabVIEW.*

Постановка проблемы. В процессе проектирования промышленных комплексов прямая разработка реальных систем слишком 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. Имитационное моделирование – предварительный этап создания любой промышленной системы. Такое моделирование используют в начале процесса проектирования для выбора компоновки и формирования требований к отдельным составляющим проектируемого станка или станочной системы, с точки зрения обеспечения заданной точности, надёжности и производительности в условиях действия дестабилизирующих факторов.

Постановка задачи. Цель данного исследования — разработка программного продукта — имитационной модели конкретного промышленного комплекса. С её помощью будет осуществлена проверка эффективности системы, нахождение слабых или неэффективных узлов и их устранение, а также её модернизация и оптимизация под современные нужды. Задача программного продукта – определить производительность комплекса.

При построении имитационных моделей используют структурный подход, согласно которому сначала обеспечивается программная реализация моделей функциональных блоков исследуемого объекта, а затем объединение полученных программных модулей с учётом взаимодействия и связей между функциональными блоками в реальном объекте.

Соответственно, имитационная модель должна с максимальной точностью воспроизводить реальные процессы, чтобы между моделируемым объектом, и его реальным воплощением была минимальная разница. В основном под имитационным моделированием подразумевают методику воспроизведения объектов и процессов, параметры которых меняются случайным образом.

Математической основой имитационного моделирования является метод статистических испытаний (метод Монте-Карло) [3]. Предварительно объект должен быть представлен в виде сложной системы. Планирование и обработка результатов моделирования производится с помощью тех же методов математической статистики, что и в случае физического эксперимента. [1][3]

В данном исследовании, при проектировании программы имитационного моделирования, будет использована среда разработки NI LabVIEW.

Среда разработки NI LabVIEW, является специализированным комплексом, предназначенным для программирования промышленных устройств и станков любой сложности, что идеально подходит для имитационного моделирования имеющейся группы устройств. Среда разработки обладает широким выбором средств, упрощающих создание систем, а отсутствие текстового кода, и программирование при помощи блоков позволяют приступить к проектированию человеку, не имеющему опыта в программировании сложных систем с применением текстовых языков.

Также, NI LabVIEW предоставляет широкие возможности визуализации полученных и результирующих данных, вплоть до демонстрации самого промышленного процесса в схематическом виде. [2]

Результаты. Промышленный комплекс состоит из трёх однотипных станков, двух накопителей, и промышленного робота. Промышленный робот берёт заготовки из первого накопителя и устанавливает их в станки. Станки обрабатывают деталь, по завершению цикла робот извлекает готовые детали из станков и переносит их во второй накопитель, откуда они движутся дальше по производству.

Задача программного продукта – определить производительность комплекса, при обработке деталей различной трудоёмкости, разным размером партий деталей, разным количестве партий. Производительность комплекса является коэффициентом, и может быть выражена в процентах. Она включает в себя единицу, делённую на сумму средней длительности рабочего цикла ($T_{ц}$) и суммарных внецикловых потерь ($\Sigma t_{п}$) (формула 1).

$$Q = \frac{1}{T_{ц} + \Sigma t_{п}}$$

Формула 1 – Производительность комплекса

В программе рассчитывается $T_{ц}$, а затем $\Sigma t_{п}$ для каждой партии деталей.

В исходные данные вводятся:

- Время безотказной работы;

- Время восстановления или замены режущего инструмента;
- Основные параметры промышленного робота:
- Скорость перемещения каретки.
- Время установки или замены заготовки.
- Параметры партии деталей:
- Число партий.
- Число деталей в каждой партии.
- Трудоёмкость обработки деталей (время обработки одной детали).

Перечисленные параметры поступают на программный модуль, имитирующий обработку деталей, их загрузку-выгрузку, и производящий вычисления производительности комплекса. Пройдя весь алгоритм, программа выводит рассчитанные значения производительности для всех партий конечному пользователю. В дальнейшем, пользователем производится статистическая обработка полученных данных, определение необходимости внесения корректив в промышленный комплекс, и подбор оптимальных параметров работы реальной системы.

Программный продукт выполнен в среде разработки NI LabVIEW. Визуальный интерфейс пользователя предельно прост и интуитивно понятен, что позволит использовать программный продукт продолжительное время с максимальной эффективностью (рис. 1).

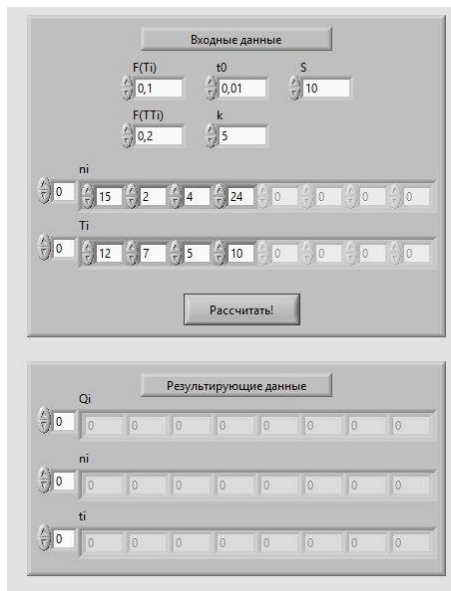


Рис. 1 – Интерфейс программного продукта

Выводы. Данное исследование несёт значительную практическую пользу, а также социально-экономический эффект в пределах предприятия использующего программный продукт для модернизации оборудования. Программный продукт снизит финансовые затраты на модернизацию и оптимизацию промышленного комплекса, а также затраты времени на тестирование внесённых изменений.

Список литературы

1. Васильев Г.Н. «Автоматизация проектирования металлорежущих станков» — М.: Машиностроение. 1987. — 280с., ил.
2. LabVIEW - Руководство Пользователя – 2007 Перевод: Николаев С.В. ftp://ftp.ni.com/pub/branches/russia/software/labview_user_manual.pdf
3. Соболев И.М. Численные методы Монте-Карло. – М.: Наука, 1973

УДК 620.1:004.94

Использование метода векторных конечных элементов для моделирования магнитных полей в неразрушающем контроле

Якимов А.Н., Мартыненко Н.В.

Луганский государственный медицинский университет

E-mail: alex_forever_82@mail.ru

Аннотация

Якимов А.Н., Мартыненко Н.В. Использование метода векторных конечных элементов для моделирования магнитных полей в неразрушающем контроле. В статье рассмотрено применение метода векторных конечных элементов для моделирования процесса намагничивания ферромагнитных объектов контроля и расчета создаваемых ими магнитных полей, а также приведены примеры реализации данного метода с использованием современных компьютерных технологий.

Введение. В практике магнитного неразрушающего контроля большое внимание уделяется вопросам поиска дефектов, определения состава и других физических свойств изделия по характеру их влияния на приложенное магнитное поле. При рассмотрении этих задач оказывается полезным использование численных методов анализа, позволяющих моделировать данный процесс с помощью компьютера.

К настоящему времени разработаны коммерческие программные продукты, такие как Elcut, ANSYS Maxwell, Comsol Multiphysics и др., предоставляющие возможность проводить расчеты для различных приложений

науки и техники, в том числе и в области магнитных полей. Однако большинство таких программ реализовано на базе точечного метода конечных элементов, который дает возможность вычислять только скалярные поля, и не работает напрямую с векторными величинами [1-3]. Для их описания применяется отдельная аппроксимация каждой из компонент вектора с использованием функций формы. При этом автоматически не выполняются граничные условия для векторного поля в смежных элементах дискретизации и значения его компонент не всегда оказываются согласованными друг с другом. Особенно это критично на границе раздела материалов с различными физическими свойствами, где использование данного подхода приводит к дополнительным погрешностям вычислений, а также необходимости введения дополнительных граничных условий и видоизменения исходных уравнений.

В связи с этим, для решения задач электромагнетизма перспективнее использовать векторный метод конечных элементов VFEM (Vector Finite Element Method) [4, 5], который позволяет работать напрямую с векторными характеристиками магнитного поля, и лишен описанных выше недостатков точечного метода конечных элементов.

Взаимодействие постоянного магнитного поля с объектом контроля описывается с использованием уравнения

$$\nabla \times (\nu \cdot \nabla \times \vec{A}) = \vec{J}_S, \quad (1)$$

где \vec{A} - векторный магнитный потенциал, \vec{J}_S - плотность тока источников магнитного поля, $\nu = 1/(\mu\mu_0)$ - магнитная проводимость вещества, μ - его относительная магнитная проницаемость. Данное уравнение должно быть дополнено соответствующими граничными условиями.

Для решения поставленной задачи методом векторных конечных элементов расчетная область разбивается на множество тетраэдральных элементов, позволяющих описывать произвольную геометрию объекта контроля. Представим решение уравнения (1) в виде

$$\vec{A} = \sum_{e=1}^M \sum_{j=1}^6 \vec{N}_j^e \cdot A_j^e, \quad (2)$$

в котором \vec{N}_j^e - векторная базисная функция, ассоциированная с ребром j тетраэдра e ; A_j^e - величина проекции векторного потенциала на его j -е ребро; M - количество используемых при дискретизации элементов.

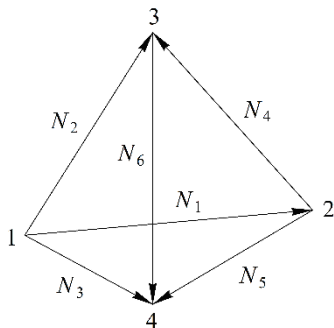


Рис. 1 - Тетраэдральный элемент разбиения с векторными базисными функциями, ассоциированными с его ребрами.

Для построения векторных базисных функций рассмотрим отдельный тетраэдр (см. рис. 1). Его ребру j ставится в соответствие векторная функция

$$\vec{N}_j = (\xi_m \nabla \xi_n - \xi_n \nabla \xi_m) \cdot L_{mn}, \quad (3)$$

где $L_{mn} = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2 + (z_n - z_m)^2}$ - длина ребра тетраэдра, с которым ассоциирована базисная функция; ξ_m и ξ_n - линейные базисные функции начальной и конечной точек ребра ($m < n$). Базисные функции ξ , называемые еще симплексными координатами, определяются из соотношения [1]

$$\xi_i = V_i / V, \quad i = \overline{1, 4}, \quad (4)$$

где V - объем тетраэдра, а V_i - объем вспомогательного тетраэдра, построенного на трех его вершинах, противоположных вершине i , и четвертой точке $P(x, y, z)$, лежащей внутри тетраэдра. С другой стороны,

$$\xi_i = \frac{a_i + b_i x + c_i y + d_i z}{6V}, \quad i = \overline{1, 4}. \quad (5)$$

где a_i, b_i, c_i, d_i - алгебраические дополнения к элементам определителя

$$\begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}, \quad (6)$$

из i -й строки и соответственно первого, второго, третьего или четвертого столбца. Из определения (4) следует, что скалярная функция ξ_i равна единице у

i -й вершине тетраэдра и обращается в ноль на грани, противоположной этой вершине.

Векторная базисная функция берется со знаком «+», если в глобальной нумерации вершин направляющий вектор ребра идет от точки с меньшим номером к точке с большим номером, в противном случае его векторная базисная функция берется со знаком «-».

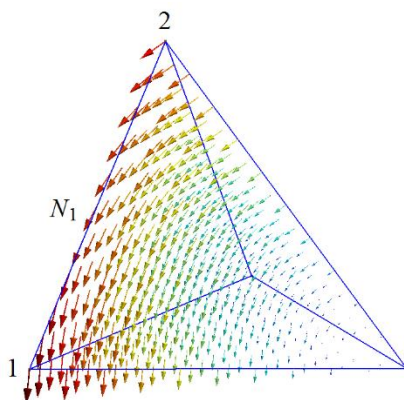


Рис. 2 - Графическое представление одной из векторных базисных функций.

На рис. 2 показан внешний вид векторной функции для одного из ребер тетраэдра. Из соотношения (3) следует, что любая базисная функция \vec{N}_j имеет единичную проекцию на ребро j и перпендикулярна к остальным ребрам. Вне тетраэдра все его векторные базисные функции полагаются равными нулю. Кроме того, можно показать [5], что $\nabla \cdot \vec{N}_j = 0$, а $\nabla \times \vec{N}_j \neq 0$, поэтому векторные функции (3) еще называют роторными функциями. Использование данных функций обеспечивает автоматическое выполнение равенства тангенциальных составляющих напряженности и векторного потенциала магнитного поля на границах элементов дискретизации:

$$A_{i1} = A_{i2}, \quad H_{i1} = H_{i2}. \quad (7)$$

Для простоты будем считать, что на внешней границе S расчетной области для векторного потенциала магнитного поля задано условие Дирихле

$$\vec{A}^D = \vec{P}(x, y, z). \quad (8)$$

Например, вдали от источников поля можно приближенно считать векторный потенциал равным нулю. На практике обычно расчетную область в

этом случае берут в 3-5 больше, чем объект контроля. Тогда формула (2) записывается в виде

$$\vec{A} = \sum_{j=1}^{n_F} A_j \cdot \vec{N}_j + \sum_{j=1}^{n_D} A_j^D \cdot \vec{N}_j^D \quad (9)$$

где n_F , n_D - число «свободных» ребер и количество ребер, лежащих на границе расчетной области; \vec{N}_j , A_j и \vec{N}_j^D , A_j^D - соответствующие им базисные функции и тангенциальные составляющая векторного магнитного потенциала.

Кроме условия Дирихле на практике часто используется граничное условие Неймана, которое задается на плоскостях симметрии или антисимметрии и позволяет существенно уменьшить размеры расчетной области и понизить размерность решаемой задачи.

Воспользуемся методом Галеркина [1, 5] и запишем уравнение (1) в виде так называемой «слабой формы» (weak-form). Для этой цели скалярно умножим его обе части на векторные базисные функции, соответствующие i -му ребру тетраэдра, и проинтегрируем по всему объему V расчетной области:

$$\int_V \vec{N}_i \cdot \nabla \times (\nu \cdot \nabla \times \vec{A}) dV = \int_V \vec{N}_i \cdot \vec{J}_S dV. \quad (10)$$

Далее, беря в учет соотношение (9) и применяя соответствующие формулы и теоремы векторного анализа, приходим к системе алгебраических уравнений вида

$$\mathbf{K} \cdot \mathbf{A} = \mathbf{b}, \quad (11)$$

где коэффициенты матрицы и правой части системы могут быть найдены по формулам

$$K_{ij} = \int_V \nu \cdot (\nabla \times \vec{N}_i) \cdot (\nabla \times \vec{N}_j) dV, \quad (12)$$

$$b_i = \int_V \vec{N}_i \cdot \vec{J}_S dV - \int_V \nu \cdot (\nabla \times \vec{N}_i) \cdot (\nabla \times \vec{N}_j^D) dV \quad (13)$$

аналитически или с помощью численного интегрирования ($i, j = \overline{1, n_F}$) [2, 5].

Решая систему уравнений (11) получаем неизвестные значения тангенциальных составляющих векторного потенциала \vec{A} на ребрах тетраэдров, на которые дискретизируется объект контроля и окружающий его воздух. Далее, магнитная индукция и напряженность магнитного поля в случае линейных магнитных свойств ферромагнетика могут быть определены по формулам:

$$\vec{B} = \nabla \times \vec{A}^e = \sum_{j=1}^6 A_j^e \cdot (\nabla \times \vec{N}_j^e) \quad (14)$$

$$\vec{H} = \nu \cdot (\nabla \times \vec{A}^e) = \sum_{j=1}^6 \nu \cdot A_j^e \cdot (\nabla \times \vec{N}_j^e), \quad (15)$$

где e - элемент разбиения, в котором лежит точка наблюдения, а \vec{N}_j^e - соответствующие ему векторные базисные функции.

Данные математические модели были верифицированы для случаев намагничивания ферромагнитных тел, для которых известно точное аналитическое решение. На рис.3 показано полученное распределение магнитного поля внутри тонкой стальной пластинки, которая находится в однородном магнитном поле с напряженностью $H_z = 1$ кА/м, направленном перпендикулярно к ее поверхности. Размеры пластинки составляли $10 \times 10 \times 1$ см, а ее магнитная проницаемость полагалась равной $\mu = 100$. Расчетная область имела форму куба со стороной 20 см. Для обеспечения однородности приложенного к пластинке поля векторный потенциал на границе расчетной области полагался равным $\vec{A} = (0, \mu_0 H_z x, 0)$.

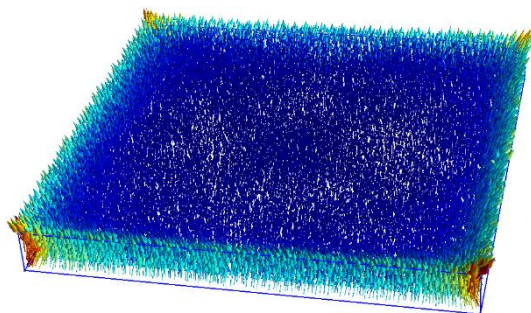


Рис. 3 - Моделирование намагничивания тонкой стальной пластинки, помещенной в однородное магнитное поле.

Ферромагнитная пластинка разбивалась на тетраэдрические конечные элементы с шагом 2.5 мм, а воздух вблизи внешней границы расчетной области дискретизировался с более крупным шагом 1 см. Разбиение от внешней границы до поверхности пластинки постепенно сгущалось. Весь объем включал в себя около 300 тыс. элементов дискретизации.

Как известно, напряженность магнитного поля внутри тонкой пластинки, помещенной в однородное магнитное поле, определяется из соотношения

$$H = H_z / \mu. \quad (16)$$

Численный эксперимент дает значение в центре тонкой пластине, равное 9.7 А/м. Таким образом, погрешность расчета находится на уровне 3 %.

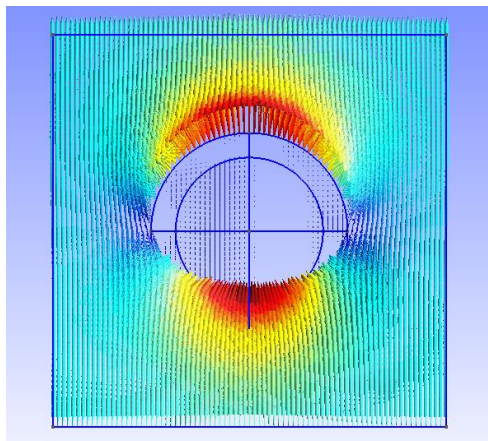


Рис. 4 - Анализ распределения магнитного поля снаружи и внутри сферического экрана, помещенного в однородное магнитное поле.

В качестве второго примера выбран тонкий сферический экран, находящийся в однородном магнитном поле с напряженностью 1 кА/м. Его внешний и внутренний радиусы составляли соответственно $R = 100$ и $r = 75$ мм, а относительная магнитная проницаемость материала экрана $\mu = 50$. Расчетная область имела форму куба со стороной 300 мм и включала около 1 миллиона тетраэдров. Для величины поля внутри экрана существует аналитическое соотношение

$$\vec{H} = \frac{\vec{H}_0}{1 + \frac{2}{9} \left(1 - \frac{r^3}{R^3}\right) \left(\frac{1}{\chi + 1} + \chi - 1\right)}, \quad (17)$$

где \vec{H}_0 - напряженность внешнего магнитного поля; $\chi = \mu - 1$ - магнитная восприимчивость ферромагнетика, из которого изготовлен экран. Кроме того, поле внутри экрана также будет однородным. Численные модели дают значение напряженности поля, равное 131 А/м, что отличается на 5.7 % от значения, полученного по формуле (17). Отклонение внутреннего поля от однородности находится на уровне 1.1 %.

Аналогичным образом математическая модель проверялась для задач намагничивания длинного ферромагнитного цилиндра, шара, эллипсоида вращения и др. Во всех случаях погрешность расчета магнитного поля не превышает 5-6 %, что свидетельствует об адекватности применяемых моделей.

В качестве последнего примера покажем результаты численного расчета магнитного поля вблизи ферромагнитной пластинки, содержащей

прямоугольный поверхностный дефект. Внешнее магнитное поле, как и в предыдущих примерах, имеет напряженность 1 кА/м и направлено вдоль пластинки перпендикулярно дефекту. Размеры пластинки составляют соответственно $10 \times 10 \times 5$ см. Ширина дефекта 0.5 мм, его длина 3 мм, а глубина 2 мм. Шаг дискретизации вблизи дефекта равен 0.1 мм, а на границах расчетной области доходит до 2 мм. Всего применяется около 500 тысяч элементов разбиения. На рис. 7 проиллюстрировано вычисленное магнитное поле рассеяния, появляющееся над дефектом.

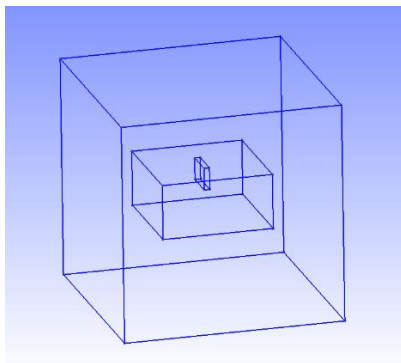


Рис. 5 - Внешний вид расчетной области, содержащей пластинку с прямоугольным поверхностным дефектом.

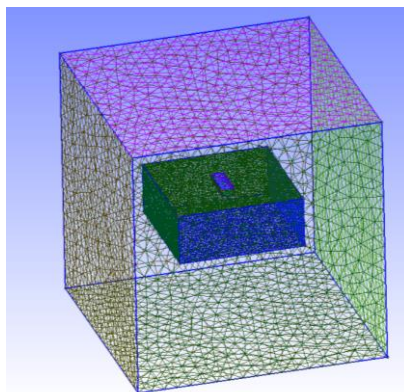


Рис.6 - Дискретизация границ расчетной области.

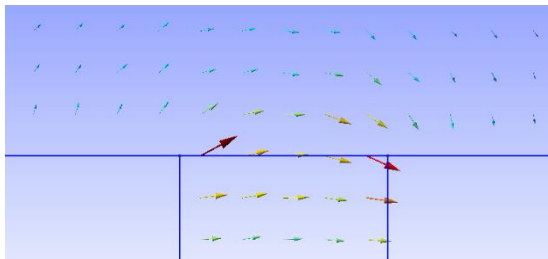


Рис. 7 - Поле рассеяния вблизи прямоугольного дефекта.

В результате можно сделать вывод, что применяемые математические модели позволяют с достаточной степенью точности вычислять пространственное распределение магнитного поля, получаемое при намагничивании ферромагнитных тел различной формы при учете их линейных магнитных характеристик, а также находить поле вблизи дефектов.

Дальнейшей целью исследований является усложнение применяемых математических моделей с целью возможности учета нелинейных магнитных свойств ферромагнетика и возникающих в нем гистерезисных явлений при перемагничивании объекта контроля. Для повышения точности численного расчета планируется использование векторных конечных элементов более высокого порядка.

Список литературы

1. Сильвестер П., Феррари Р. Метод конечных элементов для радиоинженеров и инженеров-электриков: Пер. с англ. - М.: Мир, 1986. - 229 с.
2. Ida N., Bastos J. Electromagnetics and Calculation of Fields, 2nd Edition. - N.J.: Springer-Verlag, 1997. - 537 p.
3. Sadiku M. Numerical Techniques in Electromagnetics, 2nd Edition. - CRC Press, 2001. - 750 p.
4. Баландин М.Ю., Шурина Э.П. Векторный метод конечных элементов: Учебное пособие. - Новосибирск: Изд-во НГТУ, 2001. - 69 с.
5. Jan-Ming J. The Finite Element Method in Electromagnetics, 3rd Edition. - Wiley-IEEE Press. - 2014. - 876 p.

УДК 514.18

**Сравнение двух способов решения метрических задач средствами
графического редактора AutoCAD
Карабчевский В.В., Кудимов М.И.**

Донецкий национальный технический университет
кафедра компьютерного моделирования и дизайна
E-mail: kudimov96@mail.ua

Аннотация

Карабчевский В.В., Кудимов М.И. Сравнение двух способов решения метрических задач средствами графического редактора AutoCAD. Произведено сравнение двух способов решения метрических задач. Найдены преимущества и недостатки двух способов решения метрических задач. Проанализированы полученные данные. Приведены примеры типовых метрических задач. Сделаны соответствующие выводы.

Ключевые слова: компьютерная графика, геометрическое моделирование, геометрические фигуры.

Общая постановка проблемы. Метрическими принято считать задачи, в условии или в решении которых присутствует численная характеристика. К метрическим задачам относятся задачи на построение изображений фигур по их размерам или координатам из точек, измерение расстояний, углов, площадей и другие [1].

Выделяют два основных способа решения метрических задач. Первый – это решение в пространстве, а второй – это решение на комплексном чертеже.

Считается, что первый способ более удобный и практичный в использовании чем второй.

В данной статье будет приведено сравнение этих способов на примерах решения типичных метрических задач с целью подтвердить или опровергнуть данное утверждение.

Исследования. Для сравнения был выбран ряд некоторых параметров. Первый из них это наглядность.

Примерами будут служить две задачи. Первая из них это задача на нахождения сечения цилиндра. На рисунке 1 изображено решение задачи в пространстве, а на рисунке 2 решение на комплексном чертеже. Построение в пространстве обладает большей наглядностью. На нем есть возможность увидеть построение сразу в трех проекциях (спереди, сверху и с боку), в то время как на комплексном чертеже можно увидеть только две проекции (сбоку и сверху). Так же на чертеже в пространстве можно наглядно наблюдать за тем, как именно секущая плоскость проходит через цилиндр и образует его сечение.

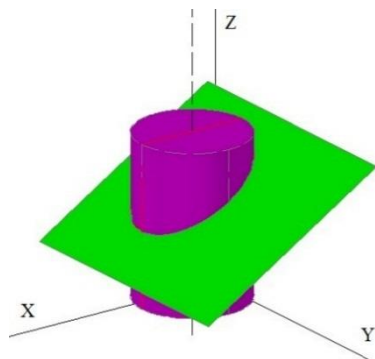


Рис. 1. Построение в пространстве

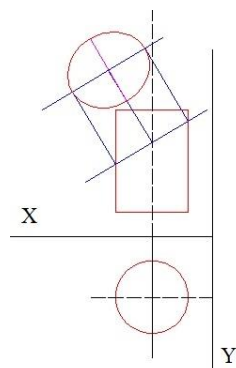


Рис. 2. Комплексный чертёж

Вторая задача, это задача на нахождение точек пересечения прямой и поверхности конуса. На рисунке 3 изображено решение на комплексном чертеже, а на рисунке 4 решение в пространстве. На построении в пространстве мы можем четко увидеть искомые точки пересечения линии и поверхности конуса. А на комплексном чертеже это не так хорошо видно.

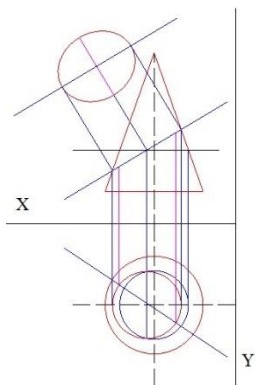


Рис. 3. Комплексный чертёж

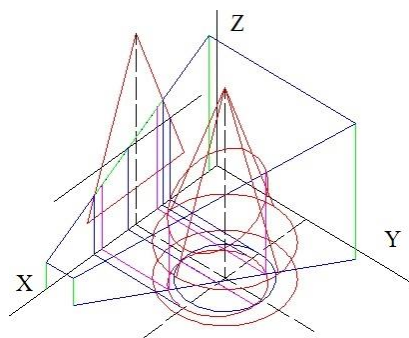


Рис. 4. Построение в пространстве

Для улучшения наглядности пространственного построения можно переключить AutoCAD из режима «2D-каркас» в режим «реалистичный». Получим следующее изображение (рис. 5). Теперь мы можем разглядеть найденные нами точки еще лучше.

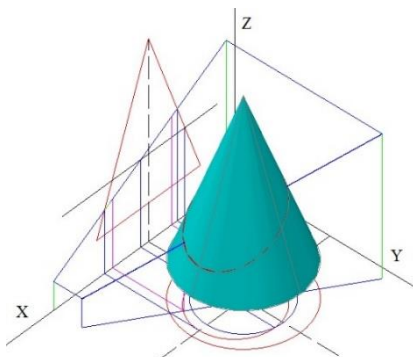


Рис. 5. Вид в режиме «реалистичный»

Помощь в поиске ошибок на комплексном чертеже

Это отдельное преимущество, которое было выделено для пространственного метода. Примером будет служить задача на отыскание перпендикуляра из точки на плоскость. На рис. 6 изображено решение задачи на комплексном чертеже и в нем допущена ошибка, но на данном рисунке она еще не видна. Допустим мы не заметили эту ошибку, так как она не видна, и продолжили построение. Вызываем команду `_ROTATE3D` и выбираем объекты, которые будем поворачивать (рис. 7).

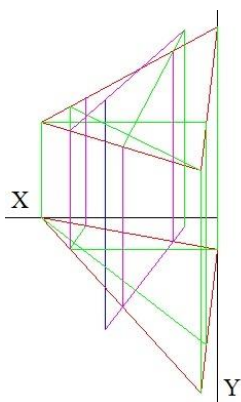


Рис. 6. Комплексный чертеж

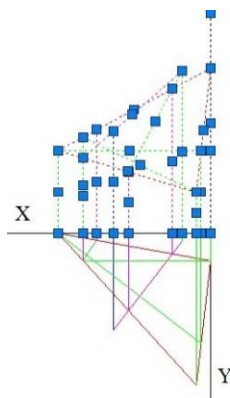


Рис. 7. Выбрали объекты для поворота

Указали осью вращения ось X и указали угол поворота -90. Нажимаем ENTER. Получили следующее построение (рис. 8).

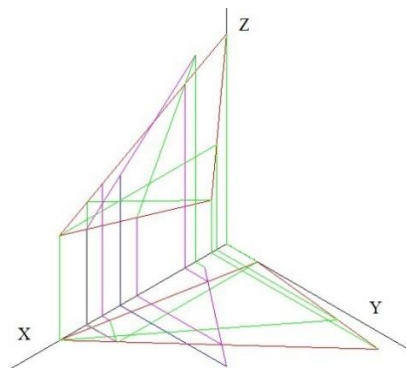


Рис. 8. Результат поворота

Ошибку все еще не видно. Поэтому продолжаем построение и когда мы дойдем до построения искомого перпендикуляра, то построить его не получится из-за той самой ошибки в начале.

В таком случае нужно проверять правильность построения перпендикуляров из вершин треугольников к оси X. И вот тут ошибка будет найдена (рис. 9).

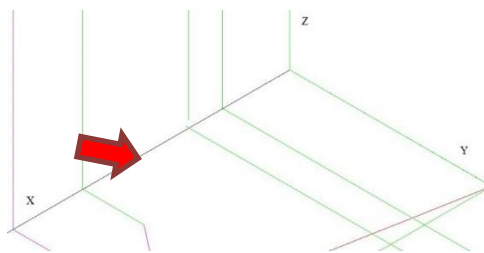


Рис. 9. Допущенная ошибка теперь найдена

Она заключалась в том, что на комплексном чертеже при построении перпендикуляров из вершин треугольников к оси X была включена объектная привязка, которая привязала конец одного из перпендикуляров не к оси X, а к началу другого перпендикуляра. А этот «другой» перпендикуляр, в свою очередь, тоже был построен не к оси X, а немного выше, но уже из-за шаговой привязки по сетке. Шаг привязки был задан маленький. Подобные ошибки появляются если проводить построение вручную.

Простота в построении. Обычно, построение в пространстве менее трудоемкое и занимает меньше времени чем построение на комплексном чертеже. Примером может служить задача на нахождение двугранного угла. На рисунке 10 изображено решение задачи на комплексном чертеже, а на рисунке 11 изображено решение в пространстве.

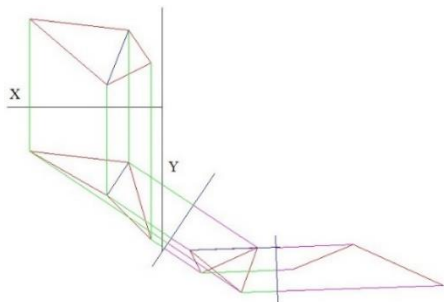


Рис. 10. Комплексный чертеж

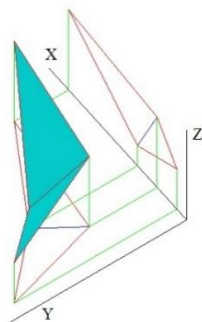


Рис. 11. Построение в пространстве

Построение в пространстве происходит за меньшее количество шагов и гораздо быстрее за счет этого и проще.

Так же построение в пространстве обладает большей наглядностью, за счет чего мы можем лучше увидеть где именно находится и как выглядит этот двугранный угол между двумя треугольниками с общей стороной (гранью). На комплексном чертеже это тоже видно, но сам угол там строится в стороне от треугольников и поэтому менее показательный.

Выводы. Проанализировав приведенное выше сравнение, приходим к выводу, что да, действительно, первый способ более удобный и практичный в использовании чем второй. Он менее трудоемкий в построении и более наглядный [2]. С его помощью можно наблюдать как именно секущая плоскость проходит через объект (рис. 1), как выглядит двугранный угол в пространстве (рис. 11) и многое другое.

Конечно, первый способ не идеальный и у него тоже есть свои минусы, но они без труда перекрываются его плюсами. Так же бывают случаи, когда этим способом воспользоваться не получается. Для таких случаев применяется второй способ [3].

А способ решения метрических задач на комплексном чертеже нуждается в усовершенствовании для повышения наглядности. Конечно, бывают случаи,

когда комплексный чертеж выглядит более наглядным и более удобным в использовании, но это все частные случаи.

Список литературы

1. Метрические задачи и способы их решения [электронный ресурс] // Курс лекций по начертательной геометрии [сайт]. [2012]. URL: <http://allrefs.net/c23/4dzt/p16/>
2. Карабчевский В.В. Методы компьютерной геометрии Донецк: ГБУЗ «ДонНТУ», Технопарк ДонНТУ «УНИТЕХ», 2010. – 179 с.
3. Карабчевский В.В. Трехмерное моделирование и начертательная геометрия в курсе «Методы компьютерной геометрии» // Міжвідомчий науково-технічний збірник "Технічна естетика і дизайн". Випуск 8. - К.: КНУБА, 2011 р. – С. 138-142.

УДК 514.18

Точечное уравнение синусоиды с осью в виде окружности и его применение при построении поверхностей технических форм методом подвижного симплекса

Давыденко И.П., Малютина Т.П.

Донбасская национальная академия строительства и архитектуры,
г. Макеевка кафедра специализированных информационных
технологий и систем

E-mail: davydenkoip@mail.ru

Аннотация

Давыденко И.П., Малютина Т.П. Точечное уравнение синусоиды с осью в виде окружности и его применение при построении поверхностей технических форм методом подвижного симплекса. В работе рассматривается точечное уравнение синусоиды с осью в виде окружности, на основании графического алгоритма ее построения, методами БН-исчисления (точечное исчисление Балюбы-Найдыша) [1]. Приведен пример точечного задания криволинейной поверхности технической формы, на основе метода подвижного симплекса (МПС) [2], с образующей в виде прямой линии и направляющей в виде синусоиды [3].

Ключевые слова: *точечное уравнение синусоиды, БН-исчисление, поверхность технической формы, метод подвижного симплекса.*

Постановка проблемы. При построении таких криволинейных поверхностей технических форм как оболочек стационарных покрытий над трибунами стадионов, в качестве направляющей поверхности принимают эллипс,

а образующей – параболу. В данной работе рассмотрим пример точечного задания криволинейной поверхности технической формы, на основе метода подвижного симплекса, с образующей в виде прямой линии и направляющей в виде синусоиды [3]. В точечном исчислении уже разработаны точечные уравнения многих кривых в различной параметризации и синусоиды в том числе [3], однако возникла необходимость в задании криволинейной поверхности технической формы, на основе метода подвижного симплекса, с образующей в виде прямой линии и направляющей в виде синусоиды, расположенной в плоскости общего положения.

Анализ последних исследований. В последних исследованиях разработаны точечные уравнения кривых второго порядка в различной параметризации [4].

Формулирование целей статьи. Цель статьи – рассмотреть точечное уравнение синусоиды, расположенной в плоскости общего положения, для последующего задания поверхностей оболочек [5].

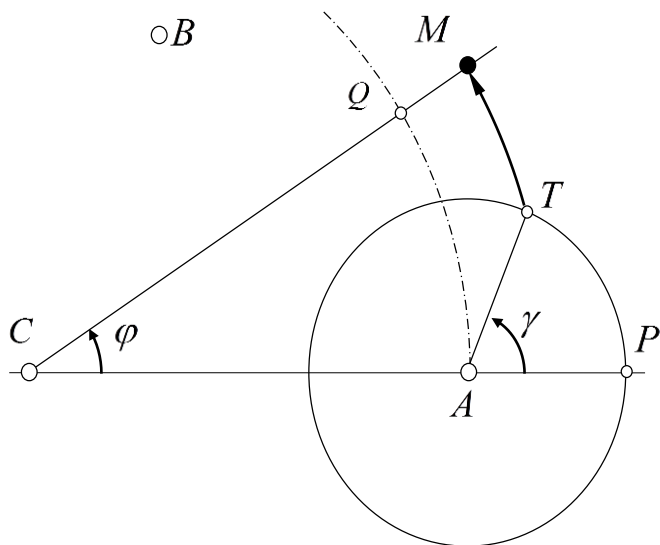


Рис. 1. Задание синусоиды с осью в виде окружности

Основная часть. Рассмотрим точечное уравнение синусоиды, осью которой является окружность [3]. Используя полярную параметризацию плоскости, зададим точечное уравнение окружности в симплексе точек ABC при $|CA| = b$ – радиус осевой окружности; $|AP| = \rho$ – радиус образующей окружности; $0 \leq \varphi \leq 2\pi$ – параметр; $|AQ| = |PT|$ (рис. 1).

Алгоритм построения:

1. A, B, C, k .

$$2. a = |BC| = \sqrt{\sum_{BB}^C} = \sqrt{\sum (B - C)^2} = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2 + (z_B - z_C)^2}.$$

$$3. b = |AC| = \sqrt{\sum_{AA}^C} = \sqrt{\sum (A - C)^2} = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2 + (z_A - z_C)^2}.$$

$$4. \sum_{AB}^C = \sum (A - C)(B - C) = \\ = (x_A - x_C)(x_B - x_C) + (y_A - y_C)(y_B - y_C) + (z_A - z_C)(z_B - z_C).$$

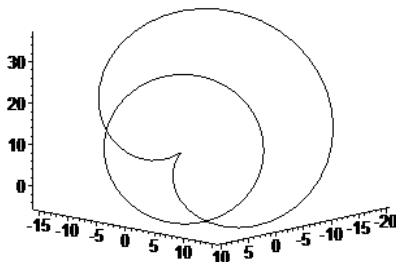
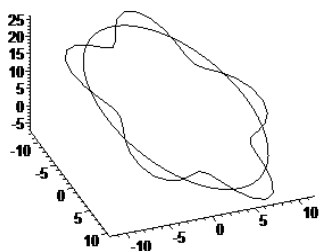
$$5. \cos \gamma = \frac{\sum_{AB}^C}{\sqrt{\sum_{AA}^C} \sqrt{\sum_{BB}^C}}.$$

$$6. 2k\pi\rho = 2\pi b \rightarrow \rho = \frac{b}{k}.$$

$$7. M = (A - C) \frac{\sin(\gamma - \varphi) \sqrt{1 + k^2 + 2k \cos k\varphi}}{k \sin \gamma} + (B - C) \frac{b \sin \varphi \sqrt{1 + k^2 + 2k \cos k\varphi}}{ak \sin \gamma} + C,$$

$$0 \leq \varphi \leq 2\pi.$$

Примеры компьютерной визуализации синусоиды приведены на рис. 2.



При $k=1$: При $k=5$:

Рис. 2. Примеры компьютерной визуализации синусоиды в *Maple*

Примеры компьютерной визуализации криволинейной поверхности технической формы, на основе метода подвижного симплекса (МПС), с образующей в виде прямой линии и направляющей в виде синусоиды, приведены на рис. 3.

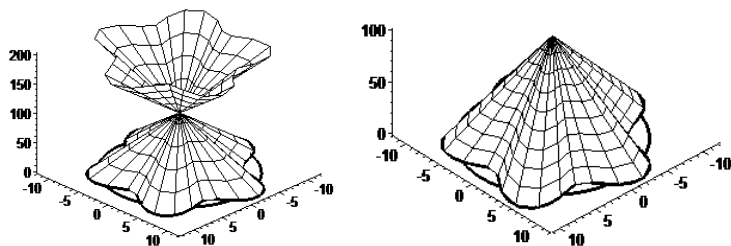


Рис. 3. Примеры компьютерной визуализации поверхности оболочки в *Maple*

Выводы. Рассмотрено точечное уравнение синусоиды, расположенной в плоскости общего положения, для последующего задания поверхности технической формы – оболочки, с образующей в виде прямой линии и направляющей в виде синусоиды.

Список литературы

1. Верещага В.М. Алгебра БН-исчисления / Балюба И.Г., Верещага В.М., Найдыш В.М. // Прикладна геометрія та інженерна графіка. Міжвідомчий науково-технічний збірник. Вип. 90. – К.: КНУБА, 2012. – С. 210-215.
2. Метод подвижного симплекса при конструировании 2-поверхностей многомерного пространства / [Балюба И.Г., Полищук В.И., Горягин Б.Ф., Малютина Т.П., Давыденко И.П. и другие] // Моделювання та інформаційні технології. Збірник наукових праць. Спец. вип. Матеріали Міжнародної наукової конференції «Моделювання – 2010», 12-14 травня 2010 р., м. Київ, Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України., Т.1. – С. 310-318.
3. Конопацький Є.В. Конструювання системи спеціальних плоских кривих типу «синусоїда» методом узагальнених тригонометричних функцій / Сборник научных трудов SWorld. – Выпуск 3. Том 12. – Иваново: МАРКОВА АД, 2013. – ЦИТ: 313-0698. – С. 76-80.
4. Давиденко І.П. Точкове задання кривих другого порядку у різноманітній параметризації // Прикладна геометрія та інженерна графіка: праці Таврійської державної агротехнічної академії. – Мелітополь: ТДАТА, 2006. – Вип. 4. – Т. 31. – С. 128-132.
5. Давиденко І.П. Точкове задання поверхонь оболонок на різноманітних планах // Зб. матеріалів Міжнародної українсько-російської науково-практичної конференції «Сучасні проблеми геометричного моделювання». – Харків: Харківський державний університет харчування та торгівлі, 2005. – С. 107-113.

УДК 51-73:517.9:519.62:53.04

**Математическое моделирование явлений тепломассопереноса
в диагностике материалов**

Степович М.А.

Калужский государственный университет им. К.Э. Циолковского

кафедра высшей математики

E-mail: m.stepovich@rambler.ru

Аннотация

Степович М.А. Математическое моделирование явлений тепломассопереноса в диагностике материалов. *Методами математического моделирования рассмотрены некоторые задачи тепломассопереноса в материалах, облученных остро сфокусированными или широкими электронными пучками.*

Общая постановка проблемы. Сфокусированные электронные пучки (электронные зонды) широко используются для диагностики различных объектов в физике, материаловедении, минералогии и в других областях науки и техники. При взаимодействии электронного зонда с мишенью в последней выделяется энергия, что может приводить к повышению температуры мишени, а в некоторых случаях изменению, в т.ч. и необратимому, её свойств. В полупроводниковых объектах микро- и наноэлектроники взаимодействие электронного пучка с мишенью приводит также к генерации электронно-дырочных пар с последующей диффузией неосновных носителей заряда в полупроводнике. Таким образом, для количественной локальной диагностики различных объектов необходимо знание закономерностей тепломассопереноса, что и является объектом изучения. В настоящей работе кратко излагаются основные идеи решения рассматриваемых задач, а подробное изложение разработанных методов изложено в цитируемой литературе.

Исследования.

1. Анализ температурных полей, возникших в результате облучения поверхности твёрдого тела остро сфокусированным пучком электронов, может иметь важное практическое значение в различных отраслях науки и техники, в частности, при диагностике объектов микро- и наноэлектроники методами тока, наведённого электронным зондом или катодоллюминесценции, использование которых позволяет проводить измерения электрофизических параметров полупроводников с высокой локальностью (около микрометра и менее – см., например). В катодоллюминесцентной микроскопии облучение образца может привести к повышению локальной температуры и, как следствие, к изменению энергетических характеристик полупроводника (ширины запрещённой зоны и т.п.). В результате изменится спектр катодоллюминесцентного излучения, что

может привести к необходимости учёта этого явления при проведении количественных измерений.

Температурное поле описывается трехмерным стационарным уравнением:

$$a^2 \left(\frac{\partial^2 \Delta T}{\partial x^2} + \frac{\partial^2 \Delta T}{\partial y^2} + \frac{\partial^2 \Delta T}{\partial z^2} \right) = -F(x, y, z).$$

Здесь a^2 – коэффициент температуропроводности, $\Delta T(x, y, z) = T(x, y, z) - T_0$, где T – температура образца в точке (x, y, z) после установления стационарного режима под воздействием электронного пучка, T_0 – температура образца до воздействия электронного пучка, а функцию источников тепла можно записать следующим образом: $F(M) = P(M)/(c\rho)$, где $P(M)$ – функция потерь энергии электронами зонда [1-3], c – удельная теплоемкость мишени, ρ – плотность материала. Функция $\Delta T(x, y, z)$ удовлетворяет следующим граничным условиям:

$$\lim_{x \rightarrow -\infty} \Delta T = 0, \quad \lim_{x \rightarrow +\infty} \Delta T = 0, \quad \lim_{y \rightarrow -\infty} \Delta T = 0, \quad \lim_{y \rightarrow +\infty} \Delta T = 0,$$

$$\lim_{z \rightarrow +\infty} \Delta T = 0,$$

а также граничному условию теплообмена с внешней средой $k \cdot \partial \Delta T(z=0)/\partial z = \alpha \Delta T(x, y, 0)$,

где α – коэффициент внешней теплоотдачи.

В результате распределение тепла $\Delta T(x, y, z)$ определится следующим образом [4, 5]:

$$\Delta T = \int_D G(x_0, y_0, z_0, x, y, z) (P(x_0, y_0, z_0)/k) dV_Q,$$

где D – область распространения тепла $(-\infty < x_0 < +\infty, -\infty < y_0 < +\infty, 0 \leq z_0 < +\infty)$, $dV_Q = dx_0 dy_0 dz_0$, а функция Грина

$$G = 1/4\pi R + 1/4\pi R_1 - \hbar \int_0^{+\infty} \left(\exp(-h\eta) / \sqrt{p^2 + (z + z_0 + \eta)^2} \right) d\eta,$$

$$\text{где } R = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2} \quad \text{и} \quad R_1 = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z+z_0)^2}.$$

Результаты некоторых расчётов для GaAs – полупроводника со средним номером в периодической таблице элементов приведены на рисунке.

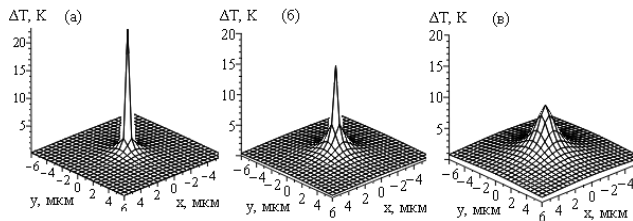


Рис1. Распределение температуры в GaAs. Расчеты проведены для энергии электронов $E_0 = 5$ (а), $E_0 = 10$ (б) и $E_0 = 20$ кэВ (в) при токе зонда 10^{-7} А.

2. Распределение неосновных носителей заряда (ННЗ), генерированных широким электронным пучком в однородном полупроводнике, может быть рассчитано как суперпозиция распределений $\Delta p(z, z_0)$ от планарных бесконечно тонких слоёв полупроводника. Здесь функция $\Delta p(z, z_0)$ описывает распределение ННЗ, генерированных плоским бесконечно тонким источником, находящимся на глубине z_0 , $z_0 \in [0, \infty)$, а z – координата, отсчитываемая от плоской поверхности вглубь полупроводника. Распределение $\Delta p(z, z_0)$ в однородном полупроводнике находится как решение дифференциального уравнения:

$$D \frac{d^2 \Delta p(z, z_0)}{dz^2} - \frac{\Delta p(z, z_0)}{\tau} = -\rho(z) \delta(z - z_0)$$

с граничными условиями

$$D \cdot d\Delta p(0, z_0)/dz = v_s \Delta p(0, z_0), \quad \Delta p(\infty, z_0) = 0,$$

а искомое распределение $\Delta p(z) = \int_0^\infty \Delta p(z, z_0) dz_0$ [6]. Здесь L – диффузионная длина, τ – время жизни, D – коэффициент диффузии и v_s – скорость поверхностной рекомбинации ННЗ.

3. Распределение неосновных носителей заряда, генерированных широким электронным пучком в многослойной планарной полупроводниковой структуре, может быть рассчитано на основе вышеизложенного подхода (см. [7-9]), однако на практике это сопряжено с определёнными усилиями и нам удалось получить аналитические выражения для числа слоёв, не более 3. Представляется перспективным использовать для решения этой задачи методов аппарата Бера

(см., например, [10]) и некоторые результаты в этом направлении получены [11], однако вычисления для конкретных материалов пока не проводились.

4. Использование катодоллюминесцентного излучения для идентификации электрофизических параметров прямозонных полупроводников. Катодоллюминесцентное (КЛ) излучение возникает при взаимодействии электронного пучка с поверхностью мишени и может быть использовано для определения параметров исследуемого полупроводника. Для прямозонных полупроводников некоторые параметры могут быть получены из измерений зависимости интенсивности монохроматической КЛ I от энергии электронов пучка E_0 . Функциональную зависимость, связывающую интенсивность монохроматической КЛ с энергией электронов пучка, запишем в следующем виде: $I = I(E_0, \Theta)$ где $\Theta = (\theta_k)$ – вектор параметров, $k = \overline{1, p}$, p – число параметров. Для определения электрофизических параметров полупроводников может быть использован т.н. конфлюэнтный анализ [12, 13], который позволяет учесть погрешности измерений I и E_0 , получить несмещенные точечные и интервальные оценки параметров Θ , а также точечные и интервальные оценки функциональной зависимости $I(E_0, \Theta)$ [14, 15]. Исходная модель оценивания параметров Θ функциональной зависимости $I(E_0, \Theta)$ с учетом погрешностей измеряемых величин имеет вид:

$$I_i = I(E_0^{(i)}, \Theta) + \delta_i, \quad E_{0i} = E_0^{(i)} + \varepsilon_i.$$
 Здесь $E_0^{(i)}$ – истинные (неизвестные) значения энергии электронов пучка, отвечающие измеренным значениям E_{0i} , $i = \overline{1, n}$, n – количество измерений; δ_i и ε_i – случайно распределенные погрешности величин I_i и E_{0i} соответственно.

В практике физических измерений часто используется нормальное распределение, поэтому примем, что ошибки измерений δ_i и ε_i – нормально распределенные случайные величины с нулевыми средними, дисперсиями $\sigma^2(I_i)$ и $\sigma^2(E_{0i})$, и коэффициентом корреляции $\rho_i = 0$. Тогда совместная плотность распределения, отвечающая вероятности получить точку с координатами (E_{0i}, I_i) будет иметь вид:

$$P_i = [2\pi\sigma(E_{0i})\sigma(I_i)]^{-1} \exp\left[-(E_{0i} - E_0^{(i)})^2 / 2\sigma^2(E_{0i}) - (I_i - I(E_0^{(i)}, \Theta))^2 / 2\sigma^2(I_i)\right].$$

Оценки параметров Θ находятся из условия минимума функционала

$$F = \frac{1}{2} \sum_{i=1}^n \left[\left(E_{0i} - E_0^{(i)} \right)^2 / \sigma^2(E_{0i}) - \left(I_i - I(E_0^{(i)}, \Theta) \right)^2 / \sigma^2(I_i) \right],$$

который учитывает сумму квадратов невязок двух величин: значения функции и значения аргумента. В аналогичном функционале, полученном традиционными статистическими методами, например в методе наименьших квадратов, учитывается невязка одной величины – чаще всего значения функции.

Но истинные значения экспериментальных точек $E_0^{(i)}$ неизвестны, а известны только их доверительные интервалы. Поэтому перед тем, как найти оценки параметров Θ , необходимо оценить $E_0^{(i)}$ и только затем находить минимум функционала по параметру Θ . Оценки параметров Θ и значений $E_0^{(i)}$ определяются из условий: $\partial F / \partial \theta_k = 0$, $k = \overline{1, p}$, и $\partial F / \partial E_0^{(i)} = 0$, $i = \overline{1, n}$. Таким образом, задача нахождения оценки параметра Θ и истинных значений $E_0^{(i)}$ эквивалентна решению системы $(n+1)$ уравнений.

С помощью функционала F можно не только оценить параметры Θ , но и с заданной вероятностью γ можно определить интервальные оценки функций $I(E_0, \Theta)$, и зная оценки параметров $\hat{\Theta}$ определить дисперсии параметров $\tilde{D}(\hat{\Theta})$:

$$P\left(I(E_0, \hat{\Theta}) - t_\gamma \sqrt{\tilde{D}(I(E_0, \hat{\Theta}))} \leq I(E_0, \Theta) \leq I(E_0, \hat{\Theta}) + t_\gamma \sqrt{\tilde{D}(I(E_0, \hat{\Theta}))} \right) = \gamma \quad (3)$$

$$\tilde{D}(I(E_0, \hat{\Theta})) = \sum_{k=1}^p \left(\frac{\partial I(E_0, \Theta)}{\partial \theta_k} \right)^2 \Big|_{\Theta = \hat{\Theta}} \cdot \tilde{D}(\hat{\Theta}).$$

5. Использование проекционного метода, основанного на теории матричных операторов [16], для реализации стохастической модели коллективного движения ННЗ. Статистические характеристики распределения ННЗ по глубине были найдены в виде частичной суммы ряда Фурье-Чебышева, при этом электрофизические параметры являлись случайными величинами и имели нормальный усеченный закон распределения. Рассмотрено влияние дисперсий этих величин на распределение ННЗ по глубине. Анализ полученных результатов показал, что если закон распределения электрофизических параметров будет близок к нормальному усеченному, то статистический разброс (среднее квадратическое отклонение) до 20 % коэффициента диффузии и времени жизни, а скорости поверхностной рекомбинации ННЗ до 100 % относительно их

средних значений не приведет к значительному статистическому разбросу распределения ННЗ по глубине (более 15 %) относительно его среднего значения [17, 18].

Список литературы

1. Михеев, Н. Н., Никоноров, И. М., Петров, В. И., Степович, М. А. Определение электрофизических параметров полупроводников в растровом электронном микроскопе методами наведённого тока и катодоллюминесценции [Текст] // Известия АН СССР. Серия физическая, Т. 54, № 2. – 1990. – С. 274-280.
2. Михеев, Н. Н., Петров, В. И., Степович, М. А. Количественный анализ материалов полупроводниковой оптоэлектроники методами растровой электронной микроскопии [Текст] // Известия АН СССР. Серия физическая, Т. 55, № 8. – 1991. – С. 1474-1482.
3. Михеев, Н. Н., Степович, М. А. Распределение энергетических потерь при взаимодействии электронного зонда с веществом [Текст] // Заводская лаборатория. Диагностика материалов, Т. 62, № 4. – 1996. – С. 20-25.
4. Амрастанов, А. Н., Гинзгеймер, С. А., Степович, М. А. Об одной возможности математического моделирования теплового воздействия электронного зонда на твёрдое тело [Текст] // Метрология та прилади, № 2 II (40). – 2013. – С. 12-15.
5. Амрастанов, А. Н., Гинзгеймер, С. А., Степович, М. А. Оценка теплового воздействия электронного пучка на однородный полупроводниковый материал. Результаты математического моделирования [Текст] // Радиационная физика твердого тела: Труды XXV международной конференции / Под ред. проф. Г.Г. Бондаренко. – М.: ФГБНУ «НИИ перспективных материалов и технологий», 2015. – С. 557-563.
6. Белов, А. А., Петров, В. И., Степович, М. А. Использование модели независимых источников для расчета распределения неосновных носителей заряда, генерированных в полупроводниковом материале электронным пучком [Текст] // Известия РАН. Серия физическая, Т. 66, № 9. – 2002. – С. 1317-1322.
7. Степович, М. А., Снопина, М. Г., Хохлов, А. Г. Использование модели независимых источников для расчёта распределения неосновных носителей заряда, генерированных в двухслойном полупроводнике электронным пучком [Текст] // Прикладная физика, № 3. – 2004. – С. 61-65.
8. Burylova, I. V., Petrov, V. I., Snopova, M. G., Stepovich, M. A. Mathematical simulation of distribution of minority charge carriers, generated in multy-layer semiconducting structure by a wide electron beam [Текст] // Физика и техника полупроводников, Т. 41, вып. 4. – 2007. – С. 458-461.
9. Снопина, М. Г., Бурьлова, И. В., Петров, В. И., Степович, М. А. Анализ модели распределений неосновных носителей заряда, генерированных в трёхслойной полупроводниковой структуре широким электронным пучком

[Текст] // Поверхность. Рентгеновские, синхротронные и нейтронные исследования, № 7. – 2007. – С. 1-6.

10. Гладышев, Ю. А. Формализм Бельтрами-Берса и его приложения в математической физике. – Калуга: Калужский гос. пед. ун-т им. К.Э. Циолковского, 1997. – 259 с.

11. Гладышев, Ю. А., Калманович, В. В., Степович, М. А. Приложение методов аппарата Берса к задачам процессов переноса в многослойной среде [Текст] // Вестник Калужского университета, № 3. – 2015. – С. 5–10.

12. Грешилов, А. А., Стакун, В. А., Стакун, А. А. Математические методы построения прогнозов. – М.: Радио и связь, 1997. – 112 с.

13. Грешилов, А. А. Анализ и синтез стохастических систем. Параметрические модели и конглоэнтный анализ. – М.: Радио и связь, 1990. – 320 с.

14. Гагарин, Ю. Е., Петров, В. И., Степович, М. А. О возможности интервального оценивания параметров полупроводников в катодолюминесцентной микроскопии. Результаты математического моделирования [Текст] // Поверхность. Рентгеновские синхротронные и нейтронные исследования, № 3. – 2004. – С. 91-94.

15. Гагарин, Ю. Е., Петров, В. И., Степович, М. А. Интервальное оценивание диффузионной длины неосновных носителей заряда в катодолюминесцентной микроскопии. Результаты математического моделирования [Текст] // Известия РАН. Серия физическая, Т. 68, № 9. – 2004. – С. 1338-1341.

16. Лапин, С. В., Егупов, Н. Д. Теория матричных операторов и ее приложение к задачам автоматического управления. – М.: МГТУ им. Н. Э. Баумана, 1997. – 496 с.

17. Серегина, Е. В., Степович, М. А., Макаренков, А. М. О модификации модели диффузии неосновных носителей заряда в полупроводниковых материалах, основанной на использовании рекурсивных тригонометрических функций, и оценка устойчивости решений для модифицированной модели [Текст] // Поверхность. Рентгеновские, синхротронные и нейтронные исследования, № 9. – 2014. – С. 72-74.

18. Серегина, Е. В., Степович, М. А., Макаренков, А. М., Филиппов, М. Н. О некоторых проблемах моделирования распределения неосновных носителей заряда, генерированных электронным пучком в полупроводниковом материале [Текст] // Поверхность. Рентгеновские, синхротронные и нейтронные исследования, № 4. – 2016. – С. 88-93.

УДК 004.67

**Информационная система для первичной обработки временных рядов
Бабенко Б.Г., Беловодский В.Н.**

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
babencko.bogdan17@yandex.ua

Аннотация

Бабенко Б.Г., Беловодский В.Н. Информационная система для первичной обработки временных рядов. Представлена разрабатываемая система для первичной обработки временных рядов

Ключевые слова: информационная система, временные ряды, сглаживание, интерполяция.

Постановка проблемы. Задача анализа временных рядов представляет собой актуальную проблему в различных областях науки и техники. В ходе решения этой задачи как правило преследуются две основные цели: определение природы ряда и прогнозирование т.е. предсказание будущих значений временного ряда по настоящим и прошлым его значениям. Все это предполагает наличия специального программного обеспечения, описанию информационной системы, позволяющей проводить первоначальную обработку временного ряда и посвящена данная статья.

Описание информационной системы. Одним из основных моментов при разработке программного продукта является формирование интерфейса, понятного и удобного для любого пользователя. В данном случае предполагается, что интерфейс проектируемой информационной системы (рис. 1) должен позволять пользователю, не имеющему особых навыков, применять систему в научных целях.

Интерфейс представляет собой рабочее окно, в котором представлены все элементы управления разрабатываемой системы.

Система позволяет:

- отобразить открытые ранее файлы для анализа;
- определить шаг и длину временного ряда;
- устанавливать временной промежуток для выбора ряда из используемой базы данных;
- заполнять присутствующие во временном ряде пустоты используя линейную или сплайн интерполяцию, в зависимости от настроек, задаваемых пользователем;
- найти и отобразить наибольшее и наименьшее значения анализируемого параметра временного ряда;

- представить временной ряд в виде таблицы;
- выполнять сглаживание временного ряда при помощи методов скользящего и экспоненциального среднего, по выбору пользователя;
- графически отображать результаты анализа ряда.

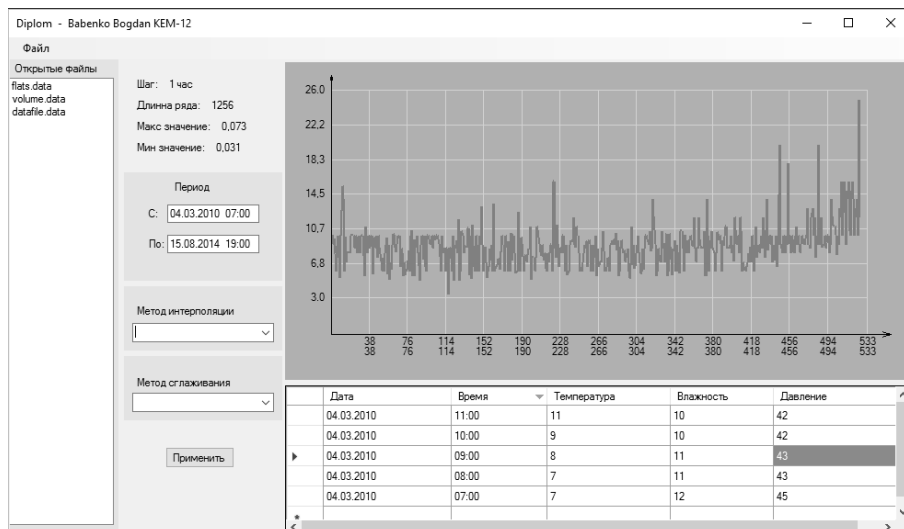


Рис. 1 – Интерфейс разрабатываемой системы

Для решения поставленной задачи предполагается использование средств и методов теории статистического анализа и обработки временных рядов.

Практическая реализация осуществляется в интегрированной среде разработки Visual Studio 2015 с использованием языка программирования C#.

Выводы. Разрабатываемая система должна позволить пользователю при минимальных затратах проводить первоначальную обработку временных рядов и может стать основой для последующей реконструкции математических моделей тех или иных процессов на базе временных рядов.

Список литературы

1. Анализ временных рядов. Курс лекций/А.Ю. Лоскутов. - МГУ - 4 с.
2. Белов А.А., Бурман В.М., Макаров С.В., Самарин А.Н., Суворовара
3. Г.П. Патент на полезную модель № 90577 «Система контроля за выбросами промышленного предприятия» от 10 января 2010 г.

УДК 004.94

3D-моделирование установки прямого восстановления железа средствами 3ds Max

Мартосенко А.Г., Пчёлкин В.Н.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
alina210595@yandex.ru

Аннотация

Мартосенко А.Г., Пчёлкин В.Н. 3D-моделирование установки прямого восстановления железа средствами 3ds Max. Выполнено изучение проблемы моделирования. Рассмотрена возможность 3D-моделирования на примере установки прямого восстановления железа типа MIDREX.

***Ключевые слова:** 3D-моделирование, 3ds Max, геометрическое моделирование, прямое восстановление железа, установка MIDREX.*

Постановка проблемы. Моделирование является общепризнанным средством познания действительности. Этот процесс состоит из двух больших этапов: разработки модели и анализа разработанной модели. Моделирование позволяет исследовать суть сложных процессов и явлений с помощью экспериментов не с реальной системой, а с ее моделью. Известно, что для принятия разумного решения по организации работы системы не обязательно знание всех характеристик системы, всегда достаточен анализ ее упрощенного, приближенного представления.

Цель статьи – изучить проблему моделирования и его виды, выполнить трехмерную визуализацию объекта (установки прямого восстановления железа типа MIDREX).

Информация о моделировании. Моделирование – это построение, совершенствование, изучение и применение моделей реально существующих или проектируемых объектов. С помощью моделирования возможно исследовать узкие места будущей системы, оценить производительность, стоимость, пропускную способность – все главные ее характеристики еще до того, как система будет создана. С помощью моделей разрабатываются оптимальные операционные планы и расписания функционирования существующих сложных систем. В организационных системах имитационное моделирование становится основным инструментом сравнения различных вариантов управляющих решений и поиска наиболее эффективного из них как для решений внутри цеха, организации, фирмы, так и на макроэкономическом уровне.

Модели сложных систем строятся в виде программ, выполняемых на компьютере. Компьютерное моделирование возникло с появлением первых

компьютеров. С тех пор сложились две перекрывающиеся области компьютерного моделирования, которые можно охарактеризовать как математическое моделирование и имитационное моделирование.

Математическое моделирование связано, в основном, с разработкой математических моделей физических явлений, с созданием и обоснованием численных методов.

Имитационное моделирование – это разработка и выполнение на компьютере программной системы, отражающей поведение и структуру моделируемого объекта.

Реальные объекты и ситуации обычно сложны, и модели нужны для того, чтобы ограничить эту сложность, дать возможность понять ситуацию, понять тенденции изменения ситуации (спрогнозировать будущее поведение анализируемой системы), принять решение по изменению будущего поведения системы и проверить его.

Геометрическое моделирование – создание геометрических моделей и оперирование ими в процессе синтеза геометрии проектируемых изделий. Геометрическое моделирование изучает методы построения численных моделей геометрии реальных или воображаемых объектов, а также методы управления этими моделями. Геометрическая модель содержит описание формы моделируемого объекта и описание связей элементов модели. Для возможности редактирования и создания подобных моделей в геометрическую модель часто включают дерево построения, хранящее последовательность и способы построения модели. Элементы геометрической модели, как правило, наделяют атрибутами, которые несут информацию о физических и других свойствах этих элементов. Геометрическое моделирование начало свое развитие с систем компьютерного черчения. Позже появились системы каркасного и поверхностного моделирования. Компьютерные системы параметрического твердотельного моделирования кардинально изменили технологию работы конструктора. Они позволили фиксировать конструкторскую мысль не в виде плоского чертежа, а в виде трехмерной модели. Для описания связей элементов модели стали применяться вариационные методы.

Человек в своей деятельности обычно вынужден решать две задачи – экспертную и конструктивную. В экспертной задаче на основании имеющейся информации описывается прошлое, настоящее и предсказывается будущее. Суть конструктивной задачи заключается в том, чтобы создать нечто с заданными свойствами. Для решения экспертных задач применяют так называемые описательные модели, а для решения конструктивных – нормативные.

В данной работе будет рассмотрена описательная модель. Описательные модели (дескриптивные, познавательные) предназначены для описания свойств или поведения реальных (существующих) объектов. Они являются формой представления знаний о действительности.

Можно назвать следующие цели описательного моделирования в зависимости от решаемых задач:

- изучение объекта (научные исследования) – наиболее полно и точно отразить свойства объекта;
- управление – наиболее точно отразить свойства объекта в рабочем диапазоне изменения его параметров;
- прогнозирование – построить модель, способную наиболее точно прогнозировать поведение объекта в будущем;
- обучение – отразить в модели изучаемые свойства объекта.

Построение описательной модели происходит по следующей схеме: наблюдение, кодирование, фиксация. Модель объекта можно построить только наблюдая за ним. То, что мы наблюдаем, необходимо закодировать либо с помощью слов, либо символов, в частности, математических, либо графических образов, либо в виде физических предметов, процессов или явлений. И закодированные результаты наблюдения нужно зафиксировать в виде модели [1].

Геометрическое моделирование, в свою очередь, может рассматриваться не только как описательное, но и как часть математического, так как построение геометрической модели сложного пространственного объекта есть ни что иное, как создание его параметрической модели. Последнее состоит из нескольких этапов: 1) представление поверхности объекта в виде совокупностей некоторых элементарных поверхностей (трехмерных примитивов): плоскостей, цилиндров, сфер и т.п.; 2) определение размеров каждого из этих примитивов; 3) определение порядка построения объекта найденных примитивов путем комбинации соответствующих аффинных преобразований (параллельных переносов, поворотов, масштабирования); 4) выражение размеров всех примитивов через некоторый единственный, базовый, размер - параметр. Построенная таким образом модель будет инвариантной относительно любых ее аффинных преобразований как целого.

Процесс создания параметрической модели объекта не требует наличия каких-либо компьютерных инструментальных средств трехмерного моделирования, он может быть осуществлен вручную. Более того, имея данную модель, можно очень быстро и легко визуализировать ее при помощи подходящих программных средств. В качестве такого программного средства была выбрана система 3ds Max 2014, которая позволяет создавать сложные движущиеся 3D-объекты из трехмерных примитивов.

Autodesk 3ds Max (ранее 3D Studio MAX) – полнофункциональная профессиональная программная система для создания и редактирования трёхмерной графики и анимации, доработанная компанией Autodesk. Содержит самые современные средства для художников и специалистов в области мультимедиа. Работает в операционных системах Windows и Windows NT (как в 32-битных, так и в 64-битных).

3ds Max располагает обширными средствами для создания разнообразных по форме и сложности трёхмерных компьютерных моделей, реальных или фантастических объектов окружающего мира, с использованием разнообразных техник и механизмов [4].

Для рассмотрения процесса моделирования используем установку для прямого восстановления железа.

Описание процесса твердофазного (прямого) восстановления железа. Твердофазное (прямое) восстановление железа – это восстановление железа из железной руды или окатышей с помощью газов (CO , H_2 , NH_3), твердого углерода, газов и твердого углерода совместно. Процесс ведется при температуре около $1000\text{ }^\circ\text{C}$, при которой пустая порода руды не доводится до шлакования [2]. Этот процесс позволяет производить наиболее высококачественный, чистый по вредным примесям металл, пригодный для использования не только в стандартных отраслях промышленности, но и также в областях, где применяются высокие технологии.

В настоящее время в мире работает много установок прямого восстановления, однако наиболее распространенными являются Мидрекс (MIDREX, США) и ХиЛ (HyL, по названию фирмы Hojalata-y-Lamina, Мексика). Способом Мидрекс осуществляется примерно $2/3$ всего мирового производства железа прямого получения, способом ХиЛ – примерно $1/4$.

Далее будем рассматривать установку типа MIDREX.

Главным отличием процесса Мидрекс является способ конверсии природного газа, которая в этом процессе осуществляется диоксидом углерода, содержащимся в отходящем из печи газе, по реакции $\text{CH}_4 + \text{CO}_2 = 2\text{CO} + 2\text{H}_2$. Перед подачей отходящего газа в конверсионную установку он проходит очистку от пыли и H_2O . Конвертированный газ, содержащий $\sim 35\%$ CO и $\sim 65\%$ H_2 , подают в печь при температуре $750\text{ }^\circ\text{C}$. Кроме этого в нижнюю часть печи подают охлажденный оборотный газ. Охлажденные окатыши содержат $\sim 95\%$ Fe и $\sim 1\%$ C . Содержание углерода в губке при необходимости может быть повышено.

Металлизированные охлажденные окатыши непрерывно выгружаются в бункер емкостью 5 тыс. т, где хранятся в инертной атмосфере до плавки в дуговых печах. Расход природного газа на процесс составляет около 350 м^3 на 1 т продукта [3].

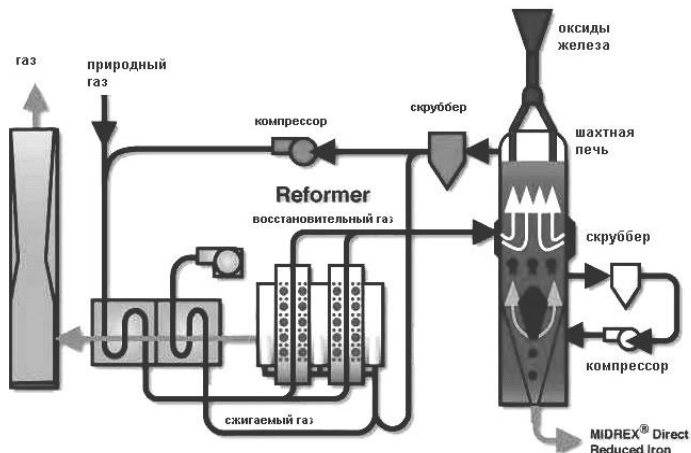


Рис.1 – Схема установки MIDREX

Построение трехмерной модели. Рассмотрим моделирование на примере основной части установки – шахтной печи. Для построения трехмерной модели понадобятся такие примитивы, как параллелепипед, усеченный конус, цилиндр, тор.

Разработанная модель будет представлена в разрезе для того, чтобы была возможность рассмотреть все компоненты шахтной печи как с внешней стороны, так и с внутренней.

В начале разработки был создан каркас модели шахтной печи, состоящий из двух полых усеченных конусов и полого цилиндра, разрезанных вдоль пополам. Также под установкой расположен вагон, представленный с помощью полого параллелепипеда и двух торов (колеса). Усеченные конусы создаются с помощью команды Cone и задаются такими параметрами: радиус1, радиус2, высота. Цилиндр (Cylinder) задается параметрами радиус и высота. Параллелепипед (Box) задается с помощью длины, ширины, высоты. Создание полой фигуры рассмотрим на примере усеченного конуса. Для начала рисуем конус1 с такими параметрами: радиус1=15, радиус2=45, высота=60 и такими координатами: X=0, Y=0, Z=100. Затем рисуем конус2, имеющий такие же координаты, но иные параметры: радиус1=12.5, радиус2=42.5, высота=60. После этого нужно из конус1 вычесть конус2. Для этого вместо Standart Primitives открываем Compound Objects и выбираем Boolean. Затем рисуем две верхние трубы, куда засыпаются металлизированные окатыши. Они представлены в виде цилиндров, имеющих поворот по X (20 градусов и -20 градусов). После этого заполняем полость шахтной печи, путем побавления цилиндров и усеченных

конусов. Так как в данной работе показана модель в разрезе, то цилиндры и конусы, изображающие полость шахтной печи, нужно разрезать пополам с помощью функции Boolean. Для окончательного представления модели требуется окрасить построенные объекты в реалистичные цвета. Цвета были подобраны в цветовом режиме RGB. Результат моделирования шахтной печи представлен на рисунке 2.

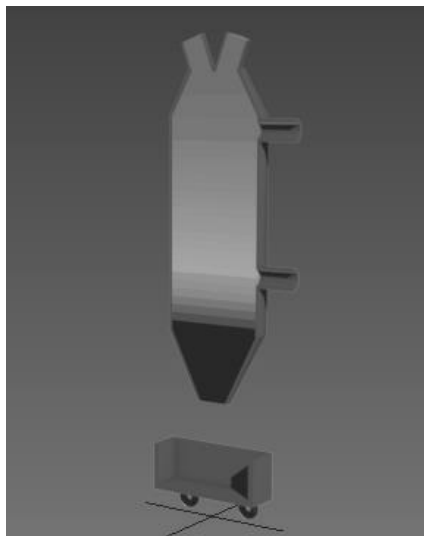


Рис. 2 – Модель шахтной печи в разрезе

Выводы. В статье рассмотрены виды моделирования, описан процесс твердофазного (прямого) восстановления железа. Также рассмотрена установка типа MIDREX, которая используется в качестве примера моделирования. Описан процесс моделирования шахтной печи установки в программной среде 3ds Max 2014.

Список литературы

1. Модели и моделирование [электронный ресурс] // E-educ.ru: [сайт]. URL: <http://e-educ.ru/tsisa18.htm>
2. Железо прямого восстановления [электронный ресурс] // Объединенная сырьевая компания: [сайт]. URL: <http://www.urmc-company.ru/production/dri>
3. Процессы твердофазного восстановления железа [электронный ресурс] // StudFiles: [сайт]. URL: <http://www.studfiles.ru/preview/1865152/page:8/>
4. Autodesk 3ds Max [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. URL: https://ru.wikipedia.org/wiki/Autodesk_3ds_Max

УДК 004.92

**Прогнозирование траектории движения подвижного объекта
распределенного симулятора тяжелой инженерной техники
Койбаш А.А., Кривошеев С.В.**

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
alexandr.koybash@gmail.com

Аннотации

Койбаш А.А., Кривошеев С.В. Прогнозирование траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники. Выполнен анализ алгоритма поиска кратчайшего пути A^* для подвижных объектов, исследовано применение архитектуры параллельных вычислений CUDA для прогнозирования. Проведен сравнительный анализ работы алгоритма на многоядерном процессоре и на GPU с использованием технологии CUDA.

Ключевые слова: прогнозирование поведения объекта, поиск кратчайшего пути, CUDA.

В настоящее время рост науки неизбежно приводит к внедрению новейших информационных разработок во многие сферы жизни. При помощи современных знаний происходят открытия новейших технологий, и значительную роль в этом играет моделирование. Одним из наиболее важных аспектов является предсказание поведения подвижных объектов с целью снижения влияния человеческого фактора. Для этого используются различные критерии, такие как вероятность определенного действия объекта, возможность выполнения какого-либо действия, а также характеристики действий и самого объекта. Есть множество способов описать схему поведения, и наиболее оптимальным является описание при помощи графов.

Наиболее рациональным является выполнение предсказания поведения объекта с помощью алгоритма A^* – нахождение кратчайших путей полученного графа. Современные моделирования очень ресурсоёмки, поэтому важную роль играет и быстродействие. Для повышения эффективности вычислений необходимо распараллеливание процессов на GPU при помощи CUDA. В отличие от CPU, графические процессоры нацелены на максимальную многопоточность.

Задача алгоритма поиска A^* состоит в том, что некий подвижный объект должен перейти из точки А в точку В по оптимальной траектории. Участок перемещения объекта содержит различные препятствия, которые необходимо обойти. Кроме этого, возможно добавление различных критериев (минимальный/максимальный угол поворота, ускорение/замедление и т.п.) для перемещения по прогнозируемой траектории.

Для решения этой задачи поле с максимально допустимой детализацией представляется в виде мультиграфа. При этом расстояние является весом ребра. Объект может ходить по горизонтали и вертикали, например, со стоимостью пути 10. Также возможно перемещение по диагонали, стоимость которого является 14 (корень из $10^2 + 10^2$). Такое поле представлено на рис 1. Чем сильнее детализация поля, тем более точным будет проложенный путь.

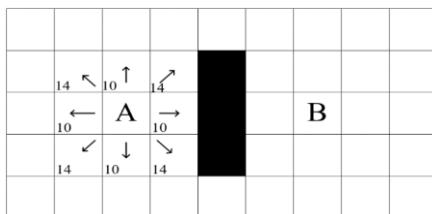


Рис.1 – Поле перемещения объекта

Поскольку исследуемым объектом является движущаяся техника, то самым рациональным является алгоритм Дейкстры, так как он наиболее оптимален при условии, что ребра не будут иметь отрицательного веса – нет убыточных движений [1]. Наилучшим решением для просчета кратчайшего пути является эвристический подход к алгоритму Дейкстры – алгоритм А*. В данном алгоритме пошагово просматриваются все пути, ведущие из вершины А в вершину В, пока не найдется путь с минимальной стоимостью.

Алгоритм реализуется следующими шагами:

1. Формируются 2 списка вершин: уже рассмотренные вершины и ожидающие рассмотрения. В список рассмотренных заносится А.
2. Для каждой соседней вершины производится расчет $F = G + H$, где G – стоимость пути от старта, а H – эвристическое приближение стоимости пути.
3. Выбирается вершина с наименьшим F . Если она является В, то сохраняется путь, двигаясь от В назад к целевой точке. Иначе эта вершина выбирается текущей, удаляется из открытого списка и заносится в закрытый.
4. Для каждой из точек, соседних с текущей, производятся следующие действия:
 - а) если соседняя точка уже рассмотрена, то она пропускается.
 - б) если соседней точки нет в списке ожидающих рассмотрения, то она добавляется в этот список. Текущая клетка для неё становится родительской. Рассчитывается F, G, H .
 - в) если соседняя клетка уже в списке ожидающих рассмотрения, то проверяется, не дешевле ли путь через эту клетку. Стоимость сравнивается по G . Если стоимость меньше, то родитель клетки меняется на текущую клетку и для неё пересчитываются стоимости G и F .

5. Если список ожидающих рассмотрения пуст, а цели нет, то это означает, что маршрута до цели не существует.

Таким образом, алгоритм A^* не только проведет движущийся объект беспрепятственно к цели, но и при этом обходит минимальное количество вершин, поскольку он эффективно использует эвристику.

С точки зрения аппаратных затрат данные вычисления являются ресурсоёмкими, т.к. необходимо учитывать множество факторов при прогнозировании траектории движения объекта. Для повышения быстродействия вычислений возможно применение не только многоядерного процессора CPU, но и графического процессора GPU.

Графический процессор GPU – это устройство, которое выполняет графическую обработку (рендеринг). Сегодня GPU эффективно применяется для обработки данных в компьютерной графике. Главной отличительной особенностью по сравнению с CPU является архитектура, максимально нацеленная на увеличение скорости расчёта.

Для графического просчета необходима параллельная обработка данных. GPU изначально мультиточечен. Архитектура спроектирована так, чтобы использовать большое количество нитей. Если современные CPU содержат несколько ядер (от 2 до 128), то в графическом процессоре количество ядер достигает несколько тысяч. Сравнительная схема CPU и GPU представлена на рис.2.

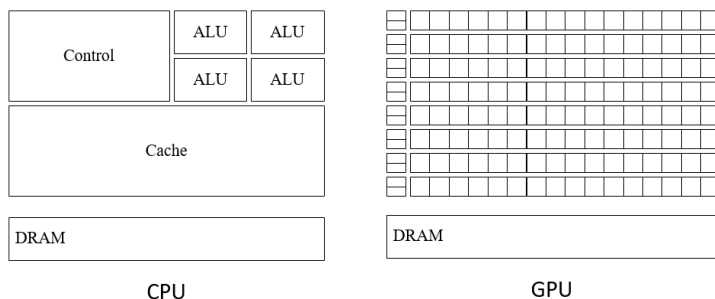


Рис. 2 – Сравнительная схема CPU и GPU

CPU лучше работает с последовательными задачами. При большом объеме потоков информации преимущество имеет GPU. Для разработки программ, использующих GPU, применяется технология CUDA – программно-аппаратная архитектура параллельных вычислений от компании Nvidia. В технологию CUDA включен унифицированный шейдерный конвейер, позволяющий программе задействовать любое АЛУ в GPU [2]. Схема организации памяти программы CUDA отображена на рис. 3.

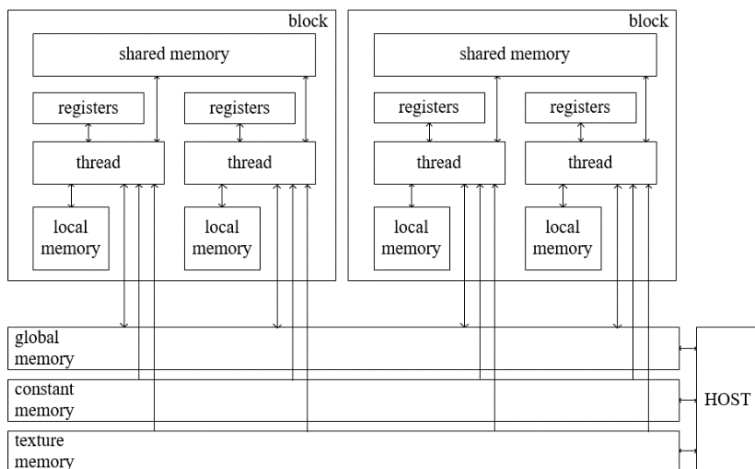


Рис. 3 – Схема организации памяти CUDA-программы.

Сравнение реализации алгоритма на CPU и на GPU показало, что для вычисления оптимальной траектории движения подвижного объекта более эффективно использовать графическую архитектуру. Использование библиотек CUDA дает возможность распараллелить процессы, тем самым увеличив производительность вычислений. Реализация данной программы на GPU также позволяет более точно вычислить траекторию, так как можно повысить интенсивность дискретизации поля.

Для прогнозирования оптимальной траектории движения был использован алгоритм A*, эффективно использующий эвристику. Во время исследования была проведена сравнительная характеристика вычисления алгоритма на многоядерном процессоре и на GPU, которая показала преимущества многопоточного вычисления этой задачи.

Список литературы

1. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ — 3-е изд. — М.: «Вильямс», 2013. — С. 499-502.
2. Джейсон Сандерс, Эдвард Кэндрот Технология CUDA в примерах. Введение в программирование графических процессоров. — ДМК Пресс, 2011. — С. 20.

УДК 004.67

Синтез оптимального узкополосного нерекурсивного цифрового фильтра

Кийко А.В., Кийко С.А.

Луганский государственный университет им. В. Даля, г. Луганск

кафедра компьютерных систем и сетей

kiyko46@mail.ru

Аннотация

Кийко А.В., Кийко С.А. Синтез оптимального узкополосного нерекурсивного цифрового фильтра. В статье изложена методика синтеза нерекурсивных цифровых фильтров для селекции узкополосных сигналов, имеющих при заданном порядке минимальную ширину полосы прозрачности. Предложен критерий их оптимальности. Выведены соотношения для настройки фильтра на заданную частоту. Приведены результаты расчета амплитудно-частотных характеристик узкополосных фильтров различных порядков.

Ключевые слова: нерекурсивные цифровые фильтры, оптимизация.

Постановка проблемы. Цифровая фильтрация сигналов широко используется в системах автоматического управления и научных исследованиях, как для выделения информативных компонент, так и для подавления шумов и помех. В цифровых фильтрах (ЦФ) можно легко изменять коэффициенты, а, следовательно, и частотные характеристики фильтра, не меняя его структуры. Это дает возможность синтезировать перестраиваемые и адаптивные фильтры. Поэтому разработка методов синтеза цифровых фильтров является важной и актуальной задачей.

Большая часть публикаций в открытой печати посвящена методам синтеза рекурсивных ЦФ. Полюсы передаточной функции рекурсивного фильтра могут располагаться произвольно внутри единичной окружности z -плоскости. Следствием такой свободы является высокая избирательность рекурсивного фильтра, которую можно получить при использовании передаточной функции низкого порядка. При одинаковых требованиях к избирательности порядок передаточной функции для нерекурсивного ЦФ с полюсами, фиксированными в начале координат, может быть в 5-10 раз выше, чем для рекурсивного. Тем не менее нерекурсивные ЦФ обладают рядом достоинств. Например, они всегда абсолютно устойчивы (при фильтрации с помощью таких фильтров не происходит накопление ошибок). А каждый вновь синтезированный рекурсивный ЦФ должен подвергаться специальному исследованию на устойчивость. Кроме того, легко создать нерекурсивный ЦФ со строго линейной фазовой характеристикой, которая требуется для учета дисперсионных искажений при

обработке речи и передаче данных. Поэтому на практике во многих случаях нерекурсивные фильтры являются более предпочтительными.

Цель статьи – разработать методику синтеза оптимальных узкополосных нерекурсивных ЦФ, обладающих при фиксированном порядке минимальной шириной полосы пропускания.

Анализ последних исследований и постановка задачи. Известны [1] три класса методов расчета нерекурсивных фильтров с линейной фазой: методы взвешивания с помощью "окна", методы частотной выборки, а также методы расчета оптимальных фильтров на основе алгоритма Ремеза. Перечисленные методы широко используются для синтеза ЦФ, имеющих одну или несколько полос пропускания. На практике часто возникает необходимость выделения из сигнала компоненты с заданной частотой, для чего необходимо использовать ЦФ с чрезвычайно малой шириной полосы пропускания. При этом достаточно синтезировать оптимальные узкополосные нерекурсивные ЦФ для центральной частоты $\omega_0 = 0$, т.е. оптимальные инфранизкочастотные нерекурсивные цифровые фильтры (ИНЦФ), так как их можно легко перестроить на любую заданную частоту ω_0 . Обычно на практике желательно иметь узкополосный ЦФ, модуль амплитудно-частотной характеристики (АЧХ) которого симметричен относительно центра ω_0 его полосы пропускания (для ИНЦФ это условие сводится к симметрии модуля его АЧХ относительно оси ординат).

Синтез оптимального ИНЦФ. Дискретная передаточная функция ИНЦФ порядка $2m$ с симметричной АЧХ, реализуемого в последовательной (каскадной) форме, имеет вид:

$$W_{2m}(z^{-1}) = \prod_{j=1}^m (1 - z_j z^{-1})(1 - \bar{z}_j z^{-1}),$$

где z_j и \bar{z}_j – комплексно-сопряженные нули дискретной передаточной функции $W_{2m}(z^{-1})$: $z_j = \rho_j e^{i\varphi_j}$, $\bar{z}_j = \rho_j e^{-i\varphi_j}$;

m – число каскадов второго порядка в цифровом фильтре.

Модуль амплитудно-частотной характеристики этого фильтра:

$$H_{2m}(\Omega; \rho, \varphi) = \prod_{j=1}^m \sqrt{1 + \rho_j^2 - 2\rho_j \cos(\Omega - \varphi_j)} \cdot \sqrt{1 + \rho_j^2 - 2\rho_j \cos(\Omega + \varphi_j)},$$

$$0 \leq \Omega \leq \pi,$$

где $\Omega = 2\pi \frac{\omega}{\omega_a}$ – относительная частота;

ω_a – частота дискретизации фильтруемого сигнала;

$\rho = (\rho_1, \dots, \rho_m)$ и $\varphi = (\varphi_1, \dots, \varphi_m)$ – m -мерные векторы, компонентами которых соответственно являются модули ρ_j и аргументы φ_j нулей z_j дискретной передаточной функции $W_{2m}(z^{-1})$ цифрового фильтра.

Физически реализуемый инфранизкочастотный фильтр можно рассматривать как некоторую аппроксимацию идеального фильтра нулевой частоты (т.е. постоянной составляющей сигнала), АЧХ которого представляет собой дискретную δ -функцию. Качество аппроксимации идеального ИНЦФ физически реализуемым ИНЦФ конечного порядка можно оценивать с помощью различных критериев. Рассмотрим следующий критерий близости АЧХ ИНЦФ к дискретной δ -функции:

$$K_{\delta}(\Omega_{\bar{n}}; \rho, \varphi) = \frac{\int_0^{\Omega_{\bar{n}}} H_{2m}(\Omega; \rho, \varphi) d\Omega}{\int_0^{\pi} H_{2m}(\Omega; \rho, \varphi) d\Omega} \leq 1$$

где K_{δ} – коэффициент, характеризующий “узкополосность” АЧХ ИНЦФ (т.е. ее сосредоточенность в окрестности $\Omega = 0$);

$\Omega_{\bar{n}}$ – правая граница “полосы пропускания” фильтра (интервала на оси Ω , для которого вычисляется величина K_{δ}); левой границей всегда является начало координат.

Величина $K_{\delta}(\Omega_{\bar{n}}; \rho, \varphi)$ характеризует близость $H_{2m}(\Omega; \rho, \varphi)$ к финитной функции с носителем $[0, \Omega_c]$. Точка $\Omega_{\bar{n}}$ разбивает интервал $[0, \pi]$ оси относительных частот на две части: “полосу пропускания” $[0, \Omega_c]$ и “полосу подавления” $[\Omega_c, \pi]$. Чем ближе величина K_{δ} к единице (при фиксированном достаточно малом значении $\Omega_{\bar{n}}$), тем ближе $H_{2m}(\Omega; \rho, \varphi)$ к дискретной δ -функции. Поэтому критерий оптимальности ИНЦФ конечного порядка $2m$ имеет вид:

$$\max_{\substack{\rho_j \in (0, \infty), \varphi \in [0, \pi] \\ j = 1, \dots, m}} K_{\delta}(\Omega_c; \rho, \varphi) = K_{\delta \text{ i i d}}(\Omega_{\bar{n}}) \leq 1.$$

Очевидно, что $K_{\delta \text{ i i d}}$ является неубывающей функцией от $\Omega_{\bar{n}}$, причем $K_{\delta \text{ i i d}}(0) = 0$ и $K_{\delta \text{ i i d}}(\pi) = 1$.

Таким образом, задача оптимизации ИНЦФ $2m$ -го порядка сводится к поиску такого расположения на комплексной z -плоскости нулей $z_j, j = \overline{1, m}$ его дискретной передаточной функции $W_{2m}(z^{-1})$, при котором ширина $2\Omega_{\bar{n}}$ полосы

пропускания фильтра минимальна (при фиксированном значении $K_{\sigma_{\text{нн}}}$, близком к 1).

Чем выше порядок ИНЦФ, тем меньше величина $\Omega_{\bar{n}}$ при одинаковых пороговых значениях $K_{\sigma_{\text{нн}}}$. При неограниченном возрастании m величина $\Omega_{\bar{n}}$ неограниченно убывает при фиксированном значении $K_{\sigma_{\text{нн}}}$:

$$\lim_{m \rightarrow \infty} \Omega_c = 0.$$

Выбрав желаемую величину ширины полосы пропускания $\Omega_{\bar{n}}$, можно синтезировать оптимальный ИНЦФ заданного порядка $2m$, имеющий максимальное значение $K_{\sigma_{\text{нн}}}$. В том случае, когда величина $K_{\sigma_{\text{нн}}}$ меньше порогового значения, необходимо увеличить порядок фильтра. В результате получим при заданном $\Omega_{\bar{n}}$ оптимальный ИНЦФ минимального порядка, АЧХ которого удовлетворяет условию:

$$K_{\sigma_{\text{нн}}}(\Omega_c) \geq P_K,$$

где $P_K < 1$ – заданное пороговое значение $K_{\sigma_{\text{нн}}}$.

Задача синтеза оптимального ИНЦФ, АЧХ которого удовлетворяет критерию максимальности K_{σ} , сводится к поиску максимума сложной нелинейной функции $2m$ переменных. Однако в данном случае справедливо следующее утверждение: нули оптимального инфранизкочастотного рекурсивного цифрового фильтра с симметричной АЧХ могут лежать только на единичной окружности в комплексной z -плоскости, т.е. все $\rho_j = 1$, $j = \overline{1, m}$. Использование этого доказанного в [2] факта позволяет вдвое уменьшить размерность задачи нелинейной оптимизации (необходимо найти только φ_j , $j = \overline{1, m}$), а также существенно упростить выражение для K_{σ} .

Практическая реализация. В соответствии с предложенной методикой разработан алгоритм и составлена программа на языке С# для расчета оптимальных значений аргументов нулей дискретной передаточной функции ИНЦФ различных порядков. Результаты расчетов приведены в табл. 1. Графики их АЧХ представлены на рис. 1

Для выделения из сигнала компонент с частотами $\pm\Omega_{\sigma}$ необходимо использовать два включенных параллельно узкополосных цифровых фильтра, настроенных на частоты $+\Omega_{\sigma}$ и $-\Omega_{\sigma}$ соответственно. Эти фильтры получаются из инфранизкочастотного фильтра путем поворота на комплексной z -плоскости всех его нулей соответственно на углы $+\Omega_{\sigma}$ и $-\Omega_{\sigma}$.

Таблица 1 – Аргументы нулей оптимальных ИНЦФ различных порядков

m=5	m=7	m=9	m=15
0,764615	0,572815	0,462218	0,294743
1,303625	0,976030	0,779651	0,491623
1,843538	1,379546	1,095208	0,688669
2,375172	1,772297	1,421614	0,885559
3,140471	2,165301	1,734010	1,073012
	2,587383	2,049301	1,286787
	3,140357	2,374426	1,475631
		2,691844	1,676320
		3,139822	1,859954
			2,065279
			2,273113
			2,457215
			2,653881
			2,920616
			3,096394

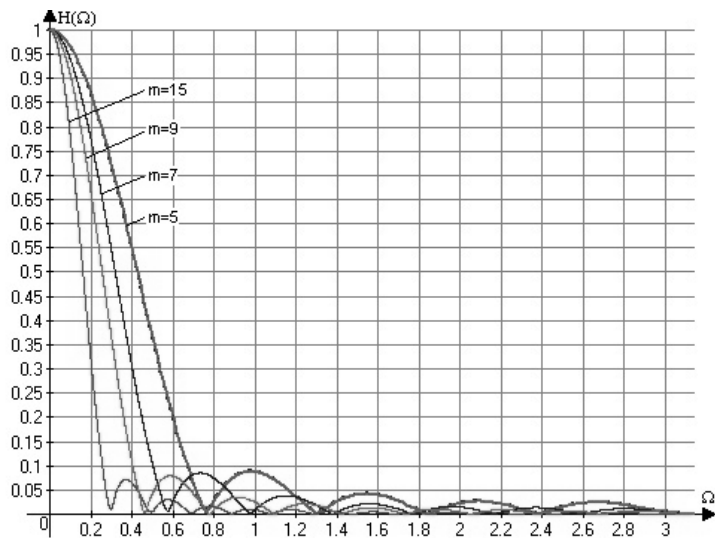


Рис. 1 – Графики АЧХ оптимальных ИНЦФ

Их общая дискретная передаточная функция имеет следующий вид:

$$W_{2m}(z^{-1}) = \frac{1}{2} \prod_{j=1}^m (1 - z_j e^{i\Omega_0} z^{-1}) (1 - \bar{z}_j e^{i\Omega_0} z^{-1}) + \frac{1}{2} \prod_{j=1}^m (1 - z_j e^{-i\Omega_0} z^{-1}) (1 - \bar{z}_j e^{-i\Omega_0} z^{-1}) =$$

$$= \frac{1}{2} \prod_{j=1}^m (1 - 2 \cos \varphi_j e^{i\Omega_0} z^{-1} + e^{i2\Omega_0} z^{-2}) + \frac{1}{2} \prod_{j=1}^m (1 - 2 \cos \varphi_j e^{-i\Omega_0} z^{-1} + e^{-i2\Omega_0} z^{-2}).$$

Обозначив $u = e^{i\Omega_0} z^{-1}$, $v = e^{-i\Omega_0} z^{-1}$, получим:

$$W_{2m}(z^{-1}) = \frac{1}{2} \prod_{j=1}^m (1 - p_j u + u^2) + \frac{1}{2} \prod_{j=1}^m (1 - p_j v + v^2) = \frac{1}{2} \sum_{k=0}^{2m} a_k u^k + \frac{1}{2} \sum_{k=0}^{2m} a_k v^k =$$

$$= \frac{1}{2} \sum_{k=0}^{2m} a_k (u^k + v^k) = \frac{1}{2} \sum_{k=0}^{2m} a_k (e^{ik\Omega_0} + e^{-ik\Omega_0}) z^{-k} = \sum_{k=0}^{2m} a_k \cos k\Omega_0 z^{-k}.$$

Коэффициенты a_k , $k = \overline{0, 2m}$ вычисляются по следующему итерационному алгоритму:

$$a_0^{(n+1)} = a_0^{(n)},$$

$$a_1^{(n+1)} = a_1^{(n)} + a_0^{(n)} p_{n+1},$$

$$\dots$$

$$a_i^{(n+1)} = a_i^{(n)} + a_{i-1}^{(n)} p_{n+1} + a_{i-2}^{(n)}, \quad i = \overline{2, 2n},$$

$$\dots$$

$$a_{2n+1}^{(n+1)} = a_{2n}^{(n)} p_{n+1} + a_{2n-1}^{(n)},$$

$$a_{2n+2}^{(n+1)} = a_{2n}^{(n)}, \quad n = 1, 2, \dots, m-1,$$

где $p_n = -2 \cos \varphi_n$.

Начальные значения коэффициентов равны:

$$a_0^{(1)} = a_2^{(1)} = 1; \quad a_1^{(1)} = p_1.$$

Таким образом, для получения коэффициентов узкополосного цифрового фильтра, настроенного на частоты $\pm\Omega_0$, достаточно умножить коэффициенты a_k инфранизкочастотного фильтра на $\cos k\Omega_0$.

Выводы. Узкополосные нерекурсивные цифровые фильтры, синтезированные с помощью предлагаемой методики, могут найти применение в системах диагностики и непрерывного контроля за техническим состоянием деталей и узлов машин, а также использоваться для анализа био-электрических сигналов.

Список литературы

1. Хемминг Р.В. Цифровые фильтры. Пер. с англ. – М.: Советское радио, 1980. – 224 с.
2. Кийко А.В. Синтез узкополосных цифровых фильтров для анализа речевых сигналов. / А.В. Кийко // Інформаційна безпека. – 2013. - № 2(10), – с. 65-83.

УДК 519.725, 625.7, 681.3

Альтернативный метод укорачивания циклических кодов

Дяченко О.Н., Дяченко В.О

Донецкий национальный технический университет, г. Донецк

кафедра компьютерной инженерии

E-mail: dyachenkoon@gmail.com

Аннотация

Дяченко О.Н., Дяченко В.О. Альтернативный метод укорачивания циклических кодов. *Предлагается использование альтернативного метода укорачивания циклических кодов, исправляющих пакеты ошибок. В статье обоснована актуальность использования укорачивания кодов, в особенности для кодов, исправляющих пакеты ошибок большой длины. Выполнен анализ существующих способов, а также их аппаратной реализации. Рассмотрена идея предлагаемого метода укорачивания, сначала на примере кода Хэмминга, а затем для кодов, исправляющих пакеты ошибок. По сравнению с традиционными подходами укорачивания кодов предлагаемый авторами отличается простотой применения, в том числе для кодов с большим параметром укорачивания.*

Введение. В настоящее и обозримое будущее время динамические изменения в различных сферах деятельности общества характеризуются лавинообразным ростом объемов самой различной информации: военной, коммерческой, мультимедийной, социально-политической, производственной, научной, технической, культурной и др. Все это признаки информационной революции.

В основе многих выдвинутых ныне теорий и концепций, объясняющих глубинные изменения в экономической и социальной структурах передовых стран мира, начавшиеся в середине XX в., лежит признание нарастания значения информации в жизни общества.

Существует множество различных теорий и классификаций научно-технических революций в истории человечества.

Э. Тоффлер выделяет три «волны» в развитии общества: аграрная при переходе к земледелию; индустриальная во время промышленной революции; информационная при переходе к обществу, основанному на знании (постиндустриальному).

А. И. Ракилов выделяет пять информационных революций: появление и внедрение в деятельность и сознание человека языка; изобретение письменности; изобретение книгопечатания; изобретение телеграфа и телефона; изобретение компьютеров и появление Интернета.

Признанный классик теории постиндустриализма Д. Белл выделяет три технологических революции: изобретение паровой машины в XVIII веке; научно-технологические достижения в области электричества и химии в XIX веке; создание компьютеров в XX веке. Белл утверждал, что, подобно тому, как в результате промышленной революции появилось конвейерное производство, повысившее производительность труда и подготовившее общество массового потребления, так и теперь должно возникнуть поточное производство информации, обеспечивающее соответствующее социальное развитие по всем направлениям.

В последние десятилетия достижения в области внедрения информационных технологий являются одним из определяющих факторов экономического потенциала общества. В результате появляется и развивается информационная инфраструктура, которая предоставляет новые услуги, такие как дистанционное образование, заказ билетов для транспорта, телеработа, телемедицина, электронная торговля, интернет-банкинг, оплата счетов и др.

Тенденция к увеличению количества информации, которую необходимо хранить, обрабатывать и передавать, неизбежно приводит к требованиям обеспечения ее достоверности. Эту задачу призваны выполнять коды, обнаруживающие и исправляющие ошибки.

Одними из наиболее эффективных для исправления ошибок являются циклические коды. Эти коды нашли широкое применение благодаря простой аппаратной реализации и высоким корректирующим способностям. Некоторые специалисты помехоустойчивого кодирования уже давно решили, что все основные теоретические вопросы относительно циклических кодов решены и свое внимание перенесли на совершенствование и разработку других кодовых конструкций [1]. Поэтому инженерам-практикам приходится самим предлагать решения для каждой конкретной технической задачи вне связи с общей стратегией развития циклических кодов.

В связи с этим решение практических вопросов построения и аппаратной реализации циклических кодов являются актуальными, тем более, учитывая все большую их популярность и востребованность для различных сфер применения.

При реализации циклических кодов во многих случаях приходится их укорачивать (например, укороченные коды Рида-Соломона над полем Галуа $GF(2^8)$ для CD-ROM, DVD и цифрового телевидения высокого разрешения - формат HDTV) [2-4, 6]. Существует несколько методов укорачивания кодов. Один из них основан на использовании двойственных порождающих полиномов. В данной работе предлагается альтернативный метод применения двойственных полиномов для кодирования и декодирования укороченных кодов, что дает преимущества при реализации кодов, исправляющих пакеты ошибок большой длины.

Основная идея альтернативного метода применения двойственных полиномов. Наиболее популярны в настоящее время коды, исправляющие пакеты ошибок: коды Файра, (255, 223, 33) код Рида-Соломона для космической связи NASA, расширенный (128, 122, 7), код Рида-Соломона над полем Галуа $GF(2^7)$ для кабельных модемов и многие другие [1-5]. Вместе с тем, циклический код Хэмминга, исправляющий одиночные ошибки, заслуживает особого внимания, поскольку является фундаментом для понимания принципов построения более мощных кодов. Поэтому, основную идею альтернативного метода применения двойственных полиномов для укорачивания циклически кодов рассмотрим на примере кода Хэмминга.

Основная идея отличия кодирования и декодирования укороченных циклических кодов заключается в следующем. Декодер выполняет исправление принятого слова традиционным способом (применяя умножение на полином, равный остатку от деления полинома X^{n-k+i} на порождающий полином и деление на порождающий полином). Но такой остаток определяется альтернативным способом, при котором параметр укорачивания не участвует. Предлагаемый способ основан на свойстве элементов полей Галуа полученных для двойственных порождающих полиномов. Эти элементы – не что иное как остатки от деления ненулевых полиномов в степенном виде, и кроме того, состояния генератора синдрома в декодере. Анализ таблицы 1 показывает взаимосвязь элементов эти полей. Добавление таблиц на 4 строки ($\deg p(z) = \deg p^*(z)$) с умножением принятого слова на полином X^{n-k+i} , где $n-k = \deg p(z)$. Существует зависимость между значениями элементов в двоичном виде. Причем рассматривать их нужно в обратном порядке следования двоичных символов, что соответствует умножению элемента от (X^{-1}) на $X^{\deg p(z)-1}$. Например, $\alpha^4 = 0011$ и $\alpha^{14} = \alpha^{-1} = 1100$, $\alpha^5 = 0110$ и $\alpha^{13} = \alpha^{-2} = 0110$, $\alpha^6 = 1100$ и $\alpha^{12} = \alpha^{-3} = 0011$ и т.д.

Ненулевые остатки от деления на порождающий полином состоят из двух подмножеств. Общее количество элементов этих подмножеств равны длине исходного кода n (сумме параметра укорачивания i и значения новой длины укороченного кода). Пересекаются эти подмножества на граничных элементах, причем они зеркально равны.

Учитывая эти свойства, можно получить остаток от деления полинома X^{n-k+i} на порождающий полином без явного учета i , а на основе только длины укороченного кода и двойственного полинома.

Таблица 1. Элементы поля Галуа $GF(2^4)$ с двойственными порождающими полиномами

$p(z)=z^4+z+1$		$p^*(z)=z^4+z^3+1$	
В виде степени	В двоичном виде	В виде степени	В двоичном виде
0	0000	0	0000
α^0	0001	$\alpha^0=\alpha^{-15}$	0001
α^1	0010	$\alpha^1=\alpha^{-14}$	0010
α^2	0100	$\alpha^2=\alpha^{-13}$	0100
α^3	1000	$\alpha^3=\alpha^{-12}$	1000
α^4	0011	$\alpha^4=\alpha^{-11}$	1001
α^5	0110	$\alpha^5=\alpha^{-10}$	1011
α^6	1100	$\alpha^6=\alpha^{-9}$	1111
α^7	1011	$\alpha^7=\alpha^{-8}$	0111
α^8	0101	$\alpha^8=\alpha^{-7}$	1110
α^9	1010	$\alpha^9=\alpha^{-6}$	0101
α^{10}	0111	$\alpha^{10}=\alpha^{-5}$	1010
α^{11}	1110	$\alpha^{11}=\alpha^{-4}$	1101
α^{12}	1111	$\alpha^{12}=\alpha^{-3}$	0011
α^{13}	1101	$\alpha^{13}=\alpha^{-2}$	0110
α^{14}	1001	$\alpha^{14}=\alpha^{-1}$	1100
α^0	0001	$\alpha^{15}=\alpha^0$	0001
α^1	0010	α^1	0010
α^2	0100	α^2	0100
α^3	1000	α^3	1000

Укороченные коды, исправляющие пакеты ошибок. Предположим, что код (511, 499) потребовалось укоротить до (272, 260)-кода [2]. Этот код исправляет все пакеты ошибок длины не более 4. В данном случае порождающий полином $g(X)=X^{12}+X^8+X^5+X^3+1$, $X^{n-k+i}=X^{251}$ и необходимо вычислить остаток $a(X) = R_{g(X)}[X^{251}]$. В [2] полином X^{251} представлен в виде $X^{251}=(X^{12})^{16}(X^{12})^4(X^{11})$ для того, чтобы воспользоваться равенством $X^{12}=X^8+X^5+X^3+1$. Повторяя возведение в квадрат полинома X^{12} и проводя редукцию по модулю $g(X)$, вычисляется $(X^{12})^4$ и $(X^{12})^{16}$, а, следовательно, и X^{251} , так что $a(X)=X^{11}+X^9+X^7+X^3+X^2+1$.

Рассмотрим вычисление для этого же остатка $a(X) = R_{g(X)}[X^{251}]$ с помощью предлагаемого способа. Сначала определяем остаток от деления

полинома в степени длины нового укороченного кода уменьшенной на единицу, т.е. X^{271} , на двойственном $g(X)$ полином $g^*(X) = X^{\text{deg}(X)} g(X^{-1}) = X^{12}(X^{-12} + X^{-8} + X^{-5} + X^{-3} + 1) = (X^{12} + X^9 + X^7 + X^4 + 1)$ любым известным способом, например, с помощью программы деления полиномов: $X^{11} + X^9 + X^8 + X^4 + X^2 + 1$. Получив остаток R и умножив $R(X^{-1})$ на полином $X^{\text{deg}(X)-1}$ находим искомый остаток $a(X) = X^{11}(X^{-11} + X^{-9} + X^{-8} + X^{-4} + X^{-2} + 1) = X^{11} + X^9 + X^7 + X^3 + X^2 + 1$.

Следует отметить важное отличие предлагаемого второго способа от первого и других известных. Из вычислений остатка во втором способе исключается параметр укорачивания i , и таким образом, становится возможной реализация кодов с большим параметром укорачивания и длиной исправляемого пакета ошибок.

Систематический код, исправляющий пакеты ошибок.

Определение параметров исходного кода.

Параметр укорачивания $i = 4$

$(3*7-4, 3*3-4)$, исходный код: $(7, 3)$, образующий полином: $C(X) = X^4 + X^3 + X^2 + 1, b = 2. \Rightarrow j = 3, i = 4$, где j параметр перемежения.

Определение параметров кода.

$(3*7-4, 3*3-4) = (17, 5)$

$n = 17, k = 5$

Определение длины исправляемого пакета ошибок

$b^* = j*b = 3*2 = 6$ (где b – длина исправляемого пакета ошибок для данного кода).

Определение образующего полинома $C^*(X)$ и $C^{**}(X)$.

$C^*(X) = X^{j*4} + X^{j*3} + X^{j*2} + 1 = X^{12} + X^9 + X^6 + 1, C^{**}(X) = X^{12}(X^{-12} + X^{-9} + X^{-6} + 1) = X^{12} + X^6 + X^3 + 1$

Определение минимального количества проверочных символов (p).

$p = n - k = 17 - 5 = 12$

Определение остатка от деления $X^{(p+i)}$ на образующий полином $R_{C^*(X)}[X^{p+i}]$.

$$\begin{array}{r} X^{16} \\ \hline X^{16} + X^{13} + X^{10} + X^4 \end{array} \left| \begin{array}{r} X^{12} + X^9 + X^6 + 1 \\ \hline X^4 + X \end{array} \right.$$

$$X^{13} + X^{10} + X^4$$

$$\underline{X^{13} + X^{10} + X^7 + X}$$

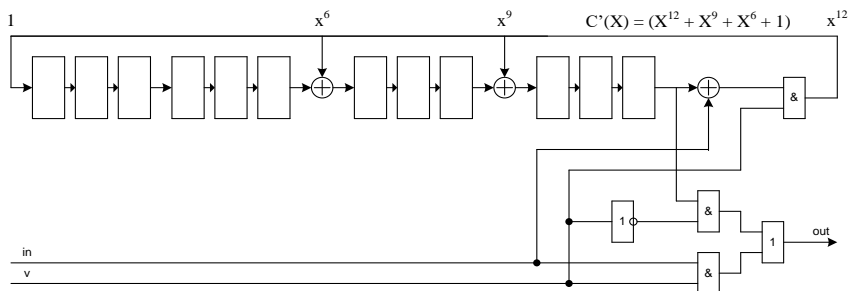
$$X^7 + X^4 + X$$

$$R_{C^*(X)}[X^{p+i}] = X^7 + X^4 + X, R_{C^{**}(X)}[X^{n-1}] = X^{10} + X^7 + X^4$$

$$R_{C^*(X)}[X^{p+i}] = X^{11}(X^{-10} + X^{-7} + X^{-4}) = X^7 + X^4 + X$$

Построение кодера для кода $(17, 5)$.

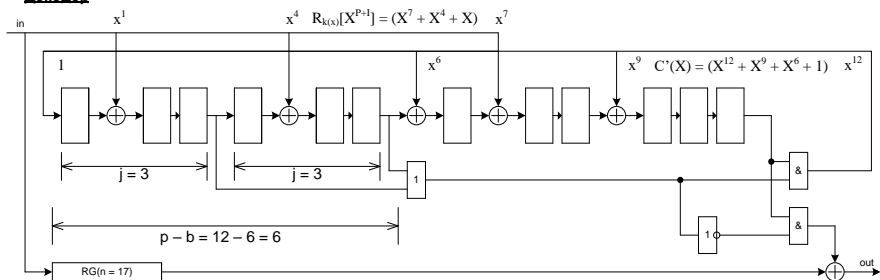
Кодер



Сигнал V используется для переключения ключей кодера. Он должен быть высокого уровня первые $k=5$ тактов и потом переключаться в низкий уровень.

Построение декодера для кода (17, 5).

Декодер



Заключение. Таким образом, в работе предложен метод построения укороченных циклических кодов, в том числе применимый для кодов с большой длиной исправляемого пакета ошибок. Вместе с тем, для таких кодов можно использовать традиционную аппаратную или программную реализацию кодеров и декодеров.

Список литературы

1. Семеренко В. П. Теория и практика CRC кодов: новые результаты на основе автоматных моделей // Восточно-Европейский журнал передовых технологий. – 2015. – 4/9. – С. 38-48.
2. Richard E. Blahut. Algebraic Codes for Data Transmission/ Cambridge University Press, 2012. – 498 p.
3. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. – М.: Мир, 1976. – 595 с.: ил.
4. Дяченко В.О., Зинченко Ю.Е., Дяченко О.Н. Исследование способов проектирования кодов Рида-Соломона// Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС КМ-2014) : V Всеукраїнська науково-технічна конференція студентів, аспірантів та молодих вчених, 22-23 квітня 2014 р., м. Донецьк : зб. доп./ Донец. націонал. техн. ун-т; редкол. В.А.Світлична. – Донецьк: ДонНТУ, 2014. – в 2 тт. – т.2. – С. 72-78.
5. Дяченко В.О., Дяченко О.Н. Анализ способов реализации кодов Рида-Соломона, исправляющих двойные ошибки// Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике: материалы Международной научно-практической конференции (Азов, 19 мая 2014 г.). – Ростов н/Д, ДГТУ, 2014. – С. 18-22.
6. Дяченко В.О., Дяченко О.Н. Особенности применения двойственных полиномов для аппаратной реализации циклических кодов // Информационные управляющие системы и компьютерный мониторинг в рамках форума “Инновационные перспективы Донбасса” (ИУС КМ-2015): VI Международная научно-техническая конференция студентов, аспирантов и молодых ученых, 20-22 мая 2015, г.Донецк: / Донец. национал. техн. ун-т; сост.: К.Н.Маренич (председатель) и др. – Донецк: ДонНТУ, 2015. - С. 130–136.
7. 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.) [Электронный ресурс]. – Ростов н/Д, ДГТУ, 2015. – С. 71-76. – Режим доступа: <http://atidstu.ru/atidgtu-rf/node/1163>.

Секция № 6
Системы искусственного
интеллекта



УДК 004.8

**Применение генетического алгоритма при построении игрового
искусственного интеллекта коллекционной карточной игры
Жудин А.Ю., Рудова Е.А.**

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: nerpb14@mail.ru

Аннотация

Рудова Е.А., Жудин А.Ю. Применение генетического алгоритма при построении игрового искусственного интеллекта коллекционной карточной игры. Рассмотрены проблемы игрового искусственного интеллекта на примере существующих коллекционных карточных игр. Определен подход разработки алгоритмов работы игрового искусственного интеллекта на основе генетического алгоритма.

Общая постановка проблемы. В последнее время игровая индустрия стала бурно развиваться и написание алгоритмов, которые способны имитировать интеллект человека, становится актуальной темой.

Игровой искусственный интеллект присутствует практически во всех жанрах компьютерных игр, но в рамках данной статьи будет рассмотрено построение игрового искусственного интеллекта в жанре коллекционных карточных игр с использованием подхода генетического алгоритма.

В компьютерных играх роль ИИ сводится к тому, чтобы создать у игрока ощущение, будто они играют против реальных, разумных соперников. Разработка ИИ для каждого случая производится отдельно и является уникальной, но существуют готовые алгоритмические решения, которые дадут базу для создания ключевого модуля. Алгоритмы поиска пути присутствует в том или ином виде почти у каждого ИИ для различных целей. Примером могут служить алгоритмы поиска путей, а именно:

- Алгоритм Дейкстры – работающий только с графами без ребер отрицательного веса;
- Алгоритм Ли, который используется при трассировке печатных плат или же в поиске кратчайшего пути в компьютерных играх.

Существуют и некоторые алгоритмические приемы взаимодействия между персонажами с ИИ. Имеет место ситуация, в которой разработчики намеренно встраивают возможность стрельбы союзных персонажей с ИИ друг в друга. Это случается, если один бот преграждает путь для стрельбы другому [1]. Уникальность каждого ИИ достигается в сфере его применения и окружающей обстановки, под которую он программируется и часто, для достижения целей

программистам приходится создавать новые решения для реализации искусственного интеллекта под свою задачу.

Рассмотрение проблемы игрового искусственного интеллекта на примере существующих коллекционных карточных игр

Игровой искусственный интеллект (ИИИ) в отличие от просто искусственного интеллекта должен быть реализован малой ценой в рамках поставленного бюджета, но чтобы этого было достаточно для достижения поставленной задачи.

Для рассмотрения аналогов ИИИ жанра коллекционных карточных игр (ККИ) были взяты две игры в разных сферах своего существования, а именно: *Hearthstone: Heroes of Warcraft* и *Orions: Legend of Wizards*. Две данные игры имеют различный бюджет, имеют разную концепцию и свой ИИИ.

Hearthstone находится в высокой категории, является распространённой игрой по всему миру. По своей сути, игра больше направлена на дружеские матчи между двумя игроками, но также имеет свой ИИИ, который, в свою очередь, благодаря возможностям режима «Приключения», может выступать за различных персонажей с различными тактиками игры. Обычно, чтобы игрок мог победить каждого из таких персонажей, ему самому надо выбрать соответствующую тактику. Чаще всего в режиме «Приключения» разработчиками применяется обманный искусственный интеллект. Обманный ИИ может использоваться в случаях, где нужен творческий подход человека и подразумевается, что ИИ имеет стартовое преимущество (или преимущество со временем) над игроком.

Orions относится к более низкой категории игр, в которых бюджет крайне мал. Имеет другую концепцию, и тактика ИИИ во всей игре является однотипной. В отличие от *Hearthstone*, игра *Orions* направлена на одиночную игру против ИИ. Поэтому, в ней реализована возможность выбора сложности с тремя уровнями ИИ для более интересной игры.

Построение ИИИ в существующих коллекционных карточных играх не целесообразно, но разработка алгоритмов принятия решения в играх, схожих с приведенными выше, и с возможностью их внедрения в дальнейшем, является актуальной.

Для реализации коллекционной карточной игры и построения для нее ИИИ, была составлена таблица различий двух ранее приведенных аналогов. В таблице 1 проведен сравнительный анализ рассматриваемых игр в жанре ККИ, результат которого будет полезен при создании игры и реализации для нее ИИ.

Таблица 1 – Сравнительный анализ реализации игр в жанре ККИ Hearthstone и Orions

Свойство	Hearthstone: Heroes of Warcraft	Orions: Legend of Wizards
Карта тратится при ее использовании	Да	Нет
Карты имеют активные способности	Нет	Да
Карты существ имеют точное расположение на поле боя	Нет	Да
ИИ имеет уровень сложности	Нет	Да
Количество применяемых карт за ход	Ограничено количество маны	Ограниче но одной картой

Можно использовать значения свойств из столбцы Orions, при этом заменить лишь значение количества применяемых карт за ход на две. Это и будет основная концепция создаваемой игры. Игрок может применить за ход две карты и у многих карт будет свойство («Серия приемов» в Hearthstone), которое будет определять, какой по счету была применена карта за ход (первой или второй) и, в зависимости от порядка применения, карты будут иметь разные характеристики.

Основная цель создания ИИИИ для данной игры: возможность выбирать уровень сложности в большом диапазоне, например, с применением шкалы от 1 до 10.

Разработка алгоритмов работы ИИИИ на основе генетического алгоритма. Генетический алгоритм (ГА) – это алгоритм поиска решения определённой задачи, основанный на эвристике. ГА потребуются лишь в его базовой основе и для достижения поставленных целей от него будет использоваться лишь ключевой состав.

Составляющие ГА необходимые для построения ИИИИ разрабатываемой коллекционной карточной игры:

- определение фитнес функции – это целевая функция, по мере возрастания которой и достигается цель;
- генотип – некоторый набор генов, который в своем базовом виде представляет набор коэффициентов, но в нашем случае каждый коэффициент может выступать другой функцией, что повлечет древовидную структуру генов (более точное определение будет приведено далее);
- определение популяций – набор решений, которые будут формироваться на каждой эпохе;
- операторы мутации, скрещивания и селекции – нужны для создания новых популяций;

- штрафная функция – функция условия выживания генотипа.

Далее для каждого из пунктов нужно определять значения и функции. Главная часть фитнес функции будет состоять, согласно поставленному условию, из двух генов, соответствующих первой и второй картам, которые разыгрываются в раунде. Именно эту комбинацию ИИИ применит в конце всех операций.

Была выделена главная часть фитнес функции, так как будет существовать и второстепенная часть. Пусть S – количество подконтрольных игроку слотов для карт. Тогда по формуле (1) фитнес функция имеет вид:

$$FF = \sum_{i=1}^2 C_i^k + \sum_{j=1}^s A_j^k \rightarrow \max, \quad (1)$$

где C_i^k – функция эффективности карты, разыгрываемой под номером i , и находящейся в общей базе карт игрока под номером k ;

A_j^k – функция эффективности использования способности карты в слоте под номером j (если карта не имеет способности или соответствующий слот пустой, то коэффициент равен нулю) и находящейся в общей базе карт игрока под номером k .

В формуле (1) после знака плюс находится второстепенная часть фитнес функции. Она определяет эффективность использования умений существ в той или иной ситуации. В некоторых случаях две части функции будут влиять друг на друга.

Было сказано, что гены в этом случае имеют древовидный вид, так как функция должна иметь свои аргументы. В аргументы функции эффективности карты будут переданы такие параметры, как:

- если это карта существа: цена, здоровье, атака, наличие активных способностей, наличие пассивных способностей, слот постановки и др.;
- если это карта заклинания: цена, урон/исцеление, цель на которую используется заклинание, другие эффекты, оставляемые после использования заклинания и др.

Каждая карта индивидуальна и имеет свою функцию эффективности и свои, необходимые для ее вычисления, аргументы. Важной особенностью является и синергия некоторых карт. Под синергией в данном случае подразумевается улучшение эффективности одной карты в присутствии или за счет другой. Поэтому, в некоторых картах в аргументы при вычислении фитнес функции передаются не только текущая обстановка на поле боя, но и другая карта, которая будет разыграна, если данный вариант «победит» в отборе.

Аргументы функции эффективности использования способностей также гибкие, и каждой способности нужен свой набор для ее последующего вычисления. Чаще всего, он, помимо перечисленных параметров, будет включать

в себя общий запас маны игрока, так как для их активации требуется некоторое ее количество.

Согласно базовому ГА необходимо создать начальную популяцию.

Для создания новых популяций требуется на каждой эпохе находить некоторое количество генотипов, которые останутся «в живых». Для этого будут оценены все генотипы и выживут только $n/3$ из их общего количества, у которых фитнес функция имела наибольшее значение. Вторая треть должна формироваться путем скрещивания выживших генотипов. Два родителя выбираются случайным образом, и методом случайной маски формируется потомок. Последняя треть формируется случайным образом, как это было при составлении начальной популяции.

Также надо учесть оператор мутации. Его работа в данном алгоритме не сводится к изменению генов согласно некоторому условию. При возникновении мутации у гена с функцией эффективности мутирует один из возможных аргументов функции. Самые явные аргументы, которые могут мутировать – слот, в который будет поставлена карта или же цель направленного заклинания. Если мутация выпала на функцию эффективности использования способности, то значение меняется на противоположное и выступает логической переменной так как способность может либо примениться, либо нет. Стоит предварительно учесть, что если мутация возникла в аргументах функции эффективности карты, то изначальный генотип так и останется без изменений, а новый генотип появится как потомок. Один коэффициент аргумента может уменьшить функцию эффективности настолько, что она не пройдет селекцию в следующей эпохе, хотя предложенный изначальный вариант может быть одним из лучших. По указанному выше принципу формируются эпохи до тех пор, пока не выполнится определенное условие. Так как существует задача построения ИИИ и его сложность нужно регулировать в большом диапазоне, то логичнее всего будет при увеличении сложности ИИИ увеличивать количество итераций алгоритма, а также количество генотипов в каждой из эпох, что даст возможность найти лучшее решение.

Ранее было сказано, что генотип имеет древовидную структуру. Пример некоторого случайного генотипа X приведен на рисунке 1.

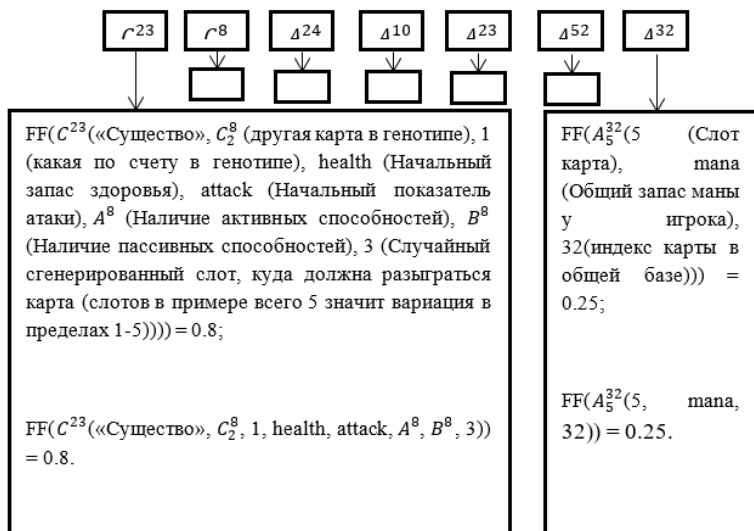


Рис. 1 – Случайно сгенерированный древовидный состав генотипа

Согласно условию, которое было установлено в определении скрещивания генотипов, было сказано, что создание потомка произойдет по схеме случайной маски. Но если реализовать именно так, то возникает проблема, которая заключается в том, что, возможно, на некоторой эпохе Y фитнес функции большинства из генотипов будут примерно (или точно) равны. Это обусловлено случайным скрещиванием генотипов, у которых на значение итоговой суммы фитнес функции показывают первые два гена, так как остальные попросту могут обнулиться. Чтобы обойти эту проблему требуется вмешаться в случайный отбор и добавить некоторое условие, которое при скрещивании двух родителей всегда будет выбирать первые два гена от разных родителей.

Последний пункт – штрафная функция. Описать в явном виде ее не получится из-за ее вариативности в разных вариантах реализации самой игры. Штрафная функция представляет собой систему условий, которые ограничивают генотипы, не удовлетворяющие некоторым условиям. Самое очевидное из условий – в аргументы функции эффективности карты существа не может поступить слот, который уже занят на поле боя или же карты заклинания, цель которой не существует.

Выводы. В качестве подхода построения игрового искусственного интеллекта коллекционной карточной игры был выбран генетический алгоритм, основной задачей которого является предоставление игроку возможности выбора сложности игры из большого диапазона.

При реализации некоторых других алгоритмов поведения искусственного интеллекта необходимо искусственно ухудшать полученный результат при расчетах комбинаций ходов или создавать множество ветвлений алгоритма для каждой из предложенных сложностей. Данный алгоритм позволяет не прибегать к таким приемам, но из-за этого необходимо для каждой из карт в базе данных представить функции эффективности.

При выполнении поиска решения анализируются не все ходы, а лишь те, что случайно были предложены и сгенерированы позднее. Сложность позволит корректировать критерий остановки генетического алгоритма и количество комбинаций на каждом шаге.

Также для достижения максимальной эффективности и вариативности игры с ИИ, помимо реализации выбора с помощью генетического алгоритма, можно добавлять, сколько необходимо дополнительных условий, которые непосредственно будут влиять на итоговое значение фитнес функции. Эти варианты могут быть реализованы как простыми ответвлениями, так и сложными функциями и могут также входить как в сам генотип, так и как дополнительный анализ решения. К примеру, в рассматриваемой абстрактной игре должна присутствовать некая функция зависимости, которая не относится к штрафной функции, а лишь анализирует поле боя и карты, которые будут доступны, если очки маны не будут затрачены.

Следует отметить, что рассматриваемый алгоритм берет за основу точное количество используемых карт (в данном случае 2) за ход при допуске варианта, когда вторая карта не будет сыграна вообще. Для переработки алгоритма для варианта, где за ход надо затратить весь запас маны, как в Hearthstone, требуется смена фитнес функции на вариант, где количество генов в каждом генотипе может быть разным, но принцип останется тот же: фитнес функция должна стремиться к максимальному значению при условиях системы штрафной функции.

Список литературы

1. Бухараев Т. Игровой искусственный интеллект [Электронный ресурс], 2006. – Режим доступа: https://ru.wikipedia.org/wiki/Игровой_искусственный_интеллект.
2. Гладков Л.А. Генетические алгоритмы: учебное пособие / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2006. – 320 с.
3. Емельянов В.В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик. – М.: Физматлит, 2003. – 432 с.
4. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Горячая линия-Телеком, 2008. – 452 с.

УДК 004.82

**Получение продукционных правил на основе ГП
Васяева Т.А., Струг С.С.**

Донецкий национальный технический университет, кафедра
автоматизированных систем управления
E-mail: www.strygok.ru@mail.ru

Аннотация

Васяева Т.А., Струг С.С. Получение продукционных правил на основе ГП. Рассмотрен эволюционный подход к извлечению знаний из набора данных. Предложен подход получения продукционной модели используя генетическое программирование.

Общая постановка проблемы. Продукционные модели - это наиболее распространенные на текущий день модели представления знаний, где знания описываются с помощью правил «если-то» (явление → реакция) и представляются в виде: ЕСЛИ условие (антецедент) ТО действие (консеквент)

Под условием понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под действием – набор действий, выполняемых при успешном исходе поиска. Внутри консеквента могут также генерироваться и добавляться в базу новые факты, которые были получены в результате вычислений или взаимодействия с пользователем.

При использовании таких моделей у систем, основанных на знаниях, имеется возможность:

- применение простого и точного механизма использования знаний;
- представления знаний с высокой однородностью, описываемых по единому синтаксису.

Эти две отличительные черты и определили широкое распространение методов представления знаний правилами.

Но вот получение таких правил является одной из наиболее трудоемких задач.

Целью данной работы является разработать математический аппарат для получения продукционных правил.

Применение генетического программирования для решения задач.

Для решения поставленной задачи предложено использовать генетическое программирование (ГП).

Решение задачи на основе ГП реализуется следующей последовательностью действий.

1. Установка параметров эволюции;
2. Инициализация начальной популяции;

3. $T:=0$;
4. Оценка особей, входящих в популяцию;
5. $T:=T+1$;
6. Отбор родителей;
7. Создание потомков выбранных пар родителей – выполнение оператор кроссинговера;
8. Мутация новых особей;
9. Расширение популяции новыми порожденными особями;
10. Сокращение расширенной популяции до исходного размера;
11. Если критерий останова алгоритма выполнен, то выбор лучшей особи в конечной популяции – результат работы алгоритма. Иначе переход на шаг 4.

Для решения задачи с помощью ГП необходимо выполнить предварительные этапы:

- определить терминальное множество;
- определить функциональное множество;
- определить фитнес-функцию;
- определить значения параметров, такие как мощность популяции, максимальный размер особи, вероятности кроссинговера и мутации, способ отбора родителей, критерий окончания эволюции и т.п.

После этого можно разрабатывать непосредственно сам эволюционный алгоритм, реализующий ГП для конкретной задачи.

Разработка метода. Каждое потенциальное решение в нашем случае будет представляться деревом, состоящим из функций, которые являются внутренними узлами деревьев, и терминалов, которые формируют листья деревьев.

Основная идея данного метода заключается в методе кодирования особей для генетического программирования. Каждая особь представляет собой дерево, которое соответствует синтаксическому выражению, представляющее множество правил в дизъюнктивной нормальной форме.

На рисунке 1 представлен пример дерева в дизъюнктивной нормальной форме. Дерево представлено двумя правилами. Данное представление особи значительно упрощает интерпретацию результата. В рассмотренном примере расшифровка будет следующей:

ЕСЛИ правило 1 ИЛИ правило 2 ТО результат 1, ИНАЧЕ результат 2.

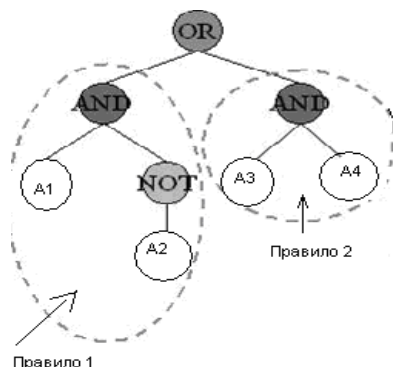


Рис. 1. Пример дерева в дизъюнктивной нормальной форме.

Определим терминальное множество.

Данные должны быть предварительно обработаны, основное назначение предобработки преобразовать входное обучающее множество в булевы переменные.

Функциональное множество состоит из логических операций: AND, OR, NOT.

Генерация начальной популяции. На данном этапе происходит генерация начальной популяции, в соответствии с заданными параметрами. Популяция состоит из набора деревьев, сгенерированных случайным образом. Генерация каждого дерева происходит рекурсивно, начиная с генерации первым функционального узла ИЛИ и его аргументов. В качестве аргументов на первом шаге может быть только узел ИЛИ. Далее для каждого дочернего узла случайным образом определяется тип и значения его аргументов по следующим принципам:

- после узла ИЛИ может быть только функциональный узел (значениями которого могут быть – ИЛИ или И);
- после узла И может быть функциональный узел (значениями которого могут быть – И или НЕ) или терминальные узлы;
- после узла НЕ может быть только терминальный узел.

Процесс выполняется по левой ветви до тех пор, пока не будет выбран дочерним терминальный узел. Затем генерируются правые ветви.

Вероятность функционального и терминального узлов меняется по следующему принципу: чем ниже вершина, тем больше вероятность терминального узла и меньше функционального. Для функционального узла на каждом последующем шаге увеличивается вероятность узла И и уменьшается вероятность узла ИЛИ.

При формировании дерева в одной ветви ИЛИ (т.е. для одного правила) не используется один и тот же терминальный символ более одного раза.

Предусмотрены методы создания деревьев: полный, растущий и комбинированный.

Применение генетических операций:

Отбор родителей. Предложено использовать отбор пропорционально значению целевой функции реализованный методом рулетки или турниром. При этом если два или более потомка имеют одинаковую фитнес-функцию, то выбирается дерево минимальной сложности.

Кроссинговер. Для древообразной формы представления используются следующие три основных оператора кроссинговера:

- узловой кроссинговер;
- кроссинговер поддеревьев;
- смешанный.

Учитывая строго определенное представление дерева необходимо модифицировать операторы кроссинговера. Модификация заключается в выполнении оператора кроссинговера для худшего правила и в поиске оптимальной точки разрыва.

Мутация. Для деревьев используются следующие операторы мутации:

- узловая;
- усекающая;
- растущая.

Как и в случае с оператором кроссинговера оператор мутации должен быть модифицирован. Модификация заключается в определении вероятности мутации в соответствии с ошибкой обучения.

Редукция. Предлагается использовать элитную стратегию.

Критерий останова. В качестве критерия останова можно выбирать указание определенного числа итераций или указание определенного числа повторения лучшего результата.

Выводы. Разработан подход построения продукционной модели на основе ГП, данный подход позволяет извлекать знания из обучающего набора данных. Данный подход применим для любой предметной области и может быть использован для построения базы знаний в экспертных системах.

Список литературы

1. _Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning.-Addison-Wesley, reading,MA.-1989.
2. Koza J.R. Genetic Programming. Cambridge:MA:MIT Press,1992.
3. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И.Д. Рудинского. - М.: Горячая линия – Телеком, 2006. – 452 с. : ил.

УДК 519.173

**Применение различных алгоритмов в решении задачи Штейнера
Нечепуренко М.С., Ольшевский А.И.**

Донецкий национальный технический университет
Кафедра искусственного интеллекта и системного анализа
E-mail: nechepurenko@mail.ua

Аннотация

Нечепуренко М.С., Ольшевский А.И. Применение различных алгоритмов в решении задачи Штейнера. Рассмотрены методы построения деревьев Штейнера. Определены особенности их построения в трёхмерном пространстве, реализован программный модуль построения дерева Штейнера, описаны особенности реализации.

Общая постановка проблемы. Актуальность данной задачи связана с бурным ростом компьютерных сетей, а также возможности применения в других областях науки и техники, таких как прокладка силовых кабелей в зданиях и сооружениях. Задача Штейнера является NP-полной. Решения 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.

Исследования. 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 , который является деревом, покрывающим все терминальные вершины, и имеет минимальную длину среди всех подобных подграфов. Этот подграф вновь называется МДШ.

Для определения расстояния между вершинами используется формула 1 – формула для вычисления расстояний между точками в прямоугольной метрике.

$$d = (p, q) = \sum (p_i - q_i), \quad (1)$$

где d – расстояние между точками;

p_i, q_i – координаты этих точек.

Таким образом, веса рёбер создаваемого графа будут равняться расстоянию между точками. Во время построения дерева рёбра могут

пересекаться. В этом случае обычно необходимо в месте пересечения добавить точку Штейнера и провести перерасчёт дерева для уменьшения его длины.

Алгоритм Крускала — эффективный алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа. Также алгоритм используется для нахождения некоторых приближений для задачи Штейнера.

Существенное отличие только что сформулированной задачи от задачи Штейнера (в ее графовой постановке) состоит в том, что в новой задаче множество вершин при поиске наименьшего остовного дерева вес каждого ребра не меняется. Поэтому алгоритм Крускала в его стандартном виде дает лишь некоторое приближение решения задачи Штейнера.

Рассмотрим алгоритм нахождения остовного дерева наименьшего веса.

Отсортируем все ребра исходного графа по возрастанию весов и сформируем из них очередь так, чтобы в "голове" очереди находилось ребро с наименьшим весом, а в "хвосте" — с наибольшим и веса ребер не убывали от "голова" очереди к "хвосту".

Метод состоит в "сшивании" искомого дерева из компонент остовного леса. Первоначально остовный лес представляет собой множество изолированных вершин исходного графа, т.е. его множество ребер пусто. На первом шаге из очереди извлекается ребро наименьшего веса и добавляется к множеству ребер исходного дерева.

На последующих шагах алгоритма из очереди извлекается по одному ребру. Если это ребро соединяет вершины, принадлежащие разным компонентам текущего остовного леса, то оно добавляется к текущему множеству ребер искомого дерева, а указанные компоненты сливаются в одну. Иначе ребро отбрасывается. Процесс повторяется до тех пор, пока число компонент остовного леса не окажется равным 1.

На рисунке 1 отображено пошаговое выполнение алгоритма Крускала.

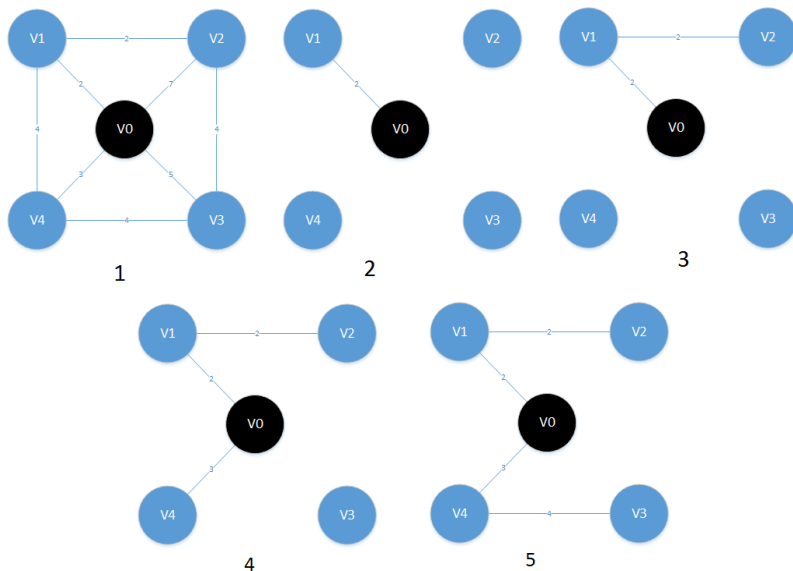


Рис. 1 – Алгоритм Крускала

Алгоритм Прима имеет очень простой вид. Искомый минимальный остов строится постепенно, добавлением в него рёбер по одному. Изначально остов полагается состоящим из единственной вершины (её можно выбрать произвольно). Затем выбирается ребро минимального веса, исходящее из этой вершины, и добавляется в минимальный остов. После этого остов содержит уже две вершины, и теперь ищется и добавляется ребро минимального веса, имеющее один конец в одной из двух выбранных вершин, а другой — наоборот, во всех остальных, кроме этих двух. И так далее, т.е. всякий раз ищется минимальное по весу ребро, один конец которого — уже взятая в остов вершина, а другой конец — ещё не взятая, и это ребро добавляется в остов. Этот процесс повторяется до тех пор, пока остов не станет содержать все вершины (или, что то же самое, $n-1$ ребро).

В итоге будет построен остов, являющийся минимальным. Если граф был изначально не связан, то остов найден не будет (количество выбранных рёбер останется меньше $n-1$).

Время работы алгоритма существенно зависит от того, каким образом мы производим поиск очередного минимального ребра среди подходящих рёбер.

Тривиальная реализация: алгоритмы за $O(nm)$ и $O(n^2 + m \log n)$.

На рисунке 2 отображено пошаговое выполнение алгоритма Прима.

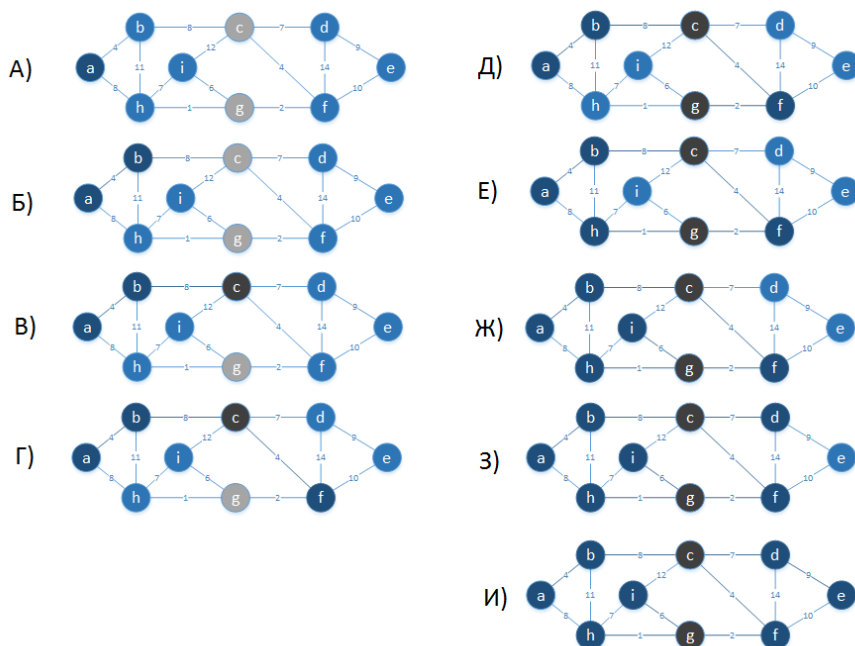


Рис. 2 – Алгоритм Прима

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.

Алгоритм использует кодирование деревьев-кандидатов, в которых символы из двоичного и недвоичного алфавитов занимают альтернативные положения.

Кодирование хромосомы осуществляется следующим образом. На n точках существует $2^{n-1}n^{n-2}$ прямолинейных деревьев Штайнера. Это означает следующее: $n-1$ двоичный символ, чередующийся с $n-2$ символами, выбранными из алфавита из n символов.

Строка, которая кодирует ПДШ, включает двоичные символы. Эти символы определяют для каждого ребра покрывающего дерева ту точку из двух возможных, которая включена в дерево точки значение 0 указывает на то, что это левая точка Штайнера, а значение 1 – правая точка Штайнера. Первый двоичный символ указывает ту точку Штайнера из первого ребра, которая определяется хромосомой. Последний двоичный символ, являющийся последним символом в

хромосоме, указывает на ту точку Штайнера, которая ассоциирована с ребром покрывающего дерева, которое выбирается алгоритмом в последнюю очередь.

Предлагаются два принципа оператора кроссинговера для этой задачи: потомки должны содержать структурные особенности (ребра, точки Штейнера) их обоих родителей; потомки должны сохранять любые ребра, являющиеся общими для двух родителей. Результирующий оператор кроссинговера генерирует одного потомка из двух родителей. Оператор копирует в потомка все пары прямолинейных ребер, соответствующих ребрам покрывающего дерева, которые делят между собой родители. Если ребра покрывающего дерева различаются только лишь в одной точке Штайнера, то оператор включает эти ребра покрывающего дерева и выбирает точку Штейнера произвольно. Затем выбирает пары прямолинейных ребер произвольно из двух родителей до тех пор, пока он не закончит формирование потомка ПДШ. Оператор отбрасывает любые пары ребер, которые создают циклы в потомках. Так как каждый родитель кодирует ПДШ, то для построения потомка ПДШ всегда берутся ребра их двух родителей. Нет необходимости вставлять произвольные ребра.

Оператор мутации модифицирует каждую позицию независимо с вероятностью в один процент. Затем мутация изменяет символ, новое значение которого выбирается из отличных от старого. Алгоритм применяет два оператора раздельно и каждый получает одного потомка.

Программная реализация. Для демонстрации выбран алгоритм Крускала, так как он достаточно прост, не привязан к выбору первоначального остова, в отличие от алгоритма Прима.

Для графического отображения дерева Штайнера использоваться графическая библиотека OpenGL. Данная библиотека позволяет быстро и просто реализовать вывод точек в виде сфер в пространстве, а также связи между ними. Разработки интерфейса выполняется с использованием библиотеки MFC. Сцена OpenGL будет интегрироваться в рабочую область программы, сохраняя привычный для пользователя вид окна.

Для хранения объектов сцены будет использоваться такая структура как стек. Стек – структура данных, представляющая собой список элементов, организованных по принципу LIFO. Координаты объекта будут храниться в двумерном массиве с тремя столбцами, где каждый из столбцов хранит соответствующую координату в пространстве. Каждый объект стека обладает, уникальным идентификатором объекта, ключом, указывающий на тип объекта, двумерным массивом, хранящем в себя координаты местоположения объекта, указателем на следующий элемент стека.

Для построения дерева Штейнера по алгоритму Крускала используется вектор объектов edges, который состоит из следующих элементов:

- x – начальная вершина;
- y – конечная вершина;

- w – вес ребра;
- $edges \{ \}$ – рёбра, вошедшие в дерево;
- $edges(x, y, w): x(x), y(y), w(w) \{ \}$.

Алгоритм Краскала находит минимальное остовное дерево. На вход алгоритм Краскала принимает неориентированный граф, который задается с помощью структуры $edge: x$ – из какой вершины, y – в какую, w – вес ребра. После того, как считали граф, мы должны отсортировать все ребра по возрастанию. Далее нужно рассматривать ребра по порядку, решая брать или не брать ребро. Мы должны взять ребро, если оно соединяет вершины, которые расположены в разных подмножествах, и не брать, если в одном. Таким образом, пробежавшись по всем ребрам и взяв только те, которые подходят, мы построим минимальное остовное дерево.

Выводы. Были определены цели решения задачи, построено минимальное дерево Штейнера. Рассмотрены основные алгоритмы построения и решения. Проанализирована эффективность этих алгоритмов. Разработан программный продукт выполняющий построение дерева Штейнера в трёхмерном пространстве.

Список литературы

1. А. В. Панюков “Топологические методы решения задачи штейнера на графе”, Автомат. и телемех., 2004, № 3, 89–99; A. V. Panyukov, “The steiner problem in graphs: topological methods of solution”, Autom. Remote Control, 65:3 (2004), 439–448
2. V.V. Vazirani, Approximation algorithms, Berlin, Springer, 2003.
3. Bang Ye Wu, Kun-Mao Chao, Steiner Minimal Trees[Online]. Available: <http://www.csie.ntu.edu.tw/~kmchao/tree07spr/Steiner.pdf> [2010, March 18]
4. F.K. Hwang, D.S. Richards and P. Winter, The Steiner tree problem, North-Holland, New York, 1992.
5. G. Robins and A. Zelikovsky, Improved Steiner Tree Approximation in Graphs[Online].

УДК 004.942

Нахождение контура фрагментов лица на мимическом изображении Потёмкин А.С.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: liho21@gmail.com

Аннотация

Потёмкин А.С. Нахождение контура фрагментов лица на мимическом изображении. В работе рассмотрены некоторые положения процесса распознавания эмоций на мимических изображениях. Подробно рассмотрен этап нахождения основных фрагментов лица, таких как: губы, нос, брови, глаза. То есть, фотографические изображения, которые отвечают определенным эмоциональным состояниям, превращаются в векторы значений, содержащих координаты контуров основных фрагментов лица.

***Ключевые слова:** мимическое изображение, базовые эмоции, выделение фрагментов лица, анализ и синтез.*

Общая постановка проблемы. Лицо несет в себе информацию о душевном состоянии и переживаниях человека. Источником этой информации является многочисленные мышцы на лице. В [1] автор, изучивши анатомию человеческой особы, определил какие именно мышцы и каким образом участвуют в определенных экспрессивных изменениях лица. Однако человек не различает движение мышц, он различает изменения более высокого уровня – движение фрагментов лица таких как: лоб, брови, глаза, нос, щеки, губы, подбородок. Определенный набор таких изменений определяется человеком как эмоции. В [2, 3] автор установил, что существуют семь основных выражений лица – конфигураций (схем) мимики, которые воспроизводят семь эмоций: счастье, удивление, страх, страдания, гнев, отвращение (презрение) и интерес. Все остальные эмоции – это смесь базовых.

Исследование. В отличие от человека, компьютеру понять какая эмоция присутствует на изображении не так просто. Одним из этапов анализа эмоций на лице является нахождение контура основных фрагментов лица таких как: рот, нос, брови, глаза, морщины. Ниже приведены методы нахождения контуров наиболее важных фрагментов лица для анализа.

Построение контура губ. Крайние точки рта можно найти программным методом, но существует модели голов, у которых распознать их с необходимой точностью достаточно сложно. Сделаем это вручную, что не требует много времени и не будет давать значительных ошибок. Обозначим их соответственно (x_n, y_n) – крайняя правая и (x_l, y_l) – крайняя левая точки.

Для построения линии, которая проходит между губами, движение начнем с крайней правой точки губ (x_n, y_n) в направлении увеличения абсциссы x с шагом δ .

$$(x_{k(c)}, y_{k(c)}) - \text{точка на искомой линии, } x_{k(c)} = x_n + k\delta.$$

Величину $y_{k(c)}$ определим так:

$$y_{k(c)} = y_{k-1(c)} + i\lambda, \quad \text{где } i = \arg \min_{l=-1,1} z''_{yy}(x_{k(c)}, y_{k-1(c)} + l\lambda). \quad (1)$$

Для построения линии, которая ограничивает верхнюю губу, движение начнем с крайней правой точки губ (x_n, y_n) в направлении увеличения абсциссы x с шагом δ . $(x_{k(b)}, y_{k(b)})$ – точка на искомой линии, $x_{k(b)} = x_n + k\delta$. Величину $y_{k(b)}$ определим так:

$$y_{k(b)} = y_{k(c)} + i\lambda, \quad i = \arg \max_{\substack{l \in \mathbb{Z} \\ l > 0}} z''_{yy}(x_k, y_{k(c)} + l\lambda). \quad (2)$$

Для построения линии, которая ограничивает нижнюю губу, воспользуемся методом для построения верхней губы с отличием в определении i :

$$i = \arg \max_{\substack{l \in \mathbb{Z} \\ l < 0}} z''_{yy}(x_k, y_{k(c)} + l\lambda). \quad (3)$$

Нахождение контура носа. Для нахождения контура носа используем несколько особенностей, которые отличают его среди всех частей лица:

со всех точек лица наивысшая точка по z находится на носу;

при движении вдоль профиля от рта к носу мы наткнемся на резкий перепад высот, который будет означать начала профиля носа;

нос переходит в лоб, причем если отойти от профиля уменьшая или увеличивая абсциссу x , то будет наблюдаться уменьшение координаты z , но когда мы доберемся до лба, то координаты z или не будут уменьшаться, или это уменьшение будет незначительным.

Для нахождения начальной точки, возьмем точку, которая находится посередине рта $(\frac{x_n + x_l}{2}, \frac{y_n + y_l}{2})$. Заметим, что эта точка находится на профиле головы.

Для нахождения начала профиля носа начнем движение с начальной точки в направлении увеличения ординаты y , проверяя при этом изменения координаты z . Таким образом дойдем к некоторой точке, к окрестности которой, начнет стремительно увеличиваться координата z (то есть $z''_{yy} > z_{y21}$, где z_{y21} – некоторое пороговое значение, которое определяется на практике). Эта точка и будет означать начало профиля носа.

Для нахождения верхушки носа будем продолжать двигаться в направлении увеличения ординаты y пока $z'_y > 0$, то есть пока будет увеличиваться координата z . Искомая точка находится при условиях:

$$z'_y(x, y + \lambda) < 0, \quad z'_y(x, y - \lambda) > 0.$$

Для нахождения конца профиля носа воспользуемся свойством в) носа. Начнем исследовать склоны с профиля, которые определяются значением:

$$z'_x(x - \delta, y), z'_x(x + \delta, y). \quad (5)$$

Условия, характеризующие наличие склонов:

$$z'_x(x - \delta, y) > z_{x11}, z'_x(x + \delta, y) < z_{x11}, \quad (6)$$

где z_{x11} – некоторое пороговое значение $z'_x(x, y)$, которое определяет тангенс угла наклона; $z_{x11} = 0.5$. Двигаясь в направлении увеличения ординаты у достигнем точки для которой:

$$\begin{aligned} z'_x(x - \delta, y) > \varepsilon, z'_x(x + \delta, y) < \varepsilon, \\ z'_x(x - \delta, y + \lambda) > \varepsilon, z'_x(x + \delta, y + \lambda) < \varepsilon. \end{aligned} \quad (7)$$

Точка (x, y) и есть искомой. Обозначим ее $(x_{нл}, y_{нл})$.

На предыдущем шаге было определен профиль носа, используя склоны по абсциссе x . Теперь, используя их, определим границу носа. Начнем движение с точки $(x_{нл}, y_{нл})$, которая представляет из себя точку перехода носа в лоб, вдоль профиля в направлении уменьшения ординаты y с шагом δ . Рассмотрим k -тый шаг.

$$x_k = x_{нл}, y_k = y_{нл} - k\lambda, \quad (8)$$

где (x_k, y_k) – точка профиля.

Зафиксируем ординату y и начнем с шагом δ уменьшать абсциссу x , наблюдая за величиной $z'_x(x, y)$. Движение будем продолжать пока $z'_x(x, y)$ не станет меньше некоторого порогового значения z_{x12} ($z_{x12} = 0.6$). Так найдем некоторую точку $(x_k^{(n)}, y_k^{(n)})$, которая будет лежать на правой границе носа. Аналогично, увеличивая абсциссу x получим соответствующую точку $(x_k^{(n)}, y_k^{(n)})$ на левой границы носа. Совокупность точек $(x_k^{(n)}, y_k^{(n)})$, $(x_k^{(n)}, y_k^{(n)})$ будет составлять границу носа, а все точки, которые лежат в этой области будут формировать сам нос.

Построение линии бровей. Линии бровей начинаются с переносицы.

Эту точку было найдено при построении профиля носа и обозначено как точку перехода нос-лоб $(x_{нл}, y_{нл})$. Рассмотрим некоторые особенности бровей. Пусть имеем точку (x, y) на брови. Тогда при уменьшении ординаты y на λ происходит резкое уменьшение координаты z . Это условие запишем так:

$$z''_{yy}(x, y) < z_{y22}, z_{y22} = 1.5. \quad (9)$$

Начнем движение с точки $(x_{нл}, y_{нл})$ в направлении уменьшения абсциссы x по линии $y = y_{нл}$. Рассмотрим k -тый шаг.

Имеем точку (x_k, y_k) :

$$x_k = x_{нл} - k\delta, y_k = y_{нл}. \quad (10)$$

Рассмотрим линию $x = x_{нл}$. Будем двигаться вдоль нее по точкам $(x_k, y_k + l\lambda)$, пока не найдем такую точку $(x_k^{(n)}, y_k^{(n)})$, которая будет удовлетворять неравенство:

$$(x_k, y_k + l\lambda) < z_{y22}. \quad (11)$$

Движение в направлении уменьшения абсцисс x прекращаем, когда выполнится условие (12), то есть когда будет наблюдаться резкое уменьшение координаты z . Аналогично строится и левая бровь.

$$z''_{xx}(x_k, y_k) > z_{y23}, z_{y23} = -1.3. \quad (12)$$

Построение контура глаз. Рассмотрим построение правого глаза. Для распознавания глаза используем ранее найденную бровь. Для нахождения крайней правой точки глаза начнем движение от средней точки брови в направлении уменьшения абсциссы x с шагом δ .

Рассмотрим k -тый шаг.

Пусть (x_k, y_k) – точка брови. Зафиксируем абсциссу x и начнем двигаться в направлении уменьшения ординаты y :

$$x_k = x_{\text{нл}} - k\delta, y_{kl} = y_k - l\lambda, l = 1, 2, \dots \quad (13)$$

Точки контура глаза найдем с условий:

$$z''_{yy}(x_k, y_{kl_1}) > z_{y24}, z''_{yy}(x_k, y_{kl_2}) < 0, z''_{yy}(x_k, y_{kl_3}) > z_{y24}, \quad (14)$$

где $l_1 < l_2 < l_3$.

Эти условия определяют наличие вогнутостей на краях глаза и выпуклости посередине. Если возникнет ситуация, когда $l_3 - l_1 < 2$, то это неравенство будет определять крайнюю правую точку. Аналогично строим контур левого глаза.

Выводы. Полученные контуры фрагментов лица будут служить основой для дальнейшего анализа и синтеза эмоций на изображении. Предложенные методы в дальнейшем предполагается реализовать в программном продукте, который будет определять эмоциональное состояние по мимическим изображениям.

Список литературы

1. Изард К. Эмоции человека / Изард К.Е. – М.: Изд-во МГУ, 1980. – 439с.
2. Ekman P. Cross-cultural studies of facial expression / Ekman P. // Ekman P., Friesen W. Darwin and facial expression: A century of research in review. – N.Y., 1973. – P. 196-222.
3. Ekman P. Measuring facial movement / Ekman P., Friesen W // Environmental Psychology and nonverbal Behavior. – 1976. – V. 1 (1). – p.56-75. Режим доступа: <http://www.paulekman.com/wp-content/uploads/2009/02/Measuring-Facial-Movement.pdf>

УДК 004.932.2

**Исследование методов описания контура объекта на изображении
Сутковая М.И., Бочаров Д.М.**

ГОУВПО Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа

E-mail: d_m_b@list.ru

Аннотация

Сутковая М.И., Бочаров Д.М. Исследование методов описания контура объекта на изображении. Рассмотрены методы описания контуров объектов на изображении. Выделены их слабые и сильные стороны. Проанализированы достоинства и недостатки рассмотренных методов.

Общая постановка проблемы. Форма объекта наиболее часто описывается его контуром или набором параметров. Суть контурного представления заключается в том, что форма каждого объекта описывается множеством его граничных точек. Различные описания отличаются друг от друга способом представления этого множества в памяти ЭВМ. В простейшем случае это множество задается в виде упорядоченного списка, где каждый элемент соответствует определенной граничной точке и представляет ее координаты. Одним из главных направлений исследований при создании систем компьютерного зрения является анализ и решение проблем математического описания контуров различных объектов на изображении и представление этой информации в форме, удобной для изменения, хранения и визуализации.

Метод цепного кодирования. В данном методе соседние точки соединяются между собой прямолинейными отрезками в горизонтальном, вертикальном или диагональном направлении. Способ кодирования отрезка в зависимости от направления движения к соседней точке определяется по принципу, изображенному на рис. 1а. Кривая представляется начальной точкой и последовательностью восьмеричных цифр с начальной точкой $A(0, 1)$ (см.рис.1б).

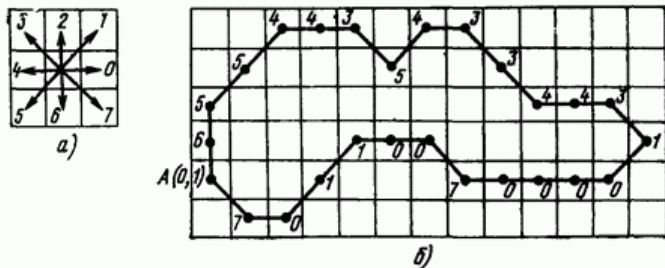


Рис. 1 – Цепное кодирование контуров фигур:
а) коды направлений движения по контуру;
б) пример кодирования контура

Код удобен тем, что допускает довольно простые процедуры вычисления метрических характеристик фигуры (площади, высоты, длины) и ее преобразования (поворота или перемещения на плоскости).

Цепное кодирование особенно удобно при сравнении формы двух кривых. Пусть существуют две цепи: $a=a_1...a_n$ и $b=b_1...b_n$. Цепная взаимно-корреляционная функция C_{ab} двух кривых вычисляется с помощью выражения

$$C_{ab} = \frac{1}{n} \sum_{i=1}^n \overline{a_i b_i}, \quad \text{где } \overline{a_i b_i} = \cos(\angle a_i - \angle b_i).$$

Если, если две кривые идентичны, их цепная функция взаимной корреляции достигает своего максимального значения, равного 1.

Итеративный подбор конечных точек. Более компактно замкнутая или незамкнутая кривая в памяти ЭВМ может быть представлена с помощью кусочно-линейной аппроксимации.

Итеративный подбор конечных точек является одним из простых методов кусочно-линейной аппроксимации.

Дано множество из N точек, представляющих собой множество точек контура. На первом этапе работы алгоритма конечные экспериментальные точки A и B соединяются прямой линией. Затем вычисляются расстояния от каждой точки до этой линии, и если все они меньше некоторого установленного заранее порога, процесс закончен. Если это не так, то находится точка (например C) с наибольшим отклонением от прямой AB и данная прямая заменяется двумя новыми отрезками AC и CB .

Данная процедура повторяется для каждого отрезка до тех пор, пока экспериментальные точки не будут «хорошо» аппроксимированы отрезками прямых.

Основное преимущество данного метода состоит в его простоте.

Недостаток метода заключается в том, что на результат аппроксимации сильно влияют отдельные точки. Например, единственная «дикая» точка (импульсный шум) может резко изменить конечный результат.

С этим недостатком можно бороться, например при помощи процесса предварительной фильтрации.

Выделение скелета объекта. Приведенные выше способы представления кривой могут применяться не только для описания множества граничных точек, но и для кодирования осевых скелетов, получаемых в результате применения процедуры преобразования к средним осям.

Цель такого преобразования заключается в том, чтобы выделить из исходного объекта его штриховое представление – скелет. Более того, преобразование выделяет также дополнительную информацию, которая вместе со скелетом позволяет восстановить исходный объект.

Пусть внутренняя часть объекта покрыта сухой травой, а его окрестность (фон) – негорючей сырой травой. Огонь возник одновременно во всех точках на границе объекта. Он будет распространяться с неизменной скоростью по направлению к центру объекта. Однако в некоторых точках линия продвижения огня от одной области границы будет пересекать фронт огня от какой-либо другой области, и эти два фронта будут гасить друг друга. Эти точки называются точками гашения огня, а множество точек гашения определяет скелет объекта.

Данная интерпретация понятия скелета дает возможность для построения более полного описания объекта. С каждой точкой x скелета объекта, имеющего границу B , сопоставляется величина $q(x)$, определяемая равенством

$$q(x)=d(x,B).$$

Функция $q(x)$ называется функцией гашения скелета. Проще говоря, это расстояние от данной точки скелета до границы объекта. Для каждого объекта, имеющего скелет S и функцию гашения q , пара (S,q) называется скелетной парой объекта. Любой исходный объект может быть восстановлен по его скелетной паре. Восстановление выполняется следующим образом: каждая точка x скелета становится центром диска с радиусом $q(x)$. Объединение всех таких дисков точно соответствует исходному объекту.

Прослеживание контуров (алгоритм жука). Метод заключается в последовательном вычерчивании границы между объектом и фоном. Пусть прослеживающая точка представляет собой жука, который ползает по изображению до тех пор, пока не наталкивается на темную область (объект). Тогда он поворачивает влево и движется по кривой, пока не выйдет за пределы объекта. После этого он начинает поворачивать вправо. Повторяя этот процесс, жук будет прослеживать границу объекта по часовой стрелке до тех пор, пока не попадет в окрестность начальной точки (см.рис.2).

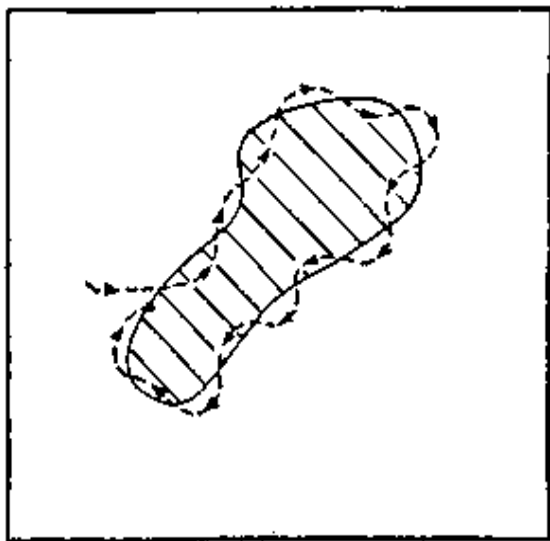


Рис. 2 – Пример прослеживания контура для объекта

Радиус кривизны траектории жука определяет «разрешающую способность» алгоритма прослеживания контуров.

Алгоритм жука не способен выделить острые изгибы контуров, кривизна которых сравнима с кривизной траектории. Кроме того, возникнут ошибочные точки перехода, если в силуэте объекта имеются «отверстия» рядом с самой границей.

Алгоритм жука для дискретного варианта может быть определен следующим образом:

- просматривается изображение до тех пор, пока не встретится элемент объекта;
- если жук находится на элементе объекта, он поворачивает налево и делает шаг;
- если жук находится на элементе фона, он поворачивает направо и делает шаг;
- жук прекращает продвижение, если он находится в пределах одного элемента от начальной точки.

Из рисунка 3 видно, что жук никогда не проникает в верхний элемент объекта.

Таким образом, последовательность переходных точек, выделяемых алгоритмом, зависит от правила первоначального просмотра изображения.

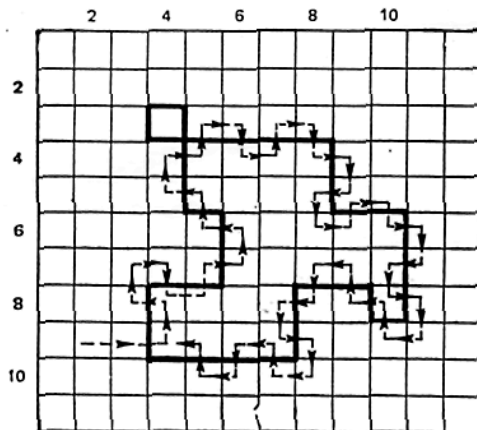


Рис. 3 – Пример прослеживания контура для объекта, заданного в дискретной форме

Выводы. В результате проведенного исследования был проведен обзор и анализ основных методов описания контуров объектов на изображении.

На основе данного анализа можно сделать выводы:

- цепное кодирование использует довольно простые процедуры вычисления метрических характеристик фигуры, что является особенно удобным при сравнении формы двух объектов;
- основное преимущество итеративного подбора конечных точек состоит в его простоте для реализации и скорости анализа изображения;
- любой исходный объект может быть восстановлен по его скелетной паре, для хранения которой требуется меньший объем информации;
- алгоритм жука позволяет управлять процессом прослеживания контурных точек объекта путем управления траекторией его движения.

Полученные результаты планируется использовать при выделении и классификации объектов на изображении, а также, представления этой информации в форме, удобной для изменения, хранения и визуализации в задачах распознавания образов.

Список литературы

1. Фильтрация и сегментация изображений для задач 3D-реконструкции / Д.М. Бочаров, Р.А. Сорокин // Информационные управляющие системы и компьютерный мониторинг: материалы V международной научно-технической конференции студентов, аспирантов и молодых ученых (Донецк, 22 – 23 апреля 2014 года) / Донецкий национальный технический университет. Донецк, 2014. – С. 398-404.
2. Выделение геометрических примитивов на фотоизображении / Р.А. Сорокин, Д.М. Бочаров // Инновационные перспективы Донбасса: материалы VI международной научно-практической конференции (Донецк, 20 – 22 мая 2015 года) / Донецкий национальный технический университет. Донецк, 2015. – Т.-5. Компьютерные науки и технологии. – С. 86-90.
3. Прэтт У. Цифровая обработка изображений. В 2-х книгах. Пер. с англ. - М.: Мир, 1982(78). - Кн.1- 312 с, Кн.2- 480 с.
4. Потапов А. А., Пахомов А. А., Никитин С. А., Гуляев Ю. В., Новейшие методы обработки изображений. – М.: Физматлит, 2008. – 496 с.
5. Цифровая обработка изображений в информационных системах / И. С. Грузман [и др.]. - Новосибирск : Изд-во НГТУ, 2002. - 352 с.
6. Исследование методов выравнивания яркости фотоснимков для задач обработки визуальной информации / М.И. Сутковая, Д.М. Бочаров // Компьютерная и программная инженерия – 2015: материалы международной научно-технической конференции студентов, аспирантов и молодых ученых (Донецк, 15 – 16 декабря 2015) / Донецкий национальный технический университет. Донецк, 2015. – С. 279-282.
7. Анализ влияния методов выравнивания яркости фотоснимков на качество выделения геометрических примитивов / А.В. Захаров, Р.А. Сорокин // Компьютерная и программная инженерия – 2015: материалы международной научно-технической конференции студентов, аспирантов и молодых ученых (Донецк, 15 – 16 декабря 2015) / Донецкий национальный технический университет. Донецк, 2015. - С. 274-279.

УДК 004.67

Применение искусственной нейронной сети для дешифрации лексикографических кодов расширенной реальности

Лащенко К.С., Гудаев О.А

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: lashenko.k.s@yandex.ru

Аннотация

Лащенко К.С., Гудаев О.А. Применение искусственной нейронной сети для дешифрации лексикографических кодов расширенной реальности. Рассмотрен способ кодирования маркеров расширенной реальности для распознавания системой расширенной реальности и отображения объектов. Определен метод дешифрации закодированных маркеров. Рассмотрена возможность применения нейронной сети с целочисленными коэффициентами в качестве декодера. Определена эффективность дешифрации нейронной сети в сравнении со стандартным алгоритмом сравнения.

Постановка проблемы. Требования к современным ЭВМ с каждым днем возрастают и трудно представить какую-либо сферу деятельности человека, в которой не использовались различные цифровые устройства, будь то суперкомпьютер, ноутбук или смартфон. Повсеместно, в той или иной степени, электроника помогает человеку выполнять различные задачи, и в соответствии с поставленными требованиями, возможности техники, порой, превосходят даже все самые оптимистичные ожидания. Примером тому служит изобретение технологии расширенной реальности. Данная технология позволяет существенно расширить рабочее пространство человека. В реальности, стол не может превратиться в стул, увеличить свои размеры или парить в воздухе. В компьютерной реальности, возможно все. Кроме того, объект расширенной реальности, являющимся, к примеру, золотой рыбкой, может в ту же секунду стать автомобилем. Такая технология предоставляет большие возможности во множестве областей деятельности человека, но и она не лишена недостатков. Любая новая технология имеет как преимущества, так и недостатки. В то же время, любая технология имеет огромный ресурс к совершенствованию.

Система расширенной реальности основана на распознавании определенных образов, получаемых из видеокамеры, так называемых маркеров. Каждый маркер, соответствует своему закодированному номеру. Если система обнаруживает на изображении в реальном времени маркер, который имеется в базе, то она выводит изображение полученной с камеры, но с изображением объекта, который был задан в соответствие полученному маркеру. Для данной

системы, база маркеров заранее определена. Ниже приведена последовательность действий системы при ее работе:

- получение кадра;
- поиск маркеров в кадре;
- дешифрация маркера;
- поиск маркера в базе;
- отображение объект на экране в месте нахождения маркера.

Маркер системы расширенной реальности, представляет собой бинарную матрицу в машинном представлении, визуально – это, матрица черно-белых точек определенного размера. После обнаружения маркера, система переводит его в бинарную последовательность, где, черная точка это 1, а белая – 0. Получив последовательность, начинается поиск такой же последовательности в базе и если найденный маркер в базе есть, то происходит получение его номера. Так как система отображает 3D-объекты в реальном времени, что означает свободное передвижение камеры вокруг области с маркерами и наблюдение за всеми сторонами объекта, то для отображения одного объекта используется минимум два маркера, чтобы по их взаимному расположению определить, какую сторону объекта следует отобразить на экране. Поиск по базе происходит сравнением каждого бита входного маркера с каждым битом каждого маркера, имеющего в базе. Поэтому скорость поиска маркера далека от идеала, из-за чего система довольно требовательна к ЭВМ. Для ускорения процесса поиска, было принято решение применить искусственную нейронную сеть с целочисленными коэффициентами, что может ускорить работу системы.

Исследования. Для проведения исследований, необходимо определить входные параметры для нейронной сети, ее архитектуру, метод обучения. Провести испытания и сравнить ее эффективность с имеющимся методом.

Входными данными для ИНС является бинарная последовательность, сформированная из маркера. Маркер представлен бинарной матрицей размером 6x6. При таком размере, бинарная последовательность состоит из 36 бит. Такой размер предоставляет большое пространство для формирования базы маркеров. Пример выбранного маркера представлен на рисунке 1.

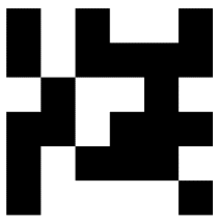


Рис. 1 – пример маркера размером 6x6

В процессе исследования, было сформировано 436 маркеров для обучающей выборки нейронной сети. Задача ИНС – получать на вход бинарную последовательность из 36 символов, выводить порядковый номер маркера, которому соответствует полученная последовательность. Для такой задачи, подходит двухслойная сеть с последовательными связями прямого распространения. Схематическая архитектура такой сети изображена на рисунке 2.

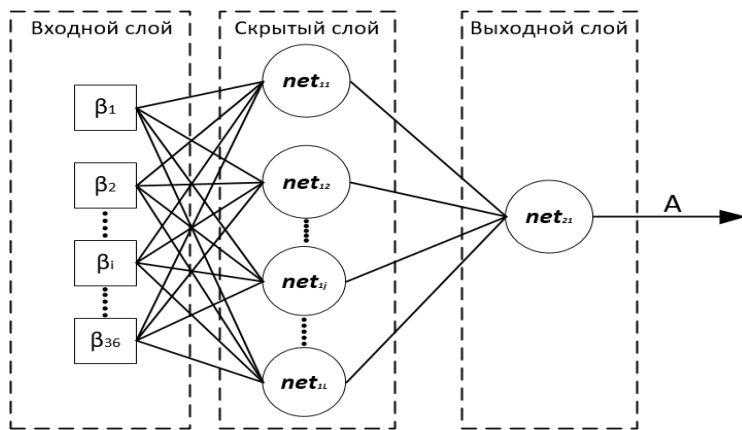


Рис. 2 – схематическая архитектура проектируемой ИНС

ИНС должна генерировать на выходе значение от 0 до 436. Для этого, в качестве активационной функции выходного слоя, подходит линейная передаточная функция, с областью определения $(-\infty; \infty)$, $f(s) = s$.

Функция активации нейронов скрытого слоя должна помогать настраивать коэффициенты нейронов и оперировать только целыми числами. Для этого, из существующих передаточных функций, подходит полулинейная (1), с областью определения $(0; \infty)$:

$$f(s) = \begin{cases} s, & s > 0 \\ 0, & s \leq 0 \end{cases} \quad (1)$$

Слой β используется для распределения сигналов входного слоя между нейронами скрытого слоя и в вычислениях не участвует.

Для оценки числа нейронов в скрытом слое однородных нейронных сетей используется формула (2) оценки необходимого числа синаптических весов – L_w .

$$\frac{m_2 * n}{1 + \log_2 n} \leq L_w \leq m_2 * \left(\frac{n}{m_2} + 1 \right) * (m * m_2 + 1) + m_2, \quad (2)$$

где m – размерность входного сигнала, m_2 – размерность выходного сигнала, n – число элементов обучающей выборки (количество входных слов).

Подставив в неравенство следующие параметры $m=36$, $m_2=1$, $n=436$, получим:

$$45 \leq L_w \leq 206702 \quad (3)$$

Таким образом, заданы рамки, в которых будет проводится подбор количества нейронов скрытого слоя. В качестве среды для проведения исследований, был выбран программный продукт MATLAB 7. В данной среде, все данные хранятся в виде массивов. Для обучения ИНС, была сформирована матрица из исходных данных обучающей выборки. Для представления матрицы использовался двумерный массив, первая строка которого содержит номера маркеров, а 36 ячеек каждого столбца занимает входное слово.

Для проведения исследований, был использован скрипт, код которого приведен ниже:

```
s=load('C:\InputData.mat'); p=s.ans(2:36+1,:); t=s.ans(1,:);  
PR=zeros(36,2); PR(:,2)=1;  
net=newff(PR,[45 1],{'poslin','purelin'},'traincgb'); % Создание ИНС  
net.trainParam.searchFcn='srchcha'; % Функция одномерной оптимизации  
net.trainParam.epochs=25000; % Число циклов обучения  
net.trainParam.show=10; % Показывать промежуточные циклы  
net.trainParam.goal=0.0057; % Ошибка обучения  
[net,tr]=train(net,p,t);
```

Данный скрипт, создает и проводит обучение ИНС по заданным параметрам. Исследования алгоритмов обучения показало, что наиболее подходит метод *Powell-Beale*, разновидность алгоритма сопряженных градиентов. Функция обучения ИНС методом *Powell-Beale* в MatLab называется *traincgb*. Вспомогательной для функции обучения ИНС использовалась функция одномерного поиска *srchcha*, реализующая разновидность метода оптимизации с применением кубической интерполяции.

Применяя разработанный скрипт, были проведены циклы обучения нейронной сети с различным количеством нейронов скрытого слоя. Так как наибольшее значение имеет количество выполняемых операций, то необходимо использовать минимально возможное количество нейронов в скрытом слое, но при этом ИНС должна выдавать правильный ответ. Чтобы получить лучшие результаты, необходимо найти сбалансированное количество нейронов скрытого слоя, при котором количество нейронов будет минимальным и ответ ИНС всегда будет правильным. Результаты обучения нейронной сети при различном количестве нейронов скрытого слоя приведены в таблице 1.

Таблица 1 – Результаты обучения ИНС

Количество нейронов	Ошибка обучения (MSE)	Максимальное разнотчение (maxD)	Количество ошибок дешифрации ИНС эталонов тезауруса (436 маркеров), после умножение на 1000 коэффициентов нейронов
43	0.0648049	1.5785	23
44	0.00569288	0.3832	Нет ошибок
45	0.00569025	0.3512	Нет ошибок
46	0.0130625	0.4402	Нет ошибок
47	0.0220102	1.1217	3

Исходя из полученных данных, лучше всего применить 44 нейрона в скрытом слое. При таком количестве, ИНС выдает правильный ответ. Кроме того, результаты полученные при обучении сети с 43 нейронами в скрытом слое показали, что дальнейшее понижение количества нейронов приводит появлению ошибок дешифрации. График динамики обучения нейронной сети представлен на рисунке 3.

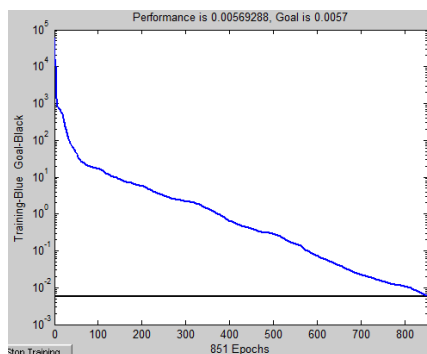


Рис. 3 – Динамика обучения ИНС

Выводы. Получив данные обучения, можно посчитать количество операций, выполняемых сетью для распознавания одного маркера. Каждый нейрон скрытого слоя выполняет 36 операций умножения (умножение каждого весового коэффициента на каждый входной сигнал), 35 операций сложения (суммирование всех входных сигналов после их умножения на весовые коэффициенты) и одна операция сравнения с нулем. Оптимальное количество нейронов скрытого слоя было выбрано 44. Общее количество операций, выполняемых скрытым слоем:

- 44 операции сравнения с нулем;
- 584 операции умножения;
- 1540 операций сложений.

Таким образом, выходной слой, в котором находится один нейрон, выполняет 44 операции умножения и 43 операции сложения. В итоге, для распознавания маркера, ИНС всегда будет выполнять 44 операции сравнения с нулем, 1628 операций умножения и 1583 операции сложения. В то время как стандартный алгоритм сравнения всегда ведет себя неоднозначно и очень зависит от того каким образом сформирована база маркеров и какой марке был получен на входе за одну единицу времени. Теоретически, если был получен зашифрованный маркер, который последний по порядку в базе, то будет выполнено $36 \cdot 436 = 15696$ операций сравнения, что практически в 5 раз больше чем выполнила бы нейронная сеть. На практике же, для тестирования стандартного алгоритма, были сформированы такие же бинарные последовательности, как и в случае с нейронной сетью но с помощью генератора случайных чисел. В большинстве случаев, количество операций сравнения незначительно отличалось от номера искомого маркера, что может свидетельствовать о том, что при правильно сформированных данных, алгоритм сравнения показывает себя намного эффективнее. В свою же очередь нейронная сеть не накладывает таких ограничений и количество выполняемых ею операций всегда будет одинаково. Кроме того, максимальное различие маркеров между собой, при использовании ИНС, может составлять всего 3 значения, что позволит использовать базы больших объемов.

Список литературы

1. Хайкин Саймон. Нейронные сети: полный курс. Второе издание / Саймон Хайкин.– Университет McMaster.– Москва. 2006. – 1015 с.
2. Новиков Ф.А. Дискретная математика для программистов/СПб. : Питер, 2000. – 304 с.
3. Фу К. Робототехника/ К. Фу, Р. Гонсалес, К. Ли; [пер. с англ.]. – М. : Мир, 1989. – С. 624.
4. Дьяконов В. Математические пакеты расширения MATLAB / В. Круглов // Специальный справочник. - СПб. : Питер, 2001. – С. 480.
5. Fiala, M. ARTag Revision 1, A Fiducial Marker System Using Digital Techniques.
6. [электронный ресурс]. – <http://www.iit-iti.nrc-cnrc.gc.ca/iit-publications-iti/docs/NRC-47419.pdf>.

УДК 004.8

**Исследование методов усовершенствования игры «Жизнь»
Савкин М.С., Савицкая И.В.**

Донецкий национальный технический университет
Кафедра искусственного интеллекта и системного анализа
E-mail: sawmaks@gmail.com

Аннотация

Савкин М.С., Савицкая И.В. Исследование методов усовершенствования игры «Жизнь». В статье рассмотрены правила игры «Жизнь» и ее возможные модификации. Проведен анализ существующих аналогов. Приведено описание возможностей модификации для превращения игры в стратегическую логическую игру для двоих.

Ключевые слова: игра «Жизнь», поле, клетка, поколение, эволюция, состояние.

Постановка проблемы. У созданной Джоном Конвеем игры «Жизнь» есть большой потенциал для дальнейшей модификации. Для выявления способов сделать игру более привлекательной для пользователей нужно проанализировать правила игры и ее возможные модификации, и на этой основе предложить способы ее преобразования.

Цель статьи – провести анализ игры и ее модификаций, привести обзор существующих аналогов и предложить способы изменения правил игры для ее усовершенствования.

Анализ правил игры «Жизнь». Игра «Жизнь» представляет собой клеточный автомат, ее изобрел английский математик Джон Конвей в 1970 году в попытке доказать возможность существования простых самовоспроизводящихся систем. Игра развивается на двумерной плоскости, разделенной на клетки. В компьютерной реализации плоскость ограничена и может быть замкнута, т.е. верхняя граница связана с нижней, а правая с левой. Каждая клетка может находиться в одном из двух состояний – живом или мертвом. Игра развивается в зависимости от начальной расстановки клеток (первое поколение), каждое следующее поколение зависит от положения клеток в предыдущем. У каждой клетки есть восемь соседей, состояние клеток определяется тремя основными правилами:

- живая клетка остается живой при наличии двух или трех живых соседей, иначе становится живой от «перенаселенности» либо «одиночества»;
- в мертвой клетке зарождается жизнь при наличии строго трех живых соседей, иначе клетка продолжает быть мертвой.

Критерием останова игры служит полное исчезновение живых клеток на поле, либо установление устойчивой (когда состояние всех клеток не меняется) или периодической (когда состояние повторяет ранее существовавшее) конфигурации.

Обзор модификаций. Возможны модификации игры в зависимости от следующих факторов: размерности, цветности, направлению алгоритма, констант эволюции, размеров игрового поля и т.д.

В зависимости от размерности поле может быть двумерным или многомерным.

В зависимости от цветности игра может быть одноцветной, черно-белой (живые клетки представлены двумя цветами) или полноцветной.

По направлению алгоритма игра может быть прямой и обратной.

Константы эволюции определяют количество живых соседей, необходимое для продолжения жизни живой клетки или зарождения жизни в мертвой. В классическом варианте клетка продолжает жить при наличии двух или трех живых соседей и зарождается при наличии трех.

Размеры игрового поля определяют ограниченность клеток на поле. Поле может быть ограниченным, неограниченным и полуограниченным.

Обзор существующих аналогов. Существует множество реализаций данной игры, все они отличаются своими функциональными возможностями и применяемыми модификациями алгоритма.

The Game Of Life – русскоязычная программа, предназначенная для компьютеров под управлением ОС Windows XP и более поздних версий, находится в свободном доступе, загрузить ее можно на сайте life.written.ru. Данная программа реализует классический алгоритм игры жизнь. Отличительной особенностью является богатый функционал. Данная программа очень удобна для отслеживания хода эволюции, поскольку имеет поле большого размера и инструменты для управления ходом времени в игре. Благодаря наличию множества инструментов для рисования новых фигур легко создавать свои собственные фигуры больших размеров за минимальное время. Также имеется возможность загружать наиболее интересные фигуры из заранее подготовленной базы фигур. Все это делает данную программу очень удобной для моделирования классического алгоритма игры «Жизнь».

Conway's Multiplayer Game of Life – браузерная реализация полноцветной модификации игры «Жизнь» на языке Javascript, доступна на сайте lifecompetes.com. Отличительной особенностью данной версии является поддержка многопользовательского режима игры. Здесь для всех игроков доступно одно общее поле, каждый игрок выбирает цвет фишек, которыми он будет играть выставляет их на поле. Таким образом все игроки, играющие одновременно, являются конкурентами. Если у игрока в распоряжении менее 12 клеток, то каждое шестое поколение ему начисляется одна дополнительная

клетка, которую он может поставить на поле. В игре присутствует список текущих лидеров (игроков с наибольшим количеством клеток в данный момент на поле) и список лидеров по максимальному количеству одновременно находящихся на поле клеток за все время игры.

Возможности усовершенствования алгоритма игры. Одним из способов сделать игру не просто средством моделирования и отслеживания эволюции, а инструментом для проверки на сообразительность и логическое мышление в постоянно меняющихся условиях является ее модификация. Необходимо видоизменить игру таким образом, чтобы появилась возможность стратегически мыслить. Для превращения игры «Жизнь» из просто наблюдательной в логическую игру, с возможностью участия в ней двух игроков (на подобии шахмат или шашек), наиболее подходящим был выбран метод внедрение черно-белой раскраски живых клеток. При этом, для увеличения продолжительности игры и спортивного азарта, нужно дать возможность игрокам влиять на ход игры посредством добавления фишек своего цвета на поле. Победителем в таком случае решено было считать игрока, который смог вытеснить все клетки оппонента с поля. С другой стороны, для предотвращения бесконечной игры и угасания интереса противников, можно считать победителем игрока, чья сумма клеток, по истечению определенного количества ходов, во всех поколениях будет максимальной. Эмпирическим путем было выявлено, что наиболее удобно играть, если размер поля будет ограничен до квадрата со стороной не более 20 клеток. В противном случае игрокам будет либо достаточно сложно влиять на ход игры и предсказывать состояние клеток в следующем ходу, либо будет недостаточно места для реализации своего стратегического потенциала. Кроме того, предусматривается возможность выбора вида границ поля (ограничены или циклично связаны).

С применением выше описанной технологии модификации предполагается создание режима игры для одного игрока, где нужно будет вытеснить с поля заранее выставленные клетки. При этом выставленные клетки будут просто эволюционировать по правилам игры, а игрок сможет влиять на ход игры путем добавления своих клеток на поле. В этом случае целью будет являться вытеснение чужих клеток за минимальное число ходов.

Также рассматривается вариант разработки режима для одного игрока, противостоять которому будет виртуальный оппонент. Реализация такого варианта игры предполагает создание искусственного интеллекта, который сможет достаточно хорошо противостоять человеку, т.е. обладать следующим набором качеств:

- делать максимально рациональные ходы;
- быть в достаточной мере непредсказуемым;
- быть не требовательным к ресурсам компьютера.

Таким образом, нужно разработать алгоритм принятия максимально выгодного решения, не основывающийся на полном переборе всех вариантов, с присутствием некоторого элемента случайности. Для создания такого алгоритма будет выгодно использовать генетические алгоритмы поиска решения. Также можно задать несколько выгодных ходов для начала игры (что-то на подобии дебюта), из которых алгоритм будет выбирать один случайным образом.

Для экономии ресурсов компьютера и увеличения быстродействия игры, также следует использовать алгоритм генерации следующего поколения не основанный на полной обработке всех клеток поля, а только тех которые могут изменить свое состояние в следующем поколении. Таким образом следует обрабатывать только живые клетки и соседние с ними мертвые. Пример работы модифицированного алгоритма представлен на рисунке 1.

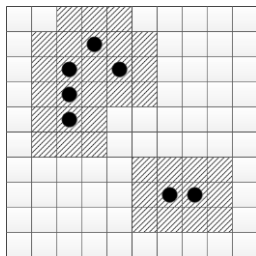


Рис. 1 – Пример выбора клеток для проверки на смену состояния в следующем поколении при использовании модифицированного алгоритма

Выводы. Был произведен анализ правил классической игры «Жизнь». Рассмотрены различные способы модификации игры. Рассмотрены существующие аналоги с разными модификациями. Были рассмотрены идеи для введения в игру стратегической и логической составляющих, для этого лучше всего подходит применение в игре черно-белой модификации. Были предложены способы реализации искусственного интеллекта для игры, наиболее подходящий для этого метод – применение генетических алгоритмов. Рассмотрены способы реализации основного алгоритма игры для оптимизации использования ресурсов компьютера.

Список литературы

1. Википедия. Свободная энциклопедия. Жизнь (игра) [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Жизнь_\(игра\)](https://ru.wikipedia.org/wiki/Жизнь_(игра)).
2. The Game of Life [Электронный ресурс]. – Режим доступа: <https://life.written.ru/>.
3. Conway's Multiplayer Game of Life [Электронный ресурс]. – Режим доступа: <http://lifecompetes.com/>.

УДК 004.932.2

Исследование методов классификации графических примитивов на фотоизображении

Захаров А.В., Сорокин Р.А.

ГОУВПО Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: eagle200877@mail.ru

Аннотация

Захаров А.В., Сорокин Р.А. Исследование методов классификации графических примитивов на фотоизображении. Определены основные типы графических примитивов. Рассмотрены и проанализированы методы определения опорных точек на фотоизображении для детектирования объектов и их классификации. Выделены достоинства и недостатки рассмотренных методов.

Общая постановка проблемы. Компьютерная обработка изображений предполагает не только обработку цифровых изображений при помощи ЭВМ, но и классификацию объектов, выполняемую при анализе этих изображений. В большинстве практических задач, распознавание образов (объектов) сводится к их идентификации или определению каких-либо свойств их графических примитивов.

В данной работе под графическим примитивом понимается простейший геометрический объект, отображаемый на экране дисплея. Основное назначение примитивов – обеспечить разработчиков и пользователей удобным набором средств для формирования геометрических объектов.

Исследование методов классификации геометрических примитивов.

Описание графического примитива обычно содержит метрическую и атрибутивную части.

Метрическая часть:

- величины, в которых задан графический примитив для отображения его на дисплее;
- величины, которые характеризуют его физическое или логическое представление.

Атрибутивная часть передает геометрические параметры, характеризующие форму и расположение графического примитива.

В основе построения любых графических элементов в современных графических библиотеках лежат три основных примитива (назовем их базовыми): точка, отрезок и треугольник. Хотя, в качестве примитива выступают иногда

прямоугольники, окружности, эллипсы, полигоны произвольных форм и др., на практике они могут быть описаны совокупностью базовых.

Классификация графических примитивов приведена в таблице 1.

В большинстве задач, решаемых в рамках компьютерного зрения, особое значение имеют точки, отрезки, треугольники, эллипсы и прямоугольники. Исходными данными в таких задачах служат монохромные изображения. Процесс получения геометрического описания пространства изображения в этом случае заключается в прохождении следующих этапов:

- ввод растрового цветного изображения;
- преобразование в полутоновое изображение;
- сглаживающая фильтрация (для устранения шумов);
- выделение краев определенными операторами;
- бинаризация;
- преобразование.

Таблица 1 - Классификация графических примитивов

Тип примитива	Описание
Точка	Простейший графический примитив, имеющий нулевую размерность. Точка характеризуется только координатами своего местоположения.
Отрезок	Совокупность точек, через которые проходит геометрический отрезок с заданными конечными точками. Характеризуется начальной и конечной точками, или начальной точкой и приращениями координат, или длиной и углом наклона.
Ломаная	Последовательности отрезков, соединяющих заданные точки.
Полигон	Область, ограниченная замкнутой ломанной.
Эллипс	Геометрическое место точек плоскости, для которых сумма расстояний до двух данных точек (фокусы) постоянна и больше расстояния между ними.
Треугольник	Частный случай полигона, ограниченный замкнутой ломанной, состоящей из трех отрезков.
Прямоугольник	Полигон, ограниченный замкнутой ломанной в форме четырехугольника, все углы которого прямые.

На этапе выделения краев первоначальной задачей является выделение особых (опорных) точек, на основе которых начинается поиск геометрических фигур сцены.

Особая точка сцены или точечная особенность – это такая точка сцены, изображение которой можно отличить от изображений всех соседних с ней точек сцены.

Для сравнения и описания точек можно использовать ее окрестность. Тогда предыдущее определение уточняется следующим образом. Под точечной особенностью понимается такая точка сцены M , лежащая на плоском участке

поверхности сцены, изображение окрестности которой можно отличить от изображений окрестностей всех других точек сцены N из некоторой другой окрестности этой точки $O(M)$.

Воспользовавшись последним определением можно дать определение точечной особенности изображения.

Точечная особенность изображения m - это такая точка изображения, окрестность которой $o(m)$ можно отличить от окрестности любой другой точки изображения $o(n)$ в некоторой другой окрестности особой точки.

Детектор Харриса. В последнее время было создано довольно много различных детекторов точечных особенностей изображений. Чаще всего используется детектор Харриса.

Для каждого пикселя (x_0, y_0) рассчитываем следующую матрицу:

$$M = \sum_{x,y \in S} \omega(x, y) \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix},$$

где $I(x,y)$ - яркость в точке (x,y) ;

$\omega(x,y)$ - весовая функция по окрестности $S(x_0,y_0)$, в качестве которой обычно берут распределение Гаусса.

У точки, окрестность которой похожа на угол, матрица будет иметь два больших положительных собственных значения. Таким образом, условие похожести точки на угол запишется следующим образом:

$$F = \min(\lambda_1, \lambda_2),$$

где λ_1, λ_2 - собственные значения матрицы.

Чтобы упростить вычисления, Харрис предложил отказаться от подсчета собственных значений матрицы, и вместо них ввел следующую функцию отклика угла:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2,$$

где k - параметр Харриса, его величина обычно лежит в пределах 0,04-0,15 и определяется эмпирически.

Далее с ней проводятся аналогичные действия - поиск локальных максимумов, и введение порогового значения.

Масочный детектор. Данный детектор основан на применении оценочного алгоритма соответствия маске того или иного фрагмента изображения. Для этого могут использоваться маски 3×3 , 5×5 , 7×7 и т.д. Сравнение с масками большего размера позволяет добиться более точных результатов, но отрицательно сказывается на быстродействии алгоритма.

Работа алгоритма заключается в следующем: для каждого пикселя изображения в соответствии с размером маски извлекается фрагмент и сравнивается непосредственно с маской.

Особый случай представляет собой проверка пикселей, находящихся на границе изображения. Для этого применяют следующий подход: вокруг изображения прорисовывается мнимый контур, который закрасен фоновым цветом и позволяет корректно детектировать углы на границы изображения.

Метод Виолы-Джонса. Метод Виолы-Джонса – алгоритм, позволяющий обнаруживать объекты на изображениях в реальном времени.

Признаки в данном методе используют суммирование пикселей из прямоугольных регионов. На рисунке 1 показано 4 различных типа признаков. Величина каждого признака вычисляется как сумма пикселей в белых прямоугольниках, из которой вычитается сумма пикселей в черных областях. Прямоугольные признаки чувствительны к вертикальным и горизонтальным особенностям изображений, результат их поиска груб. Однако, при хранении изображения в интегральном формате (в каждом пикселе изображения записана сумма всех пикселей левее и выше данного), проверка прямоугольного признака на конкретной позиции проводится за константное время, что является их преимуществом по сравнению с более точными вариантами. Каждая прямоугольная область в используемых признаках всегда смежна с другим прямоугольником, поэтому расчет признака с 2 прямоугольниками состоит из 6 обращений в интегральный массив, для признака с 3 прямоугольниками - из 8, с 4 прямоугольниками - из 9.

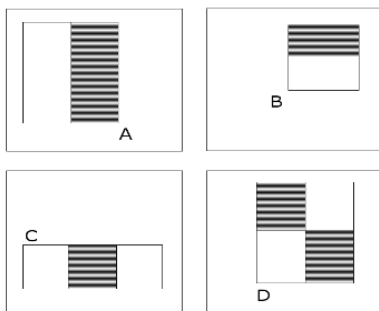


Рис.1 – Типы признаков

Высокая скорость расчета признака не компенсирует значительное количество различных возможных признаков. К примеру, при стандартном размере признака в 24x24 пикселя, возможно 162 тысячи разных признаков и расчет всех занимает продолжительное время.

Метод Далала-Триггса. Данный алгоритм демонстрирует один из лучших результатов среди известных алгоритмов детектирования объектов.

На первой стадии работы алгоритма формируются обучающие выборки из изображений «примитивов» и «не примитивов» разрешением 64×128 пикселей и вычисляются признаки.

Каждое изображение в обучающем наборе делится на ячейки размером 8×8 пикселей. Из этих ячеек методом бегущего окна формируются пересекающиеся блоки размером 2×2 (т. е. 16×16 пикселей). Всего изображение разбивается на $7 \times 15 = 105$ блоков. Каждая ячейка представляется в виде вектора.

Внутри ячейки для каждого из 64 пикселей вычисляются модуль и направление градиента. Найденное значение модуля прибавляется к одной из координат вектора v . К какой именно координате – зависит от найденного направления градиента. Для каждого блока путём конкатенации четырёх векторов, соответствующих четырём входящим в блок ячейкам, формируется вектор размерности 36 и затем нормируется. Соединение полученных со всех 105 блоков векторов в один длинный вектор, состоящий из 3780 элементов, составляет описание изображения через 3780 HOG-признаков.

Таким образом, каждому обучающему изображению соответствует точка в 3780-мерном пространстве признаков, которой, в свою очередь, соответствует значение «примитив» (+1) или «не примитив» (-1).

На втором этапе работы алгоритма с помощью метода опорных векторов между точками двух классов строится разделяющая гиперплоскость – решающая поверхность. Поиск примитивов на тестовом изображении с помощью обученного классификатора осуществляется методом бегущего окна, приводимого с помощью масштабирования к стандартному разрешению 64×128 , и вычислением HOG-признаков внутри каждого окна. В зависимости от вычисленных признаков классификатор решает, является или нет область, покрываемая окном детектора, изображением примитива.

Преобразование Хафа. Данный алгоритм предназначен для поиска объектов, принадлежащих определённому классу фигур, с использованием процедуры голосования. Процедура голосования применяется к пространству параметров, из которого и получают объекты определённого класса фигур по локальному максимуму в так называемом накопительном пространстве, которое строится при вычислении трансформации Хафа.

Алгоритм преобразования Хафа использует массив, называемый аккумулятором, для определения присутствия прямой $y = mx + b$. Размерность аккумулятора равна количеству неизвестных параметров пространства Хафа. Два измерения аккумулятора соответствуют значениям параметров m и b . Для каждой точки и её соседей алгоритм определяет, достаточен ли вес границы в этой точке. Если да, то алгоритм вычисляет параметры прямой и увеличивает значение в ячейке аккумулятора, соответствующей данным параметрам.

Потом, найдя ячейки аккумулятора с максимальными значениями, обычно поиском локального максимума в пространстве аккумулятора, могут быть определены наиболее подходящие прямые. Самый простой способ – это пороговая фильтрация. Однако в разных ситуациях разные методы могут давать разные результаты. Так как полученные прямые не содержат информацию о длине, следующим шагом является нахождение частей изображения, соответствующих найденным прямым.

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.

Выводы. В результате проведенного исследования были определены основные типы графических примитивов. Рассмотрены и проанализированы методы определения опорных точек на фотоизображении для детектирования объектов и их классификации. Выделены достоинства и недостатки рассмотренных методов.

Полученные результаты планируется использовать при выделении и классификации объектов на изображении в задачах 3D-реконструкции.

Список литературы

1. Выделение геометрических примитивов на фотоизображении / Р.А. Сорокин, Д.М. Бочаров // Инновационные перспективы Донбасса: материалы VI международной научно-практической конференции (Донецк, 20 – 22 мая 2015 года) / Донецкий национальный технический университет. Донецк, 2015. – Т.-5. Компьютерные науки и технологии. – С. 86-90.
2. Анализ влияния методов выравнивания яркости фотоснимков на качество выделения геометрических примитивов / А.В. Захаров, Р.А. Сорокин // Компьютерная и программная инженерия – 2015: материалы международной научно-технической конференции студентов, аспирантов и молодых ученых (Донецк, 15 – 16 декабря 2015) / Донецкий национальный технический университет. Донецк, 2015. - С. 274-279.
3. Исследование методов определения опорных точек на изображении / В.И. Махов, Д.М. Бочаров // Информационные управляющие системы и компьютерный мониторинг: материалы V международной научно-технической конференции студентов, аспирантов и молодых ученых (Донецк, 22 – 23 апреля 2013 года) / Донецкий национальный технический университет. Донецк, 2014. – С. 345-351.
4. Красильников Н. Н. Цифровая обработка 2D- и 3D-изображений. : Учебник для ВУЗов. – ВНУ, 2011. – 608 с.

УДК 004.932.72'1

Распознавание многоугольников на изображении с помощью алгоритма ВФОА

Минков О.В. (5 курс, факультет КНТ), Ручкин К.А., доцент каф. ИИСА

Аннотация

Минков О.В. Распознавание многоугольников на изображении с помощью алгоритма ВФОА. Выполнен обзор методов, которые могут быть применены для задачи распознавания многоугольников. Рассмотрена работа алгоритма ВФОА на примере распознавания окружности. Рассмотрена модификация алгоритма ВФОА для распознавания многоугольников.

Цель работы – рассмотреть применение эвристических алгоритмов (в частности, алгоритма ВФОА) в задаче распознавания многоугольников на изображении.

Впервые алгоритм ВФОА, известный как алгоритм оптимизации, основанный на передвижении бактерий, был предложен в 2002 году [1]. С алгоритмической точки зрения, он представляет собой многоагентную систему, которая имитирует отдельное и сгруппированное оптимальное поведение бактерий, живущих в кишечнике большинства млекопитающих. С помощью этого алгоритма существует возможность обнаруживать формы, если адаптировать его для задачи распознавания различных форм [2].

Для задачи распознавания многоугольников выбран алгоритм ВФОА, так как для нахождения им различных фигур достаточно заменить функцию определения оптимальности особи и в зависимости от фигуры будет изменено пространство поиска.

Описание работы алгоритма ВФОА для распознавания окружности.

Каждый предполагаемый круг представлен позицией бактерии, где первые два компонента вектора, x и y – координаты центра этой окружности, а третий компонент r – радиус. Пусть (x_k, y_k, r_k) будет k -м тестовым кругом популяции, где $k = 1, \dots, S$, где S – это размер популяции, т.е. обозначает общее количество тестовых кругов. В фазе инициализации случайные значения в подходящем диапазоне присвоены каждой из трех координат векторов. Давайте рассматривать $2N$ равномерно распределенных демонстрационных точек на круге.

Предположим, что искомая окружность располагается на некотором изображении. Функция $P(x, y)$ равна 1, если пиксель (x, y) является граничным, 0 – иначе. Пусть A будет граничной матрицей. Целевую функцию обозначим как J . Пусть для (x_0, y_0, r_0) тестового круга ее значение будет $J_0 = J(A, x_0, y_0, r_0)$.

Для определения является ли кандидат кругом можно рассмотреть множество кругов с центром в (x_0, y_0) и радиусом, варьирующимся в диапазоне от

$r_0 - \delta$ до $r_0 + \delta$. Назовем это множество тестовой полосой. Значение δ можно взять как процент от радиуса, предлагается $\delta = r_0/8$. После этого N тестовых точек берутся для каждого круга в тестовом диапазоне. Тестовые точки расположены на круге на равном расстоянии друг от друга. Пусть i -я тестовая точка на тестовом круге имеет радиус $r_0 + j$ ($-\delta \leq j \leq \delta$) и обозначается (x_i^j, y_i^j) .

$$x_i^j = (r_0 + j) \cos \frac{2\pi}{N} i, (1) \quad y_i^j = (r_0 + j) \sin \frac{2\pi}{N} i \quad (2)$$

где $i = 1, 2, \dots, N; j = -\delta, -\delta + 1, \dots, 0, \dots, \delta - 1, \delta$.

Чтобы измерить степень принадлежности тестовой точки к окружности центрального круга определим функцию расстояния μ . Значение функции расстояния μ_i^j для тестовой точки (x_i^j, y_i^j) определяется как:

$$\mu_i^j = P(x_i^j, y_i^j) \exp\left(-\frac{1}{2\sigma^2} (r + j + r_0)^2\right) \quad (3)$$

Значение μ может изменяться в диапазоне от 0 до 1. Когда $\mu_i^j = 1$, это означает, что выбранная точка граничная и она находится на центральном круге. Если (x_i^j, y_i^j) не граничная точка то $\mu_i^j = 0$.

Целевая функция, соответствующая (x_0, y_0, r_0) для граничной матрицы A определяется по следующей формуле

$$J(A, x_0, y_0, r_0) = 1 - \frac{1}{(2\delta+1)N} \sum_{i=1}^N \sum_{j=-\delta}^{\delta} \mu_i^j \quad (4)$$

Целевая функция взвешивает каждую граничную точку согласно радиусу круга кандидата.

Таким образом, становится возможным с помощью алгоритма ВФОА распознать окружность на изображении. Аналогичным образом можно подобрать параметры, которые идентифицируют любую другую фигуру, определить соответствующую этой фигуре целевую функцию и приведенный алгоритм можно будет использовать для поиска этой фигуры на изображении.

Описание работы алгоритма ВФОА для распознавания многоугольников. Представим бактерию позицией предполагаемой формы в виде координат x и y , ее угла поворота a и масштаба s . Форма является массивом вершин относительно ее положения.

Пусть (x_k, y_k, a_k, s_k) будет k -ой тестовой формой популяции, где $k = 1, \dots, S$, а S — это размер популяции, т.е. обозначает общее количество тестовых форм. При инициализации каждой из четырёх координат векторам присваиваются случайные значения в подходящем диапазоне. Рассмотрим N демонстрационных точек многоугольника.

Предположим, что искомая форма располагается на некотором изображении. Функция $P(x, y)$ равна 1, если пиксель (x, y) является граничным, 0 – иначе. Пусть A будет граничной матрицей. Целевую функцию обозначим как J . Пусть для (x_0, y_0, a_0, s_0) тестовой формы ее значение будет $J_0 = J(A, x_0, y_0, a_0, s_0)$.

Для определения является ли кандидат искомой формой нужно вычислить целевую функцию. Чем меньше значение целевой функции, тем более

точно форма соответствует кандидату. Целевая функция вычисляется на основании суммы расстояний от тестовых точек к ближайшим граничным точкам, если таковые имеются.

Показатель r представляет собой радиус окружности вокруг тестовой точки, где будет производиться поиск граничной точки. В общем случае предлагается брать его следующим образом:

$$r = \max(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) * 0.1, \quad (5)$$

где $i = 0 \dots N, j = 0 \dots N$.

Чтобы измерить степень принадлежности тестовой точки к форме на граничной матрице, определим функцию расстояния μ . Если существует такое k и l , что $P(x_i + k, y_i + l) = 1$, то значение функции расстояния μ_i для i -ой тестовой точки определяется как:

$$\mu_i = \min\left(\frac{\sqrt{k^2 + l^2}}{r}\right), \text{ что } \forall k, l: P(x_i + k, y_i + l) = 1, \quad (6)$$

где $k = -r \dots r, l = -r \dots r$.

Иначе $\mu_i = M$. Также $\mu_i = M$, если результат вычислений по формуле (6) больше, чем r , где M – число больше единицы. Предлагается брать $M = \frac{N}{2} + 1$.

Значение μ может изменяться в диапазоне от 0 до 1 или быть равно 2. Когда μ_i находится в диапазоне от 0 до 1, это означает, что граничная точка находится на расстоянии $\mu_i * r$ от тестовой точки. Если $\mu_i = M$, это означает, что нет граничной точки, которая лежит ближе, чем на расстоянии r от тестовой.

Целевая функция, соответствующая (x_0, y_0, a_0, s_0) для граничной матрицы A определяется по следующей формуле:

$$J(A, x_0, y_0, a_0, s_0) = \frac{1}{N} \sum_{i=1}^N \mu_i \quad (7)$$

Выводы. Проведен обзор методов, которые могут быть применены для задачи распознавания многоугольников. Выбран метод распознавания для этой задачи. Обоснованы причины выбора. Рассмотрена работа алгоритма BFOA на примере распознавания окружности и рассмотрена адаптированная работа алгоритма для распознавания многоугольников. Как мы видим, целевые функции похожи тем, что в них взвешивается каждая граничная точка согласно радиусу круга кандидата. Отличаются они своими функциями расстояния. В дальнейшем планируется адаптировать алгоритм для распознавания произвольной фигуры.

Список литературы

1. Kim D.H., Abraham A., Cho J.H. A Hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization // Information Sciences. – 2007. – № 18 (177). – P. 3918-3937.
2. S. Dasgupta, S. Das, A. Biswas, A. Abraham Automatic circle detection on digital images with an adaptive bacterial foraging algorithm // Soft Comput. – 2010. – P. 1151–1164.

УДК 004.67

**Разработка аппаратно-программного комплекса ультразвуковой
идентификации объектов**

Потопахин А.А, Гудаев О.А.

Донецкий национальный технический университет, г. Донецк

факультет компьютерных наук и технологий

E-mail: alexey_potopahin@hotmail.com

Аннотация

Потопахин А.А., Гудаев О.А. Разработка аппаратно-программного комплекса ультразвуковой идентификации объектов. Произведен выбор необходимых компонентов и создана общая схема системы для аппаратно-программного комплекса ультразвуковой идентификации объектов. Выполнено проектирование и составление возможных команд взаимодействия между клиентом и сервером.

Ключевые слова: идентификация объектов, ультразвуковой датчик расстояния, аппаратно-программный комплекс.

Аппаратный модуль системы. В качестве блока управления датчиком расстояния (HC-SR04) и серводвигателем (необходим для движения датчика расстояния) будет рассматриваться одноплатный компьютер Raspberry Pi Model B первого поколения. Выбор обусловлен низкой стоимостью компьютера. Альтернативным вариантом мог бы использоваться Arduino, стоимость которого намного ниже, чем выбранный блок управления, но Raspberry Pi имеет аппаратные возможности установления сетевого соединения через Ethernet или Wi-Fi, которое является основным в работе проектируемой системы.

Датчик расстояния HC-SR04 представляет собой плату, на которой размещены: излучатель и приемник ультразвука и управляющая электронная схема. Датчик имеет небольшие габариты и простой интерфейс – два вывода питания, один вход и один выход. Излучатель сенсор HC-SR04 излучает короткий ультразвуковой импульс (в момент времени 0), который отражается от объекта и принимается приемником сенсора. Расстояние рассчитывается исходя из времени до получения эха и скорости звука в воздухе. Сенсор получает сигнал эха, и выдает расстояние, которое кодируется длительностью электрического сигнала на выходе датчика. Следующий импульс может быть излучён, только после исчезновения эха от предыдущего. Это время называется периодом цикла. Рекомендованный период между импульсами должен быть не менее 50 мс. Если на сигнальный пин подаётся импульс длительностью 10 мкс, то ультразвуковой модуль будет излучать восемь пачек ультразвукового сигнала с частотой 40 кГц и

обнаруживать их эхо. Измеренное расстояние до объекта пропорционально ширине эха [1].

Общая схема системы. На рис. 1 представлена макетная схема системы идентификации объектов. Серводвигатель управляется с Raspberry Pi при помощи одного проводника, который подключается к 18 GPIO. Однако питание 5В в целях безопасности используется отдельно (от USB-порта).

Ультразвуковой датчик расстояния (HC-SR04) может работать при напряжении 5В. Однако Raspberry Pi не может принять сигнал от датчика напряжением 5В. Входы одноплатного компьютера рассчитаны на принятие сигнала только 3.3В. Для того, чтобы Raspberry Pi мог обработать сигнал 3.3В, необходимо в схеме использовать делитель напряжения. Это достигается при помощи двух резисторов сопротивлением 330 и 470Ом [2].

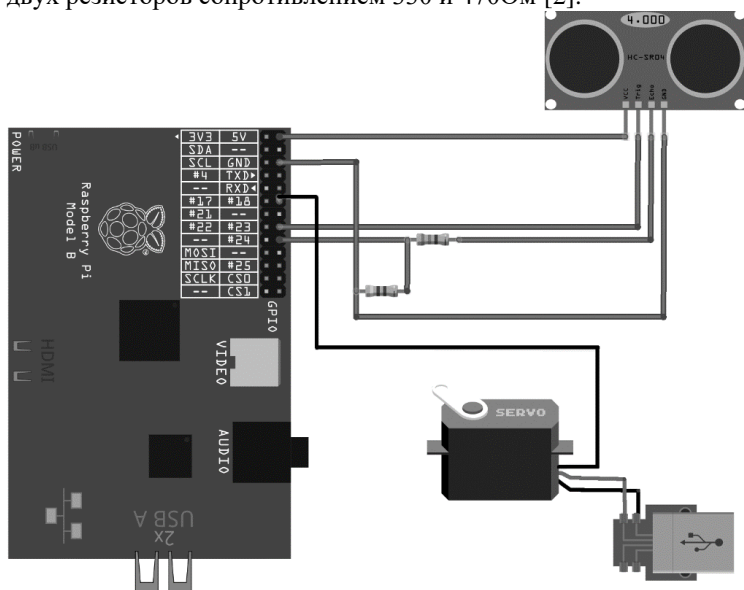


Рис. 1 – Макетная схема системы

Проектирование взаимодействия клиент-сервер. Программная архитектура системы строится по схеме клиент-сервер [3]. Между клиентом и сервером устанавливается соединение через socket интерфейс по протоколу UDP. В качестве сервера выступает блок управления датчиками и датчик расстояния, а в качестве клиента – кроссплатформенная система обработки данных (данные датчика расстояния с сервера) и распознавания объектов. Диаграмма компонентов системы представлена на рис. 2.

Главными компонентами на диаграмме являются «uos_server.py», который разворачивается на Raspberry Pi, (является сервером) и компонент

«UOSClient» – клиентская система, которая отправляет запросы серверу в формате JSON.

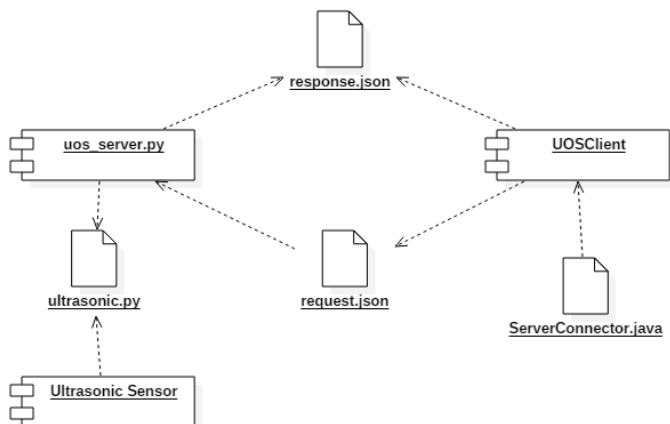


Рис. 2 – Диаграмма компонентов

Проектирование JSON команд между клиентом и сервером. Нотация JSON – простой формат обмена данными. Выбор данного формата обоснован тем, что он удобен для чтения и написания как человеком, так и компьютером.

Для работы системы необходимы следующие команды:

- команда соединения с сервером (проверка на готовность сервера);
- команда сканирования объекта;
- команда замера текущего расстояния до объекта.

Структура команды соединения с сервером (проверка на готовность сервера) представлено на рис. 3. Команда также содержит IP-адрес и имя устройства клиента.

```
{
    "command": "init",
    "client": "192.168.0.38",
    "client_name": "ASUS-N550"
}
```

Рис. 3 – Команда соединения с сервером

Ответ на соединение: сервер должен ответить готовностью и сообщить технические характеристики устройства, на котором запущен. Ответ соединения с сервером показан на рис. 4. Ответ содержит IP-адрес и порт сервера.

```
{
  "result": "ready",
  "sys_info": [
    "Linux",
    "Raspberrypi",
    "4.1.13+",
    "#826 PREEMPT Fri Nov 13 20:13:20 GMT 2015",
    "armv6l"
  ],
  "server": "192.168.0.50",
  "port": "8888"
}
```

Рис. 4 – Ответ на команду соединения с сервером

Формирование команды сканирования объекта представлено на рис. 5. Команда должна содержать название, точность сканирования (настройка датчика расстояния) и скорость сканирования (настройка серводвигателя).

```
{
  "command": "scanning",
  "accuracy": "0.2",
  "speed": "0.8",
  "client": "192.168.0.38",
  "client_name": "ASUS-N550"
}
```

Рис.5 – Команда сканирования объекта

В ответ на команду сканирования объекта клиентом, сервер должен выполнить сканирование, а затем отправить результат в виде массива расстояний (см. рис. 6).

```
{
  "result": "complete",
  "distances": [
    228.9853811264038,
    227.2884964942932,
    228.28209400177002,
    ...
  ],
  "measure": "cm",
  "server": "192.168.0.50",
  "port": "8888"
}
```

Рис. 6 – Ответ в виде массива расстояний

Команда замера текущего расстояния до объекта аналогична команде сканирования, но не содержит значения скорости, т.к. данный режим не использует сервопривод (см. рис. 7).

```
{
    "command": "get_distance",
    "accuracy": "0.02",
    "client": "192.168.0.38",
    "client_name": "ASUS-N550"
}
```

Рис. 7 – Команда замера текущего расстояния

Ответ на команду замера текущего расстояния до объекта представлен на рис. 8.

```
{
    "result": "complete",
    "distance": "228.67871522903442",
    "result": "cm",
    "server": "192.168.0.50",
    "port": "8888"
}
```

Рис. 8 – Ответ в виде текущего расстояния до объекта

Выводы. Выполнен выбор необходимых аппаратных модулей и создана общая схема системы для аппаратно-программного комплекса ультразвуковой идентификации объектов. Определены основные компоненты системы. Выполнено проектирование программной архитектуры и составлена возможная система команд взаимодействия между клиентом и сервером. Предложенные JSON команды позволят клиентской системе получать все необходимые данные и конфигурировать работу аппаратных модулей, а также получать информацию о состоянии сервера.

Список литературы

1. Ультразвуковой сонар HC-SR04 [Электронный ресурс] // Космодром – электронные компоненты: [сайт]. [2016]. URL: <http://www.kosmodrom.com.ua/el.php?name=HC-SR04>
2. Ультразвуковое измерение при помощи Python [электронный ресурс] // Компания Raspberry-Spy: [сайт]. [2015]. URL: <http://www.raspberrypi-spy.co.uk/2012/12/ultrasonic-distance-measurement-using-python-part-1>
3. Потопахин, А.А. Ультразвуковая идентификация объектов по топологическим признакам / А.А. Потопахин, О.А. Гудаев // Компьютерная и программная инженерия. Сборник материалов международной научно-технической конференции студентов, аспирантов и молодых учёных 15-16 декабря 2015 года. - Донецк, ДонНТУ, 2015. – 336-337 с.

УДК 004.8

Двумерная модель компьютерной игры в жанре пошаговой стратегии

Погребницкий Б.С., Семенова А.П.

Донецкий национальный технический университет

Кафедра искусственного интеллекта и системного анализа

E-mail: pogrebithiy@mail.ua

Аннотация

Погребницкий Б.С., Семенова А.П. Двумерная модель компьютерной игры в жанре пошаговой стратегии. В работе рассматривается понятие компьютерной пошаговой стратегии. Проводится анализ различных видов пошаговых игр. Приведено описание двумерной модели компьютерной игры в жанре игровой стратегии. Сформулированы основные особенности предлагаемой модели игры.

Ключевые слова: пошаговая стратегия, игровой процесс, игровая карта, юнит.

Анализ предметной области. Стратегические игры – это игры в которых для достижения победы необходимо планировать и продумывать последующие действия. Идейнными родителями современных компьютерных стратегий можно считать настольные игры. К таким играм можно отнести шахматы, крестики-нолики, шашки и другие. Их история начитывает тысячелетия, например игра рэндзю появилась во втором тысячелетии до нашей эры, а в игру сенет играли еще в древнем Египте более шести тысяч лет назад [1].

Целью написания данной статьи является анализ типов пошаговых стратегий, выявление особенностей компьютерных игр данного жанра и описание созданной двумерной модели компьютерной игры.

Обзор существующих аналогов. Стратегические компьютерные игры можно разделить на два класса: стратегии в реальном времени и пошаговые. Пошаговые стратегии отличны тем что в них игровой процесс разбивается на фиксирование отрезки времени (ходы или шаги), на протяжении которых игроки поочередно совершают свои действия.

Сейчас существует большое количество различных компьютерных пошаговых стратегий, каждая из которых имеет свои уникальные особенности. К самым знаменитым из них можно отнести:

1. Серия игр Civilization, первая из которых вышла в 1991 г. Ее основной особенностью является развитие выбранной нации от доисторических времен до космической эпохи, при этом игроку приходится решать политические, экономическое и военные задачи.

2. Серия игр Heroes of Might and Magic, первая из которых вышла в 1995 г. В ней существа одного типа объединяются в отряды, количество отрядов в одной армии ограничено, а размер отряда нет. Армии под предводительством героев путешествуют по миру, а при сражении с противником игра переносится на карту сражений. Также возможна прокачка героев, добыча ресурсов, развитие городов.

3. Серия игр Disciples, первая из которых вышла в 1999 г. Основным отличием является прокачка юнитов, т. е. одно существо, при наборе необходимого количества опыта, превращается в другое, которое будет заведомо сильнее и может отличаться по своим особенностям. Также возможен выбор рас, героев, развитие и завоевание городов.

4. Серия игр X-COM, первая из которых вышла в 1993 г. В ней совмещены особенности пошаговых и ролевых игр. Игрок управляет отрядом бойцов, каждый из которых может отличаться характеристиками, вооружением, способностями, а также использовать различные предметы, такие как аптечки, гранаты и т. д. По мере прохождения игры будут доступны улучшения экипировки и прокачка персонажей.

5. Серия игр Panzer General, первая из которых вышла в 1993 г. Эта игра на тему второй мировой войны. Игровые юниты представлены отрядами и техникой того времени и имеют свои особенности, поле состоит из шестиугольных клеток (гексов), каждый юнит занимает собой одну из них.

6. Серия игр Total War, первая из которых вышла в 2000 г. Эти игры отличаются от других тем, что действия на глобальной карте, такие как постройка зданий, найм отрядов, движение армий совершаются игроками во время своего хода, а битвы происходят на карте сражений в режиме реального времени.

Также к пошаговым стратегиям можно отнести серию игр Worms, первая из которых вышла в 1995г. и, в которой игроки поочередно пытаются уничтожить червячков соперника с помощью различных видов вооружения. Отличием от других игр является то, что игрок полностью управляет своими юнитами с помощью клавиатуры и мыши [2].

Помимо вышеперечисленных, существует множество других пошаговых игр, которые сыграли свою роль в становлении жанра, а с каждым годом их становится все больше, и каждая из них имеет свои отличительные особенности и нововведения.

Описание двумерной модели компьютерной игры. В результате анализа компьютерных игр жанра пошаговой стратегии были выделены несколько основных типов игрового процесса:

– используются два типа карт – карта сражений, где происходит непосредственно битва и глобальная карта, на которой доступны все остальные игровые возможности;

– используется только одна глобальная карта, на которой сосредоточен весь игровой процесс, а сражения проводятся автоматически либо вообще отсутствуют;

– используется только карта сражений, а все остальные действия производятся в меню или какой-либо иной форме, не подразумевающей основной игровой процесс.

В описываемой модели игры используется первый тип игрового процесса, так как является более распространенным и представляет наиболее широкий спектр игровых возможностей и решений. Второй тип является более подходящим для игр ориентированных на экономическую и политическую составляющие, что противоречит основному игровому замыслу, а третий тип является лишь урезанной версией первого.

Также существуют два основных типа игровых полей. К первому типу относится игровое поле, представленное в виде совокупности ячеек, как правило, квадратов или шестиугольников. Ко второму – игровое поле без ярко выраженного разделения на части.

В разработанной двумерной модели компьютерной игры в жанре пошаговой стратегии, оба вида игровых карт (глобальная карта и карта сражения) будут представляться в виде сетки с квадратными ячейками. Каждая ячейка в отдельности или их группа может содержать игровой объект. На карте сражений такими объектами являются отдельные юниты или элементы ландшафта (рис. 1).



Рис. 1 – Образец карты сражений

А на глобальной карте – армии, интерактивные объекты, такие как города или отдельные здания, с которыми можно взаимодействовать и элементы глобального ландшафта (рис. 2) [3].

Также для увеличения разнообразия вводится элемент случайности. Его, например, можно использовать в сражении для определения результата атаки. При этом основной проблемой становится соблюдение баланса. Чтобы игра не стала непредсказуемой, предлагается использовать в качестве метрики соотношение 75% к 25%, т. е. на результат работы модуля события будет влиять случайная переменная.

Важной частью любой стратегической игры является принятие решений на основе текущей игровой ситуации, и для увеличения разнообразия игрового процесса требуются факторы, которые не зависят от действий игроков на поле боя. Примером такого фактора может быть само игровое поле, правильное позиционирование на котором будет оказывать влияние на исход сражения. Во многих играх для создания игрового поля используют различные алгоритмы, которые позволяют создавать карту случайным образом. С одной стороны, это увеличивает вариативность и интерес, так как практически полностью исключает возможность повтора карт, но с другой стороны – может негативно сказываться на ее балансе и тактической составляющей. Поэтому в предлагаемой двумерной модели компьютерной игры уделяется пристальное внимание дизайну карт. Необходимо создать несколько карт, каждая из которых будет уникальной. Карта должна способствовать определенному стилю игры или усложнять его и требовать определенного тактического подхода для получения преимущества (рис 1,2).

Выводы. В статье выполнен анализ существующих компьютерных игр жанра пошаговой стратегии для формирования общих требований и выделения ключевых особенностей данного типа игр. Приводится общее описание двумерной модели компьютерной игры в жанре пошаговой стратегии, особенности ИИ, а также отличительные характеристики предлагаемой модели игры по сравнению с существующими аналогами.

Список литературы

1. Белл Р. Энциклопедия настольных игр народов мира.: Litres, 2013. – 311с.
2. Пошаговые стратегии — список лучших пошаговых стратегий на PC. [Электронный ресурс]. – Режим доступа: <http://farap.ru/strategii/poshagovyestrategii-spisok-luchshix-poshagovyux-strategij-na-pc.html> – 26.04.2016.
3. Википедия. Свободная энциклопедия. Пошаговая стратегия. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D0%B0%D0%B3%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%B5%D0%B3%D0%B8%D1%8F – 26.04.2016.

УДК 519.683.8

Разработка обучающего пакета программ работы с контейнерными классами

Ольшевский А.И., Стась А.В.

Донецкий национальный технический университет

E-mail: a_olshevskiy@mail.ru

Аннотация

Ольшевский А.И., Стась А.В. Разработка обучающего пакета программ работы с контейнерными классами Рассмотрена классификация по типам основных структур данных контейнерных классов. Описаны классы и различные конструкторы для создания этих структур. Определен подход формирования поддерживающий разные наборы операций, среди которых есть совпадающие операции. Проанализирована эффективность разных методов соответствующих операций.

Общая постановка проблемы. Для высшего образования первостепенную актуальность приобретает задача использования возможностей информационных технологий в обучении. Программирование – это искусство мыслить и достигать успеха. Процесс реформирования знаний рассматривается в контексте создания таких педагогических технологий, которые обеспечивали бы переход от формально-дисциплинарного к проблемно-активному типу обучения [1].

Язык С++ захватил пальму первенства по количеству приверженцев среди программистов. Ведь он является преемником языка Си – огромного количества наработок, готовых библиотек и примеров. Огромный плюс – это кросс-платформенность. Можно создавать программы любой сложности для Windows, Linux и MacOS.

Постановка задачи. Объектом разработки являются шаблоны классов, часть библиотеки С++, в которую входят контейнерные классы, а также алгоритмы и итераторы стандартной библиотеки шаблонов (STL — Standard Template Library).

Цель проекта – накопление и систематизация информации для обучения пользователей возможностям стандартной библиотеки шаблонов содержащей контейнеры, реализующие основные структуры данных, для проектирования и написания программ. Использование контейнеров позволяет значительно повысить надежность программ, их переносимость и универсальность, а также уменьшить сроки их разработки.

Разрабатываемый пакет программ позволит уменьшить время обучения и упростить затраты на вдумчивое освоение библиотеки для эффективного использования контейнеров.

Естественно, эти преимущества не даются даром: универсальность и безопасность использования контейнерных классов не могут не отражаться на быстродействии программы. Снижение быстродействия в зависимости от реализации компилятора может быть весьма значительным.

Исследования. Первым этапом разработки является классификация по типам основных структурам данных контейнерных классов. Контейнеры можно разделить на два типа: последовательные и ассоциативные[2].

Последовательные контейнеры обеспечивают хранение конечного количества однотипных величин в виде непрерывной последовательности. К ним относятся векторы (vector), двусторонние очереди (deque) и списки (list), а также так называемые адаптеры, то есть варианты, контейнеров – стеки (stack), очереди (queue) и очереди с приоритетами (priority_queue).

Ассоциативные контейнеры обеспечивают быстрый доступ к данным по ключу. Эти контейнеры построены на основе сбалансированных деревьев. Существует пять типов ассоциативных контейнеров: словари (map), словари с дубликатами (multimap), множества (set), множества с дубликатами (multiset) и битовые множества (bitset).

Ядро стандартной библиотеки шаблонов включает три основных элемента: контейнеры, алгоритмы и итераторы [3]. Они работают совместно один с другим, представляя тем самым готовые решения различных задач программирования.

Контейнеры – это объекты, которые содержат другие объекты.

Каждый вид контейнера обеспечивает свой набор действий над данными. Выбор вида контейнера зависит от того, что требуется делать с данными в программе.

Алгоритмы обрабатывают содержимое контейнеров. Их возможности включают средства инициализации, сортировки, поиска и преобразования содержимого контейнеров. Многие алгоритмы работают с заданным диапазоном элементов контейнера.

Итераторы – это объекты, которые в той или иной степени действуют подобно указателям. Они позволяют циклически опрашивать содержимое контейнера практически так же, как это делается с помощью указателя при циклическом опросе элементов массива.

Программная реализация. Обучающий пакет программ предоставляет пользователям теоретическую информацию по типам основных структурам данных контейнерных классов, примеры программ реализации алгоритмов и итераторов при работе со стандартной библиотекой шаблонов.

Смысл одноименных операций для различных контейнеров одинаков, основные операции применимы ко всем типам контейнеров. Стандарт определяет только интерфейс контейнеров, поэтому разные реализации могут сильно отличаться по эффективности.

Векторы (vector), двусторонние очереди (deque) и списки (list) поддерживают разные наборы операций, среди которых есть совпадающие операции. Они могут быть реализованы с разной эффективностью.

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.

Выбор контейнера позволяет изучить теоретическая информация по этому виду объекта и обеспечить выбор своего набора действий над данными (программную реализацию). Выбор вида контейнера зависит от того, что требуется делать с данными в программе.

Теоретическая информация по типу контейнера. Например: вектор – это структура, эффективно реализующая произвольный доступ к элементам, добавление в конец и удаление из конца; двусторонняя очередь эффективно реализует произвольный доступ к элементам, добавление в оба конца и удаление из обоих концов; список эффективно реализует вставку и удаление элементов в произвольное место, но не имеет произвольного доступа к своим элементам и т.п.

Векторы, двусторонние очереди и списки поддерживают разные наборы операций, среди которых есть совпадающие операции. Они могут быть реализованы с разной эффективностью.

В теоретической части приводятся схемы организации данных, как обеспечить доступ к элементам и за какое время, как будет организованно выделение память под очередной элемент или перераспределение памяти при удалении.

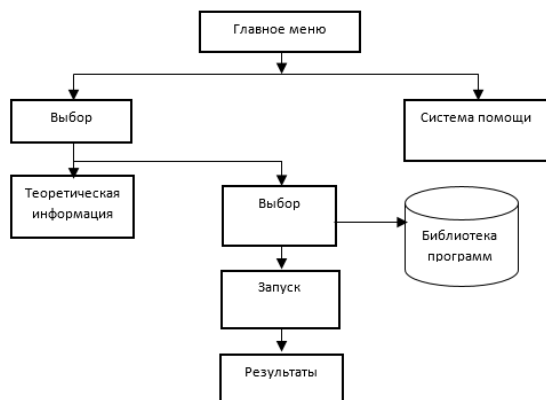


Рис. 1. Общая структура вызовов элементов программы и их взаимодействия

Библиотека программ содержит пакет обучающих программ работы с контейнерными классами, которые могут пополняться.

Программист может создавать собственные контейнерные классы на основе имеющихся в стандартной библиотеке.

Выбор программной реализации позволяет выбрать программу из набора, запустить ее и получить результат. Например работы с вектором: в файле находится произвольное количество целых чисел; программа считывает их в вектор и выводит на экран в том же порядке.

Выводы

Были определены цели, которые можно достичь, используя обучающий пакет программ, и сформулированы требования, которые должны выполнять программы реализующие методы и операции при работе с основными структурами данных контейнерных классов. Так же, был проведен анализ эффективности использования операций различными способами. Были выделены их недостатки и преимущества, которые будут учитываться при разработке программного обеспечения.

Список литературы

1. Юрков Н. К. Интеллектуальные компьютерные обучающие системы / Юрков Н. К. [Электронный ресурс]. – Режим доступа : http://www.mtas.ru/upload/library/MONOGRAFIYa_IKOS_2010.pdf
2. Павловская Т. А. С/С++. Программирование на языке высокого уровня / Павловская Т. А. – СПб.: Питер, 2003. – 461 с.
Шилдт Г. С++: базовый курс / Шилдт Г. – [3-е изд.] ; [пер. с англ.]. – М.: Вильямс, 2010. – 624 с.

Секция № 7
Медиаиндустрия и
дизайн



УДК 004.92

Особенности профессии проектировщика интерфейсов

Кладько Д.С., Харитонов В.В.

Донецкий национальный технический университет, г. Донецк

кафедра компьютерных систем мониторинга

E-mail: this_is_not_email@mail.ru

Аннотация

Кладько Д.С., Харитонов В.В. Особенности профессии проектировщика интерфейсов. Определены основные профессиональные требования к специалистам по созданию интерфейсов. Выполнен анализ базовых навыков, необходимых для успешного проектирования программ.

Ключевые слова: интерфейс, проектировщик, юзабилити, UX-дизайн, востребованность, профессиональные качества, графический дизайн, среда разработки, макет, особенности, пользователь

Актуальность. Стремительное развитие технологий проникло практически во все сферы деятельности человека. По этой причине компании, чья деятельность, так или иначе, связана с разработкой ПО, стали понимать, что необходимо совершенствовать интерфейсы своих программных продуктов, делая их не только красивыми, но еще и удобными для оптимизированного взаимодействия с пользователями. Некачественный интерфейс снижает эффективность взаимодействия пользователя с программой. Поэтому главной задачей проектировщика является создание красивого и удобного для использования ресурса, выполняющего целевые потребности пользователя.

Целью исследования является выявление основных профессиональных требований к специалистам по созданию интерфейсов. А также обзор базовых навыков дизайнера интерфейсов.

Основная часть. В процессе анализа литературы были выявлены следующие требования к специалистам направления проектирования интерфейсов:

1. *Исследовательские навыки.* Умение анализировать идеи, от их начальной стадии до реализации в макете, значительно повышает вероятность успешного достижения целей проекта. UX-дизайнер должен быть любознательным и иметь особое исследовательское чутье. Получение информации от пользователей, поиск нужных данных среди результатов многочисленных маркетинговых и научных исследований требуют знаний психологии и социологии. Способности к проведению исследований очень важны в UX дизайне, но не менее важен навык извлекать максимальную выгоду из

результатов исследований, умение использовать эти результаты в дальнейшей работе.

2. *Навыки программирования пользовательского взаимодействия.* Как известно, интерфейс – взаимодействие человека с машиной. Разрабатывая всевозможные принципы взаимодействия между пользователем и компьютером, проектировщики интерфейсов являются своего рода посредниками между живой и неживой природой.

С точки зрения теории систем любой интерфейс — место с низким КПД, в котором теряется энергия, скорость, пропускная способность, время, аудитория и деньги. Самый неэффективный вид интерфейса — пользовательский. В отличие от аппаратных и программных, пользовательский интерфейс открывает безграничный простор для человеческих решений и ошибок [3]. Основной целью при разработке интерфейса для взаимодействия человека с машиной является создание среды, которая позволит легко (интуитивно), эффективно, и с удовольствием (*user friendly*) добиваться желаемого результата во время работы с машиной. Идеальным результатом считается тот, при котором оператор (или пользователь) производит минимальные усилия для ввода, получая от машины желаемый вывод, при этом машина минимизирует вывод лишних данных.

Сценарии, диаграммы переходов, модели взаимодействия — типичные результаты проектирования взаимодействия. Они отображают желаемый опыт взаимодействия пользователя с программой, который потом будет отражен в функциональной спецификации.

Так как проектирование пользовательского взаимодействия в первую очередь направлено на создание интуитивно понятных интерфейсов, умение поставить себя на место пользователя – также является одним из наиболее важных качеств UX-специалиста.

3. *Навыки создания прототипов и макетов.* Важно отметить, что способность разрабатывать качественные макеты является необходимым условием для приятного визуального дизайна и грамотного дизайна взаимодействия, но в каждой из этих областей термин «макет» используется в разных значениях. По причине того, что обоснование и цели создания макета отличные: визуальный дизайн добивается эстетичности, а проектирование взаимодействия — удобной схемы работы. Найти хорошего специалиста по интерфейсам с навыками дизайнера очень сложно. Поэтому работодатели практикуют найм дизайнеров, у которых есть некоторый опыт работы с интерфейсами и обучают их (непосредственно во время рабочего процесса под шефством более опытного сотрудника или на специализированных тренингах и курсах) проектировать интерфейсы, проводить юзабилити-тестирование. Как показывает практика: научить дизайнера заниматься аналитикой, думать о пользователях, проводить юзабилити-исследования — легко. В то время как научить аналитика создавать дизайн — практически невозможно.

4. *Навыки работы в средах разработки интерфейсов.* В сфере создания интерфейсов для UX-специалистов нет ограничений в выборе программного обеспечения. Каждый выбирает для себя удобную среду разработки. Так как в проектировании важен результат, а не способы решения поставленных задач.

Программное обеспечение, используемое проектировщиками интерфейсов условно можно разделить на две группы:

- Программы для прототипирования;
- Графические редакторы;

Прототипы стали незаменимой частью разработки интерфейсов. Они визуализируют конечный продукт на начальных стадиях разработки и помогают выявить серьезные недостатки. Сегодня среди проектировщиков распространено множество прикладных программ и онлайн ресурсов, дающих возможность составлять блок-схемы, графики, диаграммы и схемы. В таблице 1 рассмотрены наиболее популярные настольные программы среди проектировщиков.

Таблица 1 – Наиболее распространенные программы для создания прототипов

Название	Описание
Axure RP	<i>Достоинства:</i> Возможность программирования поведения кнопок, текстовых полей, панелей и прочих виджетов, вследствие чего получившиеся макеты или прототипы приближены к окончательному результату и доступны для тестирования [1]. <i>Недостатки:</i> Малое количество поставляемых с программой виджетов. Отсутствие экспорта готового проекта в PDF. Отсутствие «горячих клавиш» для многих действий. Высокая цена по сравнению с аналогами.
Omnigraffle pro	<i>Достоинства:</i> Большое количество стандартных элементов и фигур, представленных в трех вариантах: обычный, 2D и 3D. Возможность импорта элементов в отдельные коллекции изображений. <i>Недостатки:</i> Сравнительно небольшая стандартная библиотека специальных элементов. Высокая стоимость.
Balsamiq	<i>Достоинства:</i> Наличие онлайн версии. Широкий выбор готовых наборов компонентов. Возможность экспорта нарисованного макета в форматы png и pdf, а также в xml (что можно использовать в качестве backup или пересылки макета по почте в его минимальном размере). Кроссплатформенность. <i>Недостатки:</i> Отсутствие дополнительных библиотек. Создание статичных макетов.

Как видно из таблицы, различные программы предоставляют разные возможности и инструменты для прототипирования. Отдельно взятый специалист решает, какими программными методами он будет пользоваться для решения задач.

Однако структура интерфейса является начальным этапом создания программного продукта. После создания прототипа будущей программы, необходимо проработать ее внешний вид. В прямые обязанности UX-дизайнера не входит создание готового макета интерфейса, но умение подготавливать рабочие макеты для передачи их верстальщикам является неоспоримым плюсом. Что касается создания макетов, также существует множество инструментов, позволяющих создавать самые различные варианты готовых интерфейсов. В своей работе многие дизайнеры пользовательского интерфейса применяются такие инструменты как Photoshop, Illustrator, Fireworks (см. таблица 2), а также ручку и бумагу.

Таблица 2 – Наиболее распространенные графические редакторы

Название	Описание
Photoshop	<i>Достоинства:</i> Наличие специализированного режима для проектирования интерфейсов мобильных приложений и сайтов. Поддержка векторных объектов. <i>Недостатки:</i> Ограниченные возможности растровой графики. Невозможность быстрой работы с группами слоев без использования панели слоев.
Illustrator	<i>Достоинства:</i> Возможность одновременной работы с несколькими холстами. Преимущество вектора над растровой графикой (с появлением экранов «повышенной плотности»). Работа с объектами, а не слоями. Возможность группировки объектов в символы. <i>Недостатки:</i> Потеря качества растровых изображений (при наличии их в макете) при трансформации.
Fireworks	<i>Достоинства:</i> Возможность работать с несколькими страницами (с неограниченным количеством слоев на каждой). Возможность экспорта интерактивного PDF-файла. Доступность большого количества расширений, увеличивающих арсенал программы. Менее ресурсозатратный в сравнении с конкурентами. <i>Недостатки:</i> Отсутствие множества привычных графических инструментов.

5. *Художественный вкус и знание основ дизайна.* UX дизайнеры отвечают за создание проектов интуитивно понятных интерфейсов веб-сайтов или мобильных приложений, где каждый следующий шаг вытекает из предыдущего. Для того чтобы успешно выполнять работу, UX-дизайнеры должны анализировать потребности целевой аудитории, а затем, учитывая результаты исследований, создавать проект интерфейса и тестировать его на реальных пользователях. Такие дизайнеры не работают над созданием конечного продукта, они создают вайфреймы, раскадровки и карты сайтов, основу для дальнейшей работы команды. Профессия UX-дизайнера предполагает высокий уровень владения инструментами дизайна, художественный вкус, знание основ композиции и цветоведения, шрифтовой грамотности, увлеченность, креативное

мышление, также знание особенностей психофизического воздействия на человека.

6. *Копирайтерские навыки.* Хорошие копирайтер без труда могут провести пользователя сквозь все узкие места интерфейса при помощи понятной лаконичной инструкции. Они способны объяснять сложные вещи простым языком и излагать результаты исследований в доступном для понимания виде, не упуская при этом подробностей. Помимо руководств пользователя, справочных файлов, инструкций, подписей и набросков рекламных объявлений, результатом работы многих UX-проектировщиков является руководство по стилю, в котором детально описано, как следует использовать разработанные концепции.

Выводы. Профессия проектировщика интерфейсов, безусловно, интересна, перспективна и многогранна, ведь она возникла на стыке дизайнерской и инженерной специальностей. Исходя из рассмотренных позиций, UX-дизайнер должен понимать концепцию внешнего вида пользовательского интерфейса и управлять процессом разработки визуального дизайна.

UX-дизайнер, наряду с высоким профессионализмом в области визуального представления данных, информационной архитектуры и анализа юзабилити, должен уметь ставить себя на место представителя целевой аудитории. При разработке интерфейса, специалист должен максимально учитывать все мелочи, начиная от рабочей среды пользователя и типа электронного устройства, а также способами ввода и отображения информации.

Профессия UX-дизайнера находится в постоянном развитии. Это напрямую связано с появлением новых технологий. Поэтому для достижения успеха важно своевременно повышать квалификационный уровень. В ходе исследования был проведен детальный анализ основных пакетов программ, в которых специалистам данного направления необходимо уметь работать.

Список литературы

1. Axure_RP [Electronic resource] / Википедия – свободная энциклопедия: [сайт]. - Режим доступа : https://ru.wikipedia.org/wiki/Axure_RP
2. UX & Service Design [Electronic resource] / Интернет-ресурс. - Режим доступа : <http://uidg.blogspot.com/2013/10/fresh-10-usability-ux.html>
3. Дизайн Артема Горбунова [Electronic resource] / Интернет-ресурс. - Режим доступа : <http://artgorbunov.ru/bb/ui/evil/>

УДК 004.92

Разработка клиент-серверного приложения контроля эффективности рабочего времени

Титаренко О.Ю., Павлий В.А.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного дизайна и моделирования

E-mail: maru_ _chan@mail.ru

Аннотация

Титаренко О.Ю., Павлий В.А. Разработка клиент-серверного приложения контроля эффективности рабочего времени. Выполнен анализ существующих приложений контроля эффективности рабочего времени. Определены инструментальные средства разработки клиент-серверного приложения. Изучены основные принципы работы приложения и представлены в виде блок-схем и XML запросов и ответов сервера.

***Ключевые слова:** клиент-серверное приложение, контроль эффективности рабочего времени, автоматизации учета рабочего времени, средства разработки.*

Постановка задачи. В настоящее время проблема эффективности контроля учета рабочего времени приобретает все большее значение. Это обусловлено тем, что одной из главных проблем в деятельности предприятий является нерациональное использование рабочего времени его сотрудниками. Контроль работы сотрудников любого предприятия является эффективным способом выявления нерационального использования рабочего времени. В условиях работы за компьютером очень легко симитировать бурную деятельность. Без контроля не получится узнать, как выполняются поручения и выполняются ли они вообще.

Таким образом, возникает необходимость вести учет использования рабочего времени, который позволит определить, какие приложения сотрудник использовал во время работы, когда покинул рабочее место, сколько времени он отсутствовал, а также решить вопросы дисциплины и стимулировать подчиненных эффективно использовать рабочее время для выполнения поставленных задач. Учитывая, что в крупных предприятиях работает достаточно большое количество сотрудников, возникает необходимость разработки приложения для автоматизации учета рабочего времени. В целях повышения безопасности и агрегации данных учета в одном месте разрабатываемое приложение должно строиться на основе архитектуры «клиент-сервер». Это позволит получать информацию со всех компьютеров, на который установлен клиент и формировать статистику эффективности работы каждого сотрудника на

сервере. Для реализации такого приложения необходимо выполнить следующие шаги:

- проанализировать существующие приложения контроля эффективности рабочего времени, определить их преимущества и недостатки;
- определить средства разработки клиент-серверного приложения;
- представить принцип работы приложения в виде блок-схем и основных XML запросов и ответов сервера.
- разработка клиент-серверного приложения контроля эффективности рабочего времени.

Цель работы – изучить основные особенности автоматического учета эффективности рабочего времени и разработать соответствующее клиент-серверное приложение для сотрудников предприятий.

Анализ существующих приложений контроля эффективности рабочего времени. Для анализа рассмотрим такие приложения, как Yaware.TimeTracker, CrocoTime, StaffCop Enterprise и Motivation Clock. Yaware.TimeTracker и Motivation Clock предоставляют доступ к личной статистике использования рабочего времени, в отличие от других приложений, но в то же время приложение Motivation Clock является исключительно персональным приложением. CrocoTime представляет анализ эффективности рабочего времени на уровне подразделений и отдельных сотрудников. Yaware.TimeTracker и CrocoTime обладают различными уровнями доступа для сотрудников и руководителей, а StaffCop Enterprise содержит в себе только основные возможности, которые наблюдаются во всех рассматриваемых приложениях, а именно: мониторинг работы с программами и сайтами, разделение времени на продуктивное и непродуктивное и визуальный анализ данных. Исходя из рассмотренных особенностей, ниже представлена табл. 1 со сравнением характеристик приложений.

Таблица 1. Сравнение характеристик приложений

Характеристика	Название приложения			
	Yaware.Time Tracker	CrocoTime	StaffCop Enterprise	Motivation Clock
Мониторинг работы с программами и сайтами.	+	+	+	+
Разделение времени на продуктивное и непродуктивное	+	+	+	+
Визуальный анализ данных	+	+	+	+
Доступ к личной статистике	+	-	-	+
Различные уровни доступа для сотрудников и руководителей.	+	+	-	-
Анализ эффективности на уровне подразделений и отдельных сотрудников	-	+	-	-

Определение инструментальных средств разработки клиент-серверного приложения. В приведенной ниже табл. 2 отмечено наличие или отсутствие тех или иных возможностей в некоторых популярных сегодня языках программирования.

Таблица 2. Короткое сравнение языков программирования

Характеристика	C++	Java	PHP	Perl
Скорость работы модулей	высокая	высокая	средняя	средняя
Кроссплатформенность	да	да	да	да
Надежность	низкая	высокая	высокая	высокая
Безопасность	низкая	высокая	средняя	высокая
Масштабируемость	?	высокая	?	?
Готовые библиотеки	много	много	много	много

Исходя из сравнения в таблице, лучшим решением является использование языка Java. Одно из главных его преимуществ – независимость от платформы, на которой выполняются программы. Таким образом, один и тот же код можно запускать под управлением операционных систем Windows, Linux, FreeBSD, Solaris, Apple Mac и др. Другим, не менее важным преимуществом Java, является большая схожесть с языком программирования C++. Поэтому те программисты, которые знакомы с синтаксисом C и C++ просто осваивают и Java. Кроме того, Java – полностью объектно-ориентированный язык, даже в большей степени, чем C++. Все сущности в языке Java являются объектами, за исключением немногих основных типов.

Для серверной части рациональным является использование Tapestry – объектно-ориентированного Java фреймворка для создания веб-приложений. Основной акцент в Tapestry сделан на продуктивность разработчика и лёгкость использования. Tapestry использует модульный подход к веб-разработке, используя биндинг между компонентами пользовательского интерфейса на веб-странице и соответствующими им java-классами [1].

PostgreSQL – полноценная SQL СУБД с большим списком возможностей, которая является кроссплатформенным продуктом и работает не только в широком диапазоне диалектов UNIX (Linux, FreeBSD, Solaris и т.д.), но и на платформе Microsoft Windows. PostgreSQL поставляется под лицензией BSD, что обеспечивает максимальную открытость и доступность. Для работы с PostgreSQL существует множество интерфейсов и библиотек взаимодействия из других языков программирования: Java, Python, Ruby, C, C++, PHP, Lisp, Scheme и Qt [2].

Принцип работы приложения в виде блок-схем и основных XML запросов и ответов сервера. Принцип работы клиент-серверного приложения контроля эффективности рабочего времени можно представить с помощью следующих блок-схем:

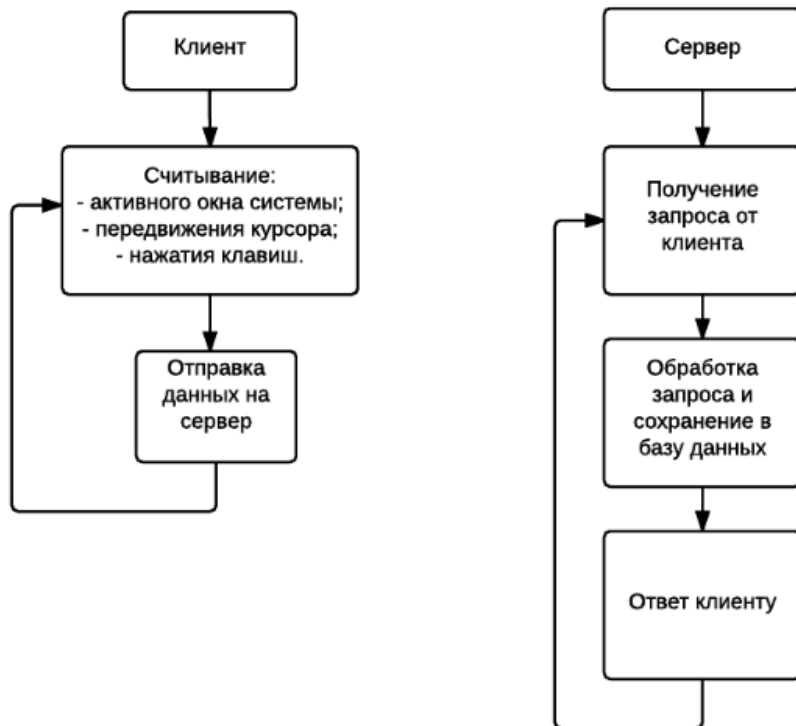


Рис. 1 – Блок-схемы клиент-серверного приложения

Как можно видеть на рис. 1, после запуска клиента начинается постоянное считывание активного окна операционной системы, нажатий клавиш и передвижений курсора. Для сохранения данных и её визуализации каждую минуту собранная информация передается на сервер с помощью следующего XML-запроса:

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE saveData>

<saveData>
  <activeWindows>
    Word.exe 30 секунд
    Photoshop.exe 10 секунд
    Chrome.exe vk.com 20 секунд
  </activeWindows>
  
```

```
<cursorMoved>
  true
</cursorMoved>
<buttonPress>
  true
</buttonPress>
<clientTime>
  1461706723
</clientTime>
</saveData>
```

Сервер принимает информацию, основываясь на IP-адресе клиента, и сохраняет её в базу данных. После этого отправляет XML-ответ в следующем формате:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE saveDataResponse>

<saveDataResponse>
  <status>
    20
  </status>
  <statusDescription>
    null
  </statusDescription>
</saveDataResponse>
```

Разработка клиент-серверного приложения контроля эффективности рабочего времени. Исходя из изученных особенностей клиент-серверной архитектуры и анализа существующих приложений контроля эффективности рабочего времени, разработка программного кода не составит труда. Дальнейшая разработка приложения будет представлена автором в виде дипломного проекта.

Выводы. Произведен анализ существующих приложений контроля эффективности рабочего времени. Определены инструментальные средства разработки клиент-серверного приложения. Изучены основные принципы работы приложения и представлены в виде блок-схем и XML запросов и ответов сервера. В результате изучены основные особенности автоматического учета эффективности рабочего времени для разработки соответствующего клиент-серверного приложения для сотрудников предприятий.

Список литературы

1. Язык программирования [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2016]. URL: www.wikipedia.org/wiki/Язык_программирования Что такое PostgreSQL? [электронный ресурс] // Виктор Вислобоков [сайт]. [2008-2010]. URL: http://postgresql.ru.net/docs/about_pgsql.html

УДК 003.628

Использование семиотики в печатной рекламе. Семиотика как средство выразительности в печатной рекламе

Пересекин А.В., Харитонова В.В.

Донецкий национальный технический университет, г. Донецк

кафедра компьютерного дизайна и моделирования

E-mail: alexperesekin@gmail.com

Аннотация

Пересекин А.В. Харитонова В.В. Использование семиотики в печатной рекламе. Изучено понятие семиотики, исследована семиотика в печатной рекламе, проведен анализ единиц языка семиотике в постере

Ключевые слова: семиотика, реклама, знак, символ, индекс.

Актуальность исследования. На сегодняшний день рекламное сообщение является самым востребованным способом распространения информации. Известно, что интенциональное рекламное сообщение обладает большим психологическим воздействием на общество и его отдельных представителей. Поэтому грамотное построение системы знаков в рекламном тексте, исследование способов их взаимодействия и влияния, позволит добиться максимально эффективного воздействия любого рекламного продукта на потребительскую аудиторию.

Цель исследования. Изучить семиотические средства рекламы. Выявить особенности использования знаков в печатной рекламе.

Понятие семиотики

Семиотика - наука о знаках и знаковых системах. Эта междисциплинарная наука возникла на стыке лингвистики, теории информации, психологии, биологии, литературы, социологии. В общей семиотике выделяются три раздела:

– синтактика — изучение объективных законов устройства знаковых систем, а также отношения между элементами языка, правилами их образования и сочетания;

– семантика — изучение смыслов, т.е. отношений между знаками и значениями, правилами обозначения и понимания;

– прагматика — изучение отношения субъектов, использующих знаковую систему, к этой системе, иными словами, отношения между языком и пользователем и правилами употребления языка.

Семиотика культуры — это знаковые средства культуры, а также рассмотрение всех культурных феноменов как текстов.

Вторичные моделирующие системы называют языками культуры или культурными кодами. К ним относят все виды культурных текстов, кроме естественного языка, искусство, социальную деятельность, модели поведения, традиции, обычаи, религиозные верования и т.д.

С точки зрения семиотики основной структурной единицей языка культуры являются знаковые системы.

Знак — это материальный предмет (явление, событие), объективно заменяющий некоторый другой предмет, свойство или отношение и используемый для приобретения, хранения, переработки и передачи сообщений (информации, знаний). Выделяют шесть типов знаков и знаковых систем: естественные, функциональные, иконические, конвенциональные, вербальные, системы записи.

Сигналы - знаки извещения или предупреждения, например цвета светофора.

Индексы - условные обозначения предметов или ситуаций, имеющие компактный вид и применяемые для того, чтобы выделить эти предметы и ситуации из ряда других.

Символы — знаки, не только указывающие на некоторый объект, но и несущие в себе добавочный смысл. Примерами символов являются государственные гербы, флаги, гимны — символические знаки достоинства государств.

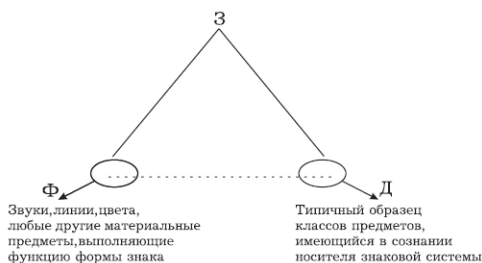


Рис. 1. – схема «треугольник Фреге»

Стрелки, «оттягивающие» противоположные углы основания треугольника в разные стороны, схематически означают крайнюю отдаленность формы (Ф) и денотата (Д), конвенционально связанных друг с другом только через значение (З) – вершину треугольника.

Символы имеют не только конвенциональный, но и иконический характер. Означающая сторона символа всегда каким-то образом связана с тем, что она обозначает, имеет с ним какое-то сходство, иногда очень опосредованное, ассоциативное. Например, весы могут быть символом справедливости, так как содержат идею равновесия. [1]

Семиотика в рекламе

Реклама в жизни современного человека выполняет следующие функции:

- ознакомить или напомнить. Обозначив продукт;
- разъяснить, показав привлекательность товара;
- одобрить, представив эмоциональные преимущества;
- убедить приобрести товар.

Все функции реализуются с помощью выполнения конкретных речевых задач.

Разную функциональную значимость имеют структурные элементы рекламы:

- заголовок – устанавливает контакт между читателем и основным текстом;
- подзаголовок – развивает идею, заложенную в заголовке;
- основной текст поясняет, для чего служит товар, доказывает его эффективность, завершается призывом;
- подписи и комментарии – содержат дополнительную информацию;
- рекламный лозунг – служит гарантом успеха; он обладает особой воздействующей силой.

Можно выделить следующие наиболее существенные знаки рекламы:

- фотография;
- сочетание слова и рисунка (сложный знак);
- язык шрифтов:
 - a) более крупный шрифт предназначается пожилым и юным покупателям;
 - b) для лучшего восприятия текста используется шрифт с засечками, более широкий интерлиньяж, неровный (рваный) правый край текста.[2]

Изображенное сообщение подвергается анализу для выявления всех возможных содержащихся в нем частных конкретных сообщений.

Предположим, рекламируются итальянские макароны фирмы «Пандзани». Реклама представляет собой две пачки макарон, банку с соусом, пакетик сыра-пармезана, помидоры, лук, перец, шампиньоны – и все это выглядывает из раскрытой сумки для провизии. Картина выдержана в бело-зеленых тонах, фон – красный. Выделим те сообщения, которые содержатся в этой рекламе.



Рис. 2. – пример использования семиотики в рекламных постерах

Первое сообщение имеет языковую природу и дано непосредственно: оно образовано подписью под рекламой, а также надписями на этикетках, включенных в изображение на правах своего рода «эмблем»: код этого изображения есть не что иное, как код естественного (русского) языка. Однако это языковое (символическое) может быть расчленено, поскольку в знаке «Пандзани» содержится не только название фирмы, но и – благодаря звуковой форме этого знака – еще одно, дополнительно значение, которое можно обозначить как «итальянский стиль», или «итальянскость».

Второе сообщение, заложенное в данную рекламу, предполагает отвлечение от первого, языкового, сообщения, и мы окажемся перед изображением как таковым (включая этикетки).

Третье, тоже иконическое сообщение основано на том, что в этом изображении содержится целый ряд дискретных знаков, прежде всего они вызывают представление о «походе на рынок». Означающее «поход на рынок» в свою очередь предполагает наличие двух эмоциональных представлений – представление о свежих продуктах и о домашнем способе их приготовления.

Наличие **второго знака** тоже очевидно. Его означающим служат помидоры, перец и трехцветная раскраска рекламной картинке, а означаемым – страна Италия, точнее – качество «итальянскость». Этот знак даже избыточен по отношению к коннотативному знаку языкового сообщения, каким является итальянское звучание слова «Пандзани».

Третий знак в составе рассматриваемого второго иконического сообщения благодаря изображению вперемешку самых разнородных продуктов, подсказывает мысль о комплексном обслуживании: он убеждает, что, с одной стороны, фирма

«Пандзани» (Panzani) способна поставлять все, что необходимо для приготовления сложного блюда, а с другой – что баночный концентрат соуса не уступает по своим качествам свежим продуктам, в окружении которых он изображен.

Что касается **четвертого знака**, то сама композиция рекламы заставляет вспомнить о множестве картин, изображающих всякую снедь, и тем самым отсылает к эстетическому означаемому: перед нами «натюрморт», т. е. «мертвая природа».

К выделенным нами четырем дискретным знакам в составе второго иконического сообщения можно добавить еще один, **пятый знак**, указывающий, что мы имеем дело именно с рекламой, а не с чем-либо другим. [3]

Реклама как креолизованный текст. В современной коммуникативной среде естественный язык является важнейшим и универсальным средством общения, но совсем не единственным. Термин «креолизованный», «креолизация» отсылает к буквальному его значению: **креолизованный язык** – язык, образовавшийся благодаря взаимодействию двух языков – коренного местного и колониального английского – и ставший основным средством общения в данном коллективе, например, новомеланезийский язык, образовавшийся при взаимодействии английского языка с меланезийским.

Сообщение, заключенное в тексте, может быть, в частности, представлено вербально (словесный текст) или иконически, т. е. изобразительно.

Примерами креолизованных текстов могут служить тексты радиовещания и телевидения, средства наглядной агитации и пропаганды, плакаты, ориентированные на аудиторию – от узкопрофессиональной до широкой, и прежде всего – рекламные тексты.



Рис. 3. – пример креолизованного текста

Метафора в креолизованном тексте

Важным элементом, «скрепляющим» вербальную и иконическую стороны креолизованной рекламы, является такой стилистический инструмент, как метафора. **Метафора** – слово или выражение, которое употребляется в переносном значении на основе сходства в каком-либо отношении двух предметов или явления для большей образности: перенесение на один предмет (явление) характерных признаков другого предмета (явления). Метафора

содержит только то, с чем сравнивают товар, что значительно расширяет емкость, убедительность и выразительность образа-анalogии, который через рекламный текст увязывается с рекламируемым товаром.

Красивые метафорические включения способны изящно и кратко поведать о таких, на первый взгляд, сухих и скучных вещах как параметры товара, особенности его применения, а также о самой фирме, ее духе и имидже. [4]

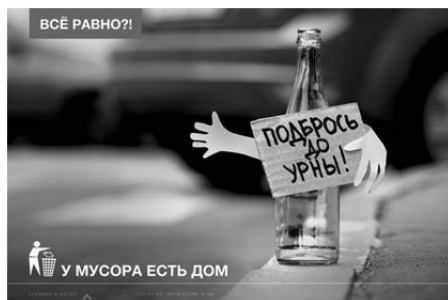


Рис. 4. – метафора в семиотике

Вывод. Рассмотрено понятие семиотики и перечислено 3 раздела – синтактика, прагматика и семантика; также рассмотрены основные единицы семиотики – знаки, сигналы, индексы и символы. Проиллюстрирована схема Фреге зависимости формы, денотаты и значения. Изучены семиотические средства рекламы. Выявлены особенности использования знаков в печатной рекламе. Проведен анализ рекламного сообщения-изображения на примере рекламы макарон «Пандзани». Исследовано понятие креолизованного текста в рекламе, с помощью словесного текста или иконически. Рассмотрена метафора, как основная часть креолизованного текста, которая окрашивает предмет, повышает убедительность и выразительность образа.

Список источников

1. Знаки и знаковые системы в культуре [Электронный ресурс] // Энциклопедия Экономиста [сайт]. URL: <http://www.grandars.ru/college/sociologiya/semiotika-kultury.html>
2. Семиотика и реклама [Электронный ресурс] // Студопедия.Орг – поиск лекций и конспектов [сайт]. URL: <http://studopedia.org/1-93033.html>
3. Семиотический анализ рекламного сообщения [Электронный ресурс] // Libma – библиотека обучающей и информационной литературы [сайт]. URL: http://www.libma.ru/delovaja_literatura/semiotika_reklamy/p8.php
4. Семиотика рекламы [текст]: учебное пособие / Е.А. Елина, 2008. – 140 с.

УДК 004.02

Основные требования к расписанию и методы его составления в условиях высшего учебного заведения

Лоленко В.А., Хоруженко А.С.

Донецкий национальный технический университет, г. Донецк

кафедра компьютерного моделирования и дизайна

E-mail: viktoria.lol@mail.ru

Аннотация

Лоленко В.А., Хоруженко А.С. Основные требования к расписанию и методы его составления в условиях высшего учебного заведения. Статья посвящена наиболее распространенным требованиям к расписаниям учебных занятий вуза. Проведены анализ и сравнение методов составления расписаний.

***Ключевые слова:** расписание, требования, классические методы решения задач, эвристические методы.*

Постановка проблемы. В процессе составления расписания учебного процесса часто возникает проблема с возникновением неточностей и конфликтных ситуаций. Таким образом, возникает необходимость определить требования к расписанию и выбрать такой метод его составления, который позволил бы уменьшить вероятность возникновения конфликтов до минимума. Для реализации этого необходимо выполнить следующее:

- определить само понятие расписания, критерии его оптимальности;
- определить требования, согласно которым будет разрабатываться расписание;
- провести анализ методов решения подобной задачи и выбрать тот из них, который позволит получить максимальную оптимальность расписания при минимальной вероятности возникновения конфликтов.

Цель статьи – провести анализ методов решения задачи составления расписаний и определить набор требований, которые позволят составить его наиболее оптимально, и метод, который будет наиболее приемлемым для его составления.

Составление расписания – одна из наиболее распространенных задач в планировании и оптимизации учебного процесса в образовательных учреждениях. От того, насколько хорошо составлено расписание, зависит эффективность работы преподавателей, усвоение учебного материала студентами, рациональное использование интеллектуальной и материальной баз вуза.[1]

С экономической точки зрения можно утверждать, что при оптимальном составлении расписания занятий уменьшатся постоянные затраты учебного заведения на электроэнергию, тепло, водоснабжение и т.п., поскольку

эффективность использования ресурсов учебного заведения, таких как аудиторный фонд, вычислительная техника, оборудование лабораторий, значительно повысится. [2]

Само по себе расписание учебных занятий – это документ, регламентирующий работу студентов, преподавателей, всего учебного заведения, распределяющий содержание учебного плана и всех учебных программ по календарным дням учебного года и обеспечивающий их реализацию.

Все требования можно подразделить на 2 группы: основные и дополнительные, и на несколько категорий: методические, организационные и требования по обеспечению самостоятельной работы студентов. [3,4]

К основным можно отнести:

- вместительность аудиторий, которая должна быть достаточной для групп, которые в ней занимаются, при этом возможен вариант, когда в одной аудитории проводятся занятия одновременно для нескольких групп студентов в потоке;

- обязательное выполнение требований к оборудованию аудиторий, в которых проводятся занятия: телевизионная аудитория, размер доски, специализированные стенды и т. п.;

- условие, что преподаватели из других вузов могут проводить занятия только в определенные дни и часы.

- К дополнительным можно отнести:

- проведение лекций в начале дня, а практик – в конце;

- отсутствие окон в расписании занятий студентов, и в то же время возможность наличия окна в расписании преподавателя;

- предоставление преподавателям хотя бы одного дня, свободного от аудиторных занятий в вузе;

- количество занятий в пятницу должно быть меньше, чем в остальные дни недели;

- отсутствие, по возможности, первого занятия в понедельник утром. [4]

Обзор групп методов составления расписания. Так как задача составления расписания в общей формулировке относится к классу NP-полных задач, большинство методов ее решения основаны на разных эвристиках. [5]

Все известные работы в этом направлении можно условно разделить на 2 группы: к первой группе относятся работы, использующие классические методы решения задачи составления расписания. К ним относятся методы динамического программирования, целочисленного программирования, нелинейного программирования, методы имитационного моделирования: ветвей и границ, метод раскраски графа, задача о назначениях и другие. Вторая группа работ основана на современных методах решения задач целочисленного программирования, использующих эвристические алгоритмы решения данных задач: генетические алгоритмы, метод муравьиных колоний, локально-эволюционный метод. [2,5]

Классические методы. Специфической характеристикой классических методов является полная наибольшая степень официализации (математическая точность) как положение задачи составления расписания занятий, так и алгоритмов ее реализации (обращаются «жесткие» алгоритмы). [2, 7]

Особенности:

- возможность разрабатывать итеративные методы решения задачи составления расписания учебных занятий, обладающие приемлемым временем сходимости и точностью решения;

- возможность оценивать влияние на время и точность решения задачи интересующих нас факторов.

Недостатки:

- использование в своей основе итерационной процедуры поиска или улучшения некоторого начального приближения (опорного плана расписания); как следствие, результат зависит от некоторого начального приближения, и возникает проблема выбора его значения; [6]

- громоздкость и сложность получаемой математической модели и резкий (экспоненциальный) рост временных затрат на поиск приемлемого решения с ростом объемов исходной информации. [6]

Эвристические методы. В положении эвристических методов, как правило, лежит использование разнообразных видов эвристик, или эвристических алгоритмов, при подготовке которых используются интуитивные предположения, не подтвержденные соответствующим математическим обоснованием. [2, 7]

Особенности:

- Универсальность. Эвристические методы применяются во всех сферах науки и практики, можно даже сказать, что овладение эвристическими методами определяет искусство анализа. [8]

- Разнообразность. На практике не представляется возможным выработать какую-то общую схему разработки эвристических методов, поскольку, как указано выше, они могут применяться во многих сферах науки и практики – от программирования до управления предприятиями и анализа хозяйственной деятельности.

В программировании они используются в сочетании с методами перебора с целью сократить количество проверяемых вариантов. Некоторые варианты согласно выбранной эвристике изначально считаются бесперспективными и не проверяются, что позволяет значительно ускорить работу алгоритма по сравнению с методом полного перебора. [9]

Недостатки: нахождение только лишь приближенного решения (локального экстремума). В связи с этим возникает проблема поиска и сопоставления полученного результата с глобальным экстремумом. [6]

Несмотря на указанный недостаток, эвристические алгоритмы продолжают оставаться наиболее эффективным алгоритмом поиска «лучшего»

в определенном смысле решения в тех случаях, когда нахождение оптимального наилучшего решения практически затруднительно или невозможно. [6,7]

Генетический алгоритм. Генетические алгоритмы — современные интенсивно развивающиеся методы решения тех задач целочисленного программирования, которые отличаются относительно большой размерностью. Существенное отличие и превосходство генетических алгоритмов по сравнению с методами из группы классических выражается в следующем:

- генетический алгоритм работает с командами, в которых представлен список параметров, зависящих напрямую от аргументов целевой функции;

- генетический алгоритм в процессе поиска использует некоторое количество точек поискового инструмента. То есть можно сказать, что процесс распараллеливается, а не переключается от точки к точке, как это выполняется в классических методах. Таким образом, можно сделать вывод, что генетический алгоритм оперирует со всей совокупностью допустимых решений;

- генетический алгоритм в процессе работы не использует никакой вспомогательной информации, что значительно увеличивает скорость его реализации и сокращает время на его выполнение;

- генетический алгоритм применяет в равной мере как вероятность установки для создания новых точек поиска, так и детерминированные установки для переключения от одних точек к другим.

- В целом, можно сказать, что генетический алгоритм является наиболее эффективным при составлении расписания, поскольку он позволяет устранять большинство из возможных недостатков. [6,7]

Выводы. В рамках данной работы были проанализированы основные требования к расписаниям учебных заведений и рассмотрены основные группы методов их составления. Как утверждают авторы работ [2, 6, 12], расписания, которые получены в результате использования классических методов, обычно обладают большим количеством недостатков, чем, к примеру, расписания, созданные по генетическому алгоритму.

Список литературы

1. С.П. Семенов, Я.Б. Татаринцев Сравнительный анализ подходов к автоматизации составления расписаний учебных занятий в образовательных учреждениях // Известия Алтайского государственного университета // Выпуск №65 / том 1 / 2010

2. А. Б.Сидорин, Л. В.Ликучева, А. М.Дворянкин Методы автоматизации составления расписания занятий. Часть 1. Классические методы // Известия Волгоградского государственного технического университета // Выпуск № 7 / том 12 / 2009

3. В.П. Ерунов, И.И. Морковин Формирование оптимального расписания учебных занятий в ВУЗе // Вестник Оренбургского государственного университета // Выпуск № 9 / том 3/ 2001

4. Ю.В. Береговых, Б.А. Васильев, Н.А. Володин Алгоритм составления расписания занятий // Наукова електронна бібліотека періодичних видань НАН України // Выпуск №2/Штучний інтелект/ 2009

5. А.М. Дворянkin, В.С. Чальшев Обзор методов составления расписания ВУЗов // Известия Волгоградского государственного технического университета // Выпуск № 11 / том 9 / 2011

6. А. Б. Сидорин, Л. В. Ликучева, А. М. Дворянkin Методы автоматизации составления расписания занятий. Часть 2. Эвристические методы оптимизации // Известия Волгоградского государственного технического университета // Выпуск № 7 / том 12 / 2009

7. О. В. Свиридова, А. В. Михайлов Обобщенный анализ методов проектирования и разработки автоматизированного составления расписания образовательных учреждений // Молодой ученый // 2013. Выпуск №6 / с. 214-216 / 2013

8. Характеристика эвристических способов и приемов анализа хозяйственной деятельности [электронный ресурс]. // Студенческая библиотека онлайн: [сайт]. [2016]. URL: http://studbooks.net/53643/ekonomika/harakteristika_evristicheskikh_sposobov_priemo_v_analiza_hozyaystvennoy_deyatelnosti

9. Эвристические методы [электронный ресурс]. // Основы программирования: [сайт]. [2016]. URL: <http://www.kufas.ru/programming173.htm>

10. Метод роя частиц [электронный ресурс]. // Википедия – свободная энциклопедия: [сайт]. [2001-2016]. URL: https://ru.wikipedia.org/wiki/Метод_роя_частиц

11. Кравчук Д. Алгоритм роя частиц [электронный ресурс]. // Хабрахабр: [сайт]. [2006-2016]. URL: <https://habrahabr.ru/post/105639>

12. В.В. Яковлев, Ю. Г. Васильев, Б.М. Калмыков Решение задачи составления расписания занятий в вузе с помощью нейроподобных сетей // Вестник Чувашского университета // Выпуск №2 / 2008

УДК 004.92

**Некачественный дизайн как одна из проблем юзабилити сайта
Ефремченко И.А., Харитонов В.В.**

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
efr-inessa@yandex.ru

Аннотация

Ефремченко И.А., Харитонов В.В. Некачественный дизайн как одна из проблем юзабилити сайта. В статье рассмотрены основные проблемы юзабилити сайтов. Проанализированы основные ошибки и проблемы, которые допускаются в дизайне сайтов. Также рассматриваются правила построения интерфейсов.

Ключевые слова: дизайн, цвет, шрифт, web-система, проблемы юзабилити сайтов, удобство использования сайтов, юзабилити сайтов.

Постановка проблемы. Основная цель разработчиков интерфейсов – спроектировать интерфейс так, чтобы пользователю было легко и удобно им пользоваться. Из-за большого количества неквалифицированных дизайнеров интерфейса сайтов потребитель сталкивается с неудобными для использования сайтами. Именно юзабилити определяет правила для проектирования дизайна web-системы. Таким образом, появляется необходимость рассмотреть проблемы, которые могут возникнуть в процессе создания дизайна сайта.

Цель статьи. Провести анализ распространенных проблем, встречающимися при проектировании интерфейсов. Рассмотреть основные правила, используемые при создании дизайна.

Основные понятия. Качество информационной web-системы зависит от большого количества факторов, которые в какой-либо степени влияют на удобство использования системы в целом конечным пользователем. На этапе оценки данного параметра вводится базовое понятие «юзабилити» (от англ. usability).

Юзабилити - степень, с которой продукт может быть использован определёнными пользователями при определённом контексте использования для достижения определённых целей с должной эффективностью, продуктивностью и удовлетворённостью.

Согласно стандарту ISO-9241 основными критериями юзабилити являются:

– Обучаемость: как легко пользователи Вашего сайта могут выполнить основные задачи, впервые встретившись с незнакомым им интерфейсом?

– Эффективность: как только пользователи Вашего сайта ознакомятся с дизайном, как быстро они могут выполнить то же задание?

– Запоминаемость: После того, как пользователь Вашего сайта снова вернется к главному через определенный промежуток времени, как легко ему будет восстановить свои "навыки" в работе с данным интерфейсом?

– Ошибки: сколько ошибок делает пользователь, как серьезны эти ошибки, насколько легко пользователь может исправить эти ошибки?

– Удовлетворенность: насколько пользователю приятно пользоваться данным интерфейсом? [1].

Одним из главных моментов при работе с поведенческими факторами на сегодня является удобство использования ресурса. Сайт, который учитывает правила юзабилити, воспринимается посетителями лучше и имеет достаточный показатель времени, которое было проведено на сайте. Все это способствует работе пользовательских факторов на сайт, подъему позиций сайта в поисковых системах и привлечению посетителей.

Определение основных проблем юзабилити сайтов. Несмотря на то, что web-системы основаны на простых элементах, которые формируют интерфейс, а именно: ссылки, текстовые поля, текст, графика, меню и кнопки, проблемы в юзабилити неизбежны и имеют общий характер для многих проектов в Интернете. К проблемам юзабилити следует отнести:

Дизайн сайта.

Структура сайта и организация страниц.

Навигация.

Контент.

Дизайн сайта. Дизайн сайта — понятие более сложное, чем просто внешнее оформление сайта. Под дизайном сайта понимается удобство web-системы для пользователя — насколько выделены нужные для постоянного применения элементы, легко ли читается текст и т.д. [2].

От дизайна зависит его будущее, так как именно внешний вид в основном строит показатели поведенческих факторов, которые влияют посещаемость.

Рассмотрим основные ошибки и проблемы, которые допускаются в дизайне сайтов:

– Выбор цвета. Большинство людей считает, что цвет сайта, не играет важной роли при создании дизайна и они сильно ошибаются. В веб-дизайне цвет так же важен, как и структура сайта и навигация. Цвета могут привлекать внимание и стимулировать пользователя на совершение конкретного действия на сайте. Все, что связано с цветом, начиная от фона сайта и цвета кнопок и заканчивая логотипом, может оказать огромное влияние на впечатление пользователей от сайта.

Так например, области, фон которых представлен более теплыми оттенками красной части спектра, кажутся крупнее, чем области, цвет фона

которых смещен к голубой части. Объекты экрана на белом фоне или на фоне цвета из средней части спектра кажутся ярче и легче воспринимаются при разном внешнем освещении. Наибольший контраст между двумя областями экрана достигается в том случае, если цвет фона одной из них — черный или близок к любой из границ спектра, а цвет фона другой — белый или взят из средней части спектра. Данные рекомендации справедливы и для соотношения между цветами символов и фона. Следует учитывать, что некоторые комбинации цвета, например, голубой цвет символов на красном фоне, неприятны для глаз [3].

У каждого цвета есть свое эмоциональное значение. Зачастую цвета имеют свое значение для разных людей, но есть и распространенные представления о том, с чем ассоциируется конкретный цвет у большинства людей. Рассмотрим наиболее распространенные цветовые ассоциации:

- голубой — успокаивает;
- красный — волнует и утомляет;
- зеленый — настраивает на добродушный и безынициативный лад;
- желтый — веселый, оптимистичный, вызывает легкомысленный настрой;
- оранжевый — раскрепощает фантазию;
- фиолетовый — губителен для глаз, цвет зависти, тревоги, неудовлетворенности;
- коричневый — угнетает умственную активность;
- черный — мрачный, способствует возникновению головных болей, но снижает число ошибок.

Таким образом, непродуманное использование цвета может вызвать у пользователя негативное эмоциональное состояние, даже отвлекать его от работы, мешая сосредоточиться на выполняемом задании.

Основным принципом дизайнера интерфейса должно быть осторожное использование цветов. Есть несколько правил эффективного использования цвета в пользовательских интерфейсах. Ниже представлены наиболее важные из них:

1. Необходимо использовать ограниченное количество цветов. В интерфейсе web-системы должно быть не более семи цветов.

2. Необходимо использовать контрастные цвета при разработке дизайна. Так как экран ограничен в отображении цветов, то разница между близкими оттенками цвета может стать совершенно неразличимой на экране монитора.

3. Яркие цвета необходимо использовать на нейтральном, не ярком фоне. Яркие (вибрационные) комбинации цвета, создают иллюзию того, что они вибрируют на экране. Такие сочетания цветов неудачны и могут вызвать у пользователя головную боль.

4. Необходимо избегать чистых цветов. Белый цвет получается в результате максимального свечения монитора. И чем это свечение интенсивнее, тем сложнее пользователю долгое время воспринимать визуальную страницу сайта.

Чем менее резким будет контраст между текстом и фоном, тем проще пользователю воспринимать информацию.

5. Серый цвет воспринимается как цвет неактивного элемента. Поэтому необходимо избегать его использование для кнопок и ссылок.

6. Использование смены цвета в зависимости от состояния объекта улучшает восприятие пользователем.

Типографика. Большому количеству интерфейсов присущи проблемы, которые касаются шрифтового оформления, выравнивания и интервалов. Подобно другим визуальным элементам, шрифты способствуют организации информации и созданию определенного настроения.

Помимо утилитарной функции визуализации текста шрифт играет эстетическую и психологическую роль, оказывая влияние на все аспекты взаимодействия с пользователем. Красивый шрифт, отвечающий содержанию и эстетической задаче сайта, легко и приятно читать, в то время как некачественный или не удачно подобранный шрифт превращает чтение в испытание и оставляет у посетителя неприятное впечатление [4].

Поскольку типографика важна при создании сайтов, существуют определенные правила ее использования:

1. Семейство шрифтов. Подбор шрифта начинается с выбора гарнитуры шрифта. Выбор гарнитуры шрифта зависит от процесса чтения и восприятия текста. Для основного текста рекомендуется использовать шрифт без засечек. Так как при использовании шрифта с засечками, при отображении текста на дисплее устройства, возникает проблема сглаживания и неровности засечек.

2. Размер шрифта. Размер базового шрифта в браузере равен 16px. Разница в размере шрифта текста и заголовка должна отличаться не более чем в полтора раза. Изменяя размер шрифта, можно придать тексту визуальную значимость и привлечь внимание посетителей к наиболее важным фрагментам текста.

3. Выравнивание текста. В web-типографике используется выравнивание текста по левой стороне. Стандартное выравнивание по ширине образует промежутки между словами, которые неприятны при чтении информации.

4. Межбуквенный, межсимвольный и межстрочный интервал. Межбуквенный интервал оказывает влияние на читабельность и восприятие текста. Чем меньше шрифт, тем дальше друг от друга должны стоять буквы, чем крупнее — тем плотнее должны быть расположены буквы.

Оптимальное значение межстрочного интервала в 1.4 — 1.6 раза больше размера шрифта.

5. Длина строки. В строке для сплошного чтения должно помещаться от 30 до 75 знаков (7-10 слов на одной строке). Чем шире строка текста, тем больше должен быть межстрочный интервал. Межстрочное расстояние не должно быть меньше пробела между словами.

6. Начертание шрифта. Управляя начертанием шрифта, можно влиять на восприятие текста. Так, например, курсив придает тексту торжественность. Жирное начертание плюс увеличенный размер шрифта позволяет выделить нужное содержимое.

Flash-анимация. Flash -анимация — это анимация, которая является элементом дизайна web-системы. В основном применяется для привлечения внимания пользователя. Когда flash-анимацию используют при создании сайта, ставка делается на то, что у человека самый развитый орган восприятия и получения информации — это глаза. Считается, что сведения понимаются и запоминаются лучше, если они носят динамический характер. Утверждают, что при использовании flash-роликов реклама становится более эффективной, нежели статические графические изображения. Несомненно, доля истины здесь есть, но сейчас разработчики web-систем считают, что обращаться к flash-анимации нужно только в том случае, когда задача не решается другими средствами [5].

При использовании flash-анимации необходимо чувствовать меру. Не следует использовать анимацию для украшения сайта, а также размещать её на каждой странице сайта. Анимация не должна перегружать сайт, так как зачастую это сказывается на нагрузке сервера, что замедляет работу сайта.

Некоторые типы браузеров не поддерживают plug-in модули, значит — пользователь не будет иметь возможности в полной мере наслаждаться элементами flash на сайте, а сайт, целиком реализованный на технологии web-систем, интегрированной с flash-модулями может быть и вовсе недоступен большинству пользователей, которые используют настройки своего браузера «по умолчанию».

Визуальный шум. Визуальный шум возникает из-за большого количества навигационных элементов, множества броских неструктурированных картинок, горизонтальной прокрутки в пределах одного экрана. При наличии шума важные элементы сайта теряются на его фоне. Чтобы избежать этой ошибки, на стадии разработки макета сайта стоит выбирать сбалансированный дизайн.

Рассмотрим основные ошибки в интерфейсах сайтов, которые касаются визуального шума:

1. Расстановка акцентов. Существует два вариант. Либо на странице отсутствуют какие-либо акценты, в таком случае страница становится скучной и неинтересной пользователю. Либо на странице множество акцентов, которые в совокупности теряются, в таком случае пользователь теряется в нужной ему информации.

2. Группировка объектов по смыслу. Связанные части необходимо объединять вместе. Это позволит пользователю лучше воспринимать информацию и сделает дизайн страницы яснее.

Примеры ресурсов с высоким юзабилити. Для того, чтобы лучше понять суть юзабилити и усвоить описанные выше правила, необходимо рассмотреть примеры сайтов с высоким юзабилити.

1. Google Play Store. Интернет-магазин Google Play Store предлагает пользователю большое количество программных продуктов (см. рис. 1а).

Дизайн ресурса прост и привлекателен. Экран свободен от лишней информации, что частично связано с расположенным в его левой части навигационным меню. Найти пользователю нужный программный продукт информации не составит труда и займет всего пару кликов.

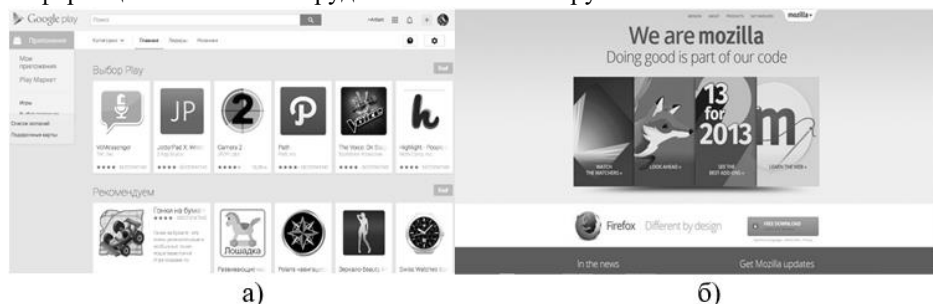


Рис. 1 – Интерфейс: а – интернет-магазин Google Play Store, б – сайт компании Mozilla Corporation

2. Официальный сайт компании Mozilla Corporation является простым и удобным пользователю (см. рис. 1б). Выявив основные цели своей целевой аудитории, Mozilla разместила необходимую им информацию (сведения о компании и продуктах) прямо на домашней странице. Браузер можно скачать тут же.

Выводы. Проанализированы проблемы дизайн как одна из проблем юзабилити. Сделан вывод, что некачественный дизайн снижает удобство пользования сайтом, усложняет процесс поиска и усвоения информации. Рассмотрены ошибки, которые допускаются при создании пользовательского интерфейса. Приведены правила, которые позволяют улучшить юзабилити сайта. Такие улучшения смогут удовлетворить потребности пользователя

Список литературы

1. Юзабилити - общие понятия [электронный ресурс] // Документация AOW: [сайт]. [2005-2016]. URL: <http://doc.artofweb.ru/>
2. Дизайн сайта, что это такое? [электронный ресурс] // Веб-студия Михаила Христенко.: [сайт]. [2015]. URL: http://www.hms-web.ru/index.php?p=articles_design

3. Основные принципы разработки пользовательского интерфейса [электронный ресурс] // Pandia: [сайт]. [2009-2016]. URL: <http://pandia.ru/text/78/164/51745.php>

4. Эффективный сайт. Веб-шрифты [электронный ресурс] // Новый репортер: [сайт]. [2015]. URL: <http://newreporter.org/2013/02/18/effektivnyj-sajt-veb-shrifty/>

5. Флеш-анимация как элемент дизайна сайта [электронный ресурс] // 4Style: [сайт]. [2012]. URL: <http://www.4style.ru/stil-dizain/89-flash>

УДК 130.2

Японская мультипликация. Особенности японской мультипликации Сидорчук В.И., Харитонов В.В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного дизайна и моделирования
E-mail: vladimir.sidorchuk.96@mail.ru

Аннотация

Сидорчук В.И., Харитонов В.В. Японская мультипликация. Особенности японской мультипликации. Рассмотрено определение термина «аниме», связанное с японской анимацией; изучены особенности аниме; сделаны выводы

Ключевые слова: аниме, анимация, медианпространство, мультипликация, дизайн, рисовка.

Актуальность. Начиная со второй половины 1980-х годов, японская поп-культура стала быстро распространяться по всему миру, её любители появились в восточной Азии, США и европейских странах. В наше время поп-культура уже не отождествляется с низкопробными формами развлечений или досугом рабочего класса, как это представлялось до 1960-х гг. Скорее она передаёт то, как люди воспринимают жизнь. Так, согласно Джону Триту, массовая культура служит «условным обозначением для несчётного числа способов, делающих современных людей “современными” или даже просто “людьми”». Японская поп-культура весьма разнообразна, но, пожалуй, аниме и манга стали её главными представителями. [2]

Постановка проблемы. Вслед за ростом популярности аниме за границами Японии, также растет и количество противников этого жанра мультипликации. Большинству зрителей не нравится графическая составляющая в аниме - "большие глаза", непривычная рисовка и спецэффекты. Основная часть

критики, высказываемой в адрес аниме, можно отнести к любой ветви современной визуальной культуры.

В современном медиапространстве часто используется термин «аниме». Далеко не всегда авторы употребляют данное понятие в соответствии с его исходным значением. В данной статье мы постараемся дать развернутое определение «аниме», а также обозначить наиболее характерные особенности художественного стиля.

Основная часть. Аниме – это японская мультипликация. Главным отличием аниме от мультипликации других стран является то, что большая часть продукции рассчитана на разные возрастные категории (дети, подростки, взрослые). Зачастую аниме отличается характерной манерой отрисовки персонажей и фонов.

Термином “аниме” определяют только базовую культуру, на основе которой создаются соответствующие произведения, и более ничего. Почему же вообще имеет смысл использовать отдельные термины, почему нельзя просто говорить “японская анимация”? Зачем отдельные термины?

Японская культура анимации, безусловно, самая развитая и разнообразная в мире, при этом она во многом отличается от других национальных культур анимации и комиксов.

Поднимая вопрос, о существовании значительной национальной комикс-анимационной культуры можно говорить только в приложении к США, Японии и Франции (ФРГ и обе Кореи хотя, но на данный момент не столь опытные, Российские аниматоры также стремятся, но до такого профессионализма далеко). При этом США и Франция—“европейские” страны с христианской ментальностью, а Япония—“восточная” страна с буддийско-синтоистской ментальностью.

Поскольку, с точки зрения менталитета Россия все-таки ближе к Франции и США, японская массовая культура, частью которой являются аниме и манга, для нас не очевидна и не всегда понятна “с первого взгляда”. Отсюда и необходимость отдельных терминов для японской анимации и комиксов—даже если они иногда кажутся похожими на американские или французские, глубинные культурные различия между ними все-таки весьма значительны.



Рис. 1 – пример аниме

Отрисовка. Первым, что бросается незнакомым с аниме людям, это “огромные” глаза у персонажей. Стоит подчеркнуть, что именно к их детализации уделяют больше всего внимания, по отношению ко всем остальным деталям лица. Нос и рот зачастую изображают несколькими волнистыми линиями. Исключениями могут послужить моменты, когда персонаж говорит. Однако существуют произведения, в которых нос, рот и скулы наравне с прочими деталями лица рисуются и затеняются с большей точностью. Пример можно увидеть на рисунке 2.

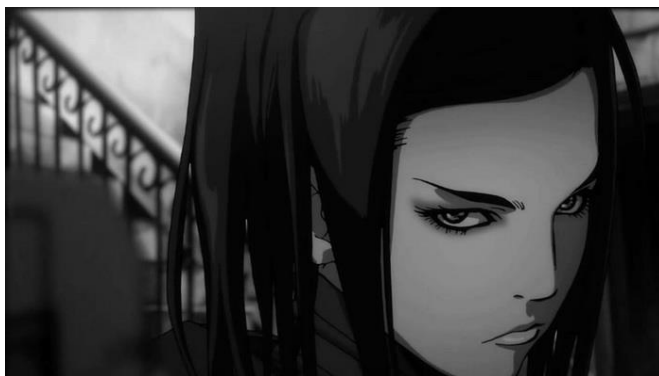


Рис. 2 – рисовка из аниме: “Ergo Proxy”

Волосы в аниме обычно состоят из прядей. Волосы, а также детали одежды персонажей, зачастую подчиняют ветру или инерции, заставляя их несинхронно двигаться при перемещениях персонажа. Волосы различных цветов изначально были способом «персонифицировать» героев, сделать их заметно различными. Сегодня, когда образы персонажей прорабатываются вплоть до мельчайших деталей лица и манер поведения, разноцветные волосы не являются необходимостью, скорее—традицией. Кроме того, цвет волос часто отражает характер персонажа. К примеру, рыжие волосы являются характерным индикатором вспыльчивости. Светлые волосы также указывают на иностранное происхождение героя, поскольку большинство японцев имеют тёмные волосы.[1]



Рис. 3 – примеры разной рисовки

Также следует заметить, что даже в одном произведении может использоваться несколько стилей рисовки. Этим самым, автор задает контраст мультипликации. Пример можно увидеть на рисунке 4.

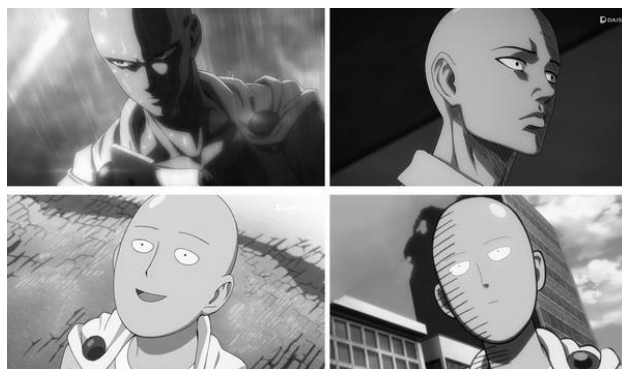


Рис. 4 – разная рисовка в одном аниме

Передача эмоций в аниме передается особенно. Помимо традиционных изменений выражений лица или тона голоса, используется ряд других приёмов. Эмоции могут изображаться подчеркнуто нереалистично, гипертрофированно – персонажи говорят с закрытыми глазами, чтобы передать безапелляционность, или принимают картинно демонический вид, когда проявляют гнев (Рис. 5а)[2]. В комедийных ситуациях, с целью показать несерьёзность чувства, применяются пиктограммы, такие, как картинка «капельки пота» и «вздвухся вен», возникающие поверх головы героя (либо в рамочке над ней), или же меняют цвет и формы деталей лица (Рис. 5б).



Рис. 5 – Передача
эмоций: злость;



Рис.6 – Передача
эмоций: смущение;

Выводы. В результате исследования выявлено, что в отличие от других видов анимации, графика японской мультипликации находится под сильным влиянием канонов и шаблонов, а потому обычно играет подчиненную роль по отношению к другим аспектам анимационного произведения. Анимация Японии также отличается анимации других стран тем, что представляет собою развивающийся обособленный культурный слой, который включает в себя множество идейных символов, графических особенностей, форм и типажей.

Список литературы

1. Особенности аниме [электронный ресурс] // AC MAGAZINE – журнал для молодых и активных: [сайт] [2016]. URL: <http://www.animacity.ru/node/218>
2. Mikhailova Y.D., Torchinov E.A. Images at an Impasse: Anime and Manga in Contemporary Russia // Japan and Russia. Three Centuries of Mutual Images. – Folkestone: Global Oriental, 2008. P. 175-191

УДК 004.92

Применение инфографики к отображению аналитической информации энерго- и ресурсопотребления

Юлина Ю.А., Харитонов А.Ю.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
E-mail: julia-julina@yandex.ru

Аннотация

Юлина Ю.А., Харитонов А.Ю. Применение инфографики к отображению аналитической информации энерго- и ресурсопотребления. Выполнен анализ инфографики к отображению аналитической информации энерго- и ресурсопотребления. Определены рекомендации по проектированию графического интерфейса пользователя на основе инфографики.

Ключевые слова: информация, визуализация данных, инфографика.

Актуальность исследования. Познавая окружающий мир, человек постоянно имеет дело с информацией. Она помогает человеку правильно оценить происходящие события, принять обдуманное решение, найти наиболее удачный вариант своих действий. Информация стала одним из важнейших стратегических, управленческих ресурсов, наряду с ресурсами - человеческим, финансовым, материальным. Ее производство и потребление составляют необходимую основу эффективного функционирования и развития различных сфер общественной жизни. Данные могут выглядеть по-разному – как огромные таблицы с числами, как большие географические карты, как схемы, тексты, иллюстрации.

Более наглядной и эффективной формой визуализации данных является графическая, а, например, при моделировании и обработке изображений – единственно возможная. Многие программы для финансовых, научных, технических расчётов используют эти и некоторые другие способы визуализации данных. Визуальное представление информации является прекрасным инструментом при проведении научных исследований, наглядным и веским аргументом в научных статьях и дискуссиях [1].

Цель работы — провести анализ инфографики к отображению аналитической информации энерго- и ресурсопотребления, рассмотреть принципы визуализации данных графического интерфейса пользователя с учетом правил инфографики.

Постановка задачи исследования. Сформулировать рекомендации по созданию инфографики на основе аналитической информации энерго- и ресурсопотребления, проектирования графического интерфейса пользователя на основе инфографики.

Решение задач и результаты исследований. Визуализация информации – это представление числовой и текстовой информации в виде графиков, диаграмм, структурных схем, таблиц, карт и т. д.

Инфографика – способ передачи информации с помощью графики и текста. Инфографика может использоваться в любой научной или профессиональной сфере, где есть необходимость схематизации материала: в ряде естественных и гуманитарных наук, в рекламе, маркетинге, журналистике, статистике и во многих других областях знаний.

Инфографика работает там, где нужно показать устройство и алгоритм работы чего-либо, соотношение предметов и фактов во времени и пространстве, продемонстрировать тенденцию, показать, как что-то выглядит, организовать большие объёмы информации. Используется там, где сложную информацию нужно представить быстро и чётко.

Современные методы визуализации информации широко используются для представления и анализа результатов компьютерного моделирования. Визуализация является одним из наиболее перспективных направлений повышения эффективности методов анализа и представления информации.

Визуализация аналитических данных – это наглядное представление больших массивов числовой и другой информации для оценки ситуации и принятия решений. Преобразование данных в интуитивно понятные и научно обоснованные графические представления, простые для понимания и использования.

Цели визуализации – это реализация основной идеи информации, это то, ради чего нужно показать выбранные данные, какого эффекта нужно добиться – выявления отношений в информации, показа распределения данных, композиции или сравнения данных.

Отношения в данных – это то, как они зависят друг от друга, связь между ними. С помощью отношений можно выявить наличие или отсутствие зависимостей между переменными.

Распределение данных – то, как они располагаются относительно чего-либо, сколько объектов попадает в определенные последовательные области числовых значений.

Композиция данных – объединение данных с целью анализа общей картины в целом, сравнения компонентов, составляющих процент от некоего целого.

Сравнение данных – объединение данных, с целью сравнения некоторых показателей, выявление того, как объекты соотносятся друг с другом, сравнение компонентов, изменяющихся с течением времени.

При визуализации аналитической информации энерго- и ресурсопотребления школами г.Донецка были использованы такие области инфографики как визуализация результатов сравнения и распределения данных.

Для инфографики информационной аналитики наиболее важна визуализация результатов сравнения данных, так как аналитик находится в переплетении сравнений данных разного рода.

Можно выделить пять видов сравнения, какими может быть выражен любой аспект данных:

- временное сравнение – показывает развитие объекта в некоторых хронологических рамках;
- покомпонентное сравнение – позволяет выделить каждый компонент в процентах от общего целого;
- позиционное сравнение – помогает изобразить соотношение данных между собой;
- корреляционное сравнение – показывает зависимость данных между собой;
- частотное сравнение – отображение попадания объектов в определенные интервалы.

Инфографика может быть представлена в виде иллюстраций, графиков, диаграмм, структурных схем, таблиц, карт. При создании инфографики аналитической информации наиболее часто используются диаграммы и графики. В зависимости от цели и данных можно выбрать выбирают график.

Общенаучные формы свертывания данных широко применяются в аналитике, например, диаграммы сравнения, которыми могут выступать столбовые, круговые, лепестковые диаграммы, точечные графики. Линейные диаграммы, графики с областями и гистограммы могут содержать в одном аргументе для одной категории несколько значений, которые могут быть как абсолютными, так и относительными.

Для системы анализа энергопотребления было решено использовать точечные и линейные графики, круговые диаграммы. Круговая диаграмма отображает процент, занимаемый каждым значением внутри набора данных, в виде разбитого на части круга. Точечный график рассеивания показывает распределение ограниченного набора точек. Например, связь между потреблением энергетического ресурса и показателями, влияющими на его потребление.

Неформализованным и привлекательным способом передачи информации являются наглядные изображения результатов анализа. Аналитики все чаще прибегают к инфографике, чтобы наглядно подтвердить выявленные тенденции, полученные результаты и упростить процесс восприятия сложной аналитической информации. Успех визуализации напрямую зависит от правильности ее применения, а именно от выбора типа графика, его верного использования и оформления.

В процессе создания интерфейса на основе инфографики следует придерживаться следующих этапов:

- выбор темы – должна быть актуальна и востребована;
- сбор информации;
- систематизация и структуризация – организация ранее собранной информации, создание плана ее презентации;
- создание эскиза и раскраска – создание макета на основе которого будет построен интерфейс;
- создание дизайна – выбор шрифта, стиля, цветовой палитры;
- вычитка и корректура – окончательная проверка дизайна и контента.

Информационный дизайн часто является связующим материалом, который скрепляет компоненты дизайна. Однако во всех случаях информационный дизайн сводится к принятию решений о том, как представить информацию, чтобы людям было легче воспринимать и использовать ее. [2]

Макет страницы – это место встречи информационного дизайна, дизайна интерфейса и дизайна навигации, которые совместно формируют единую конструкцию, связывающую части в целое. Для описания макета страницы составляется подробный документ, называемый схемой страниц или прототипом страниц. Прототипы страниц необходимы прежде всего в работе над визуальным дизайном сайта, где они служат первым шагом процесса формального определения дизайна. Прототип интерфейса для web-ориентированной системы анализа потребления энергии и ресурсов школами г. Донецка представлен на рис.1.

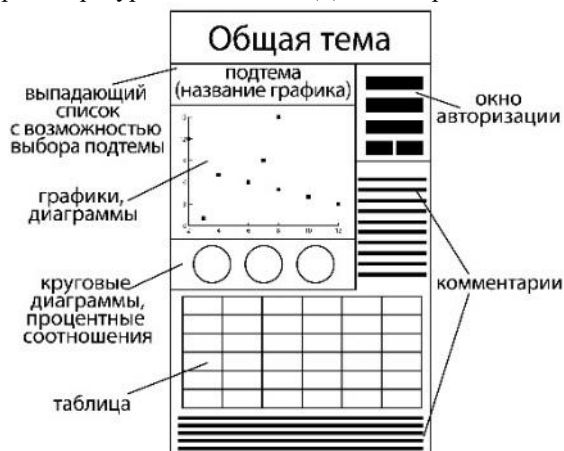


Рис.1 – Схема интерфейса страницы

Для того, чтобы разработать дизайн будущего приложения, подойдет любой графический редактор. В своей работе буду использовать такие редакторы как Adobe Photoshop и Adobe Illustrator. В процессе разработки макета

определяется вариант размещения основных блоков на странице, выбор шрифтов и цветовой схемы.

Цвет – это очень важная часть любого дизайна, особенно когда речь идет о создании необходимого контраста между фоном и текстом, фоном и визуальными данными. Цвета позволяют подчеркнуть идею, заложенную в контенте, и способствуют лучшему восприятию графической информации. Поэтому выбирать цвета необходимо исходя из темы своей работы, для того чтобы инфографику сделать более информативной.

Выводы. Произведен анализ инфографики к отображению аналитической информации энерго- и ресурсопотребления, рассмотрены принципы визуализации данных графического интерфейса пользователя с учетом правил инфографики. Сформулированы рекомендации по проектированию графического интерфейса пользователя на основе инфографики. Это позволит успешно разрабатывать более удобные интерфейсы на основе инфографики, обеспечит успешность и удобство в использовании проектируемого приложения. С учетом данных рекомендаций в дальнейшем будет разработано web-ориентированный программный продукт в рамках дипломного проекта.

Список литературы

1. Визуализация данных в науке и технике [электронный ресурс] // ПиКО - Кафедра Прикладной и Компьютерной Оптики [сайт]. URL:http://aco.ifmo.ru/el_books/computer_visualization/lectures/1.html
2. Дж.Гарретт. Веб-дизайн: книга Джесса Гарретта.Элементы опыта взаимодействия, 2008 – 186с.
3. Р.Крам. Инфографика. Визуальное представление данных, 2015 – 382с.
4. 8 принципов создания отличной инфографики [электронный ресурс] // Infogra [сайт]. URL:<http://infogra.ru/infographics/8-printsipov-sozdaniya-otlichnoj-infografiki>
5. Занимательная инфографика [электронный ресурс] // CMSmagazine аналитический портал рынка веб-разработок [сайт]. URL:http://www.cmsmagazine.ru/library/items/graphical_design/interesting-infographics

УДК 004.92

Анализ приложений для визуализации математических выражений на веб-ресурсах

Еремеев В.В., Павлий В.А.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна

E-mail: flexer_17@mail.ru

Аннотация

Еремеев В.В., Павлий В.А. Анализ приложений для визуализации математических выражений на веб-ресурсах. Выполнен анализ наиболее популярных плагинов и библиотек для визуализации математических выражений в браузере. Определены требования для разработки наилучшей альтернативы существующим программным продуктам.

Ключевые слова: браузерное приложение, плагин, математические выражения, визуализация.

Постановка проблемы. В процессе разработки ресурсов, где необходимо использовать сложные математические выражения как иллюстрации, дизайнеры часто сталкиваются с проблемами визуализации этих выражений. Существует множество программных продуктов для визуализации сложных математических выражений, формул и конструкций, написанных на разных языках и имеющих определенные особенности, преимущества и недостатки. Таким образом возникает проблема выбора конкретного плагина, максимально удовлетворяющего требованиям. Для реализации этого необходимо выполнить следующее:

- проанализировать популярные средства визуализации и построения математических выражений, выделить области их применения, достоинства и недостатки;
- определить характеристики, помогающие выбрать наиболее подходящий плагин, исходя из последующего использования разрабатываемого ресурса;
- представить концепцию для разработки наиболее универсального в использовании программного продукта, выполняющего поставленную задачу.

Цель статьи – провести анализ распространенных плагинов для построения математических выражений на веб-ресурсе, определить перечень характеристик, а также разработать концепцию для создания наиболее универсального плагина для выполнения данной задачи.

Анализ наиболее популярных плагинов. Для визуализации математических выражений на веб-ресурсах существуют различные программные продукты, имеющие свои особенности, преимущества и недостатки, определяющие их требования, методы использования, способы записи формулы и вывода на экран самого математического выражения. Следует рассмотреть наиболее популярные из них:

1. Javascript-библиотека jsTeXrender. Использует лицензию GPL v3 (в том числе, серверной части). Отображение формулы на странице осуществляется подключением скрипта. Все формулы в формате LaTeX пишутся внутри тегов `<pre lang="eq.latex"/>` и `<code lang="eq.latex">`. К достоинствам данной библиотеки относятся: легкость использования. Формулы остаются читаемы даже при выключенном скрипте. При добавлении новых формул ничего настраивать не надо. Среди недостатков можно выделить следующее: скрипт подгружается с внешнего ресурса и требует настроенный сервер.

2. Javascript-библиотека jsMath. В отличие от описанного в предыдущем пункте jsTeXrender, jsMath не требует настроенного сервера для работы, а рисует формулы сама. Формулы, по умолчанию необходимо писать в формате LaTeX. jsMath находит их сама и перерисовывает как надо. При желании, маркеры начала и конца формулы можно настроить. Рисует формулы jsMath хорошо. Достаточно сказать, что именно jsMath используется в стандартном интерфейсе блокнота Sage. К достоинствам данной библиотеки относятся: простота использования и установки, только Javascript, без серверной части. Среди недостатков можно выделить следующее: для качественного отображения предлагает пользователю установку специальных шрифтов.

3. Сервис Texify.com. Пользователь вводит свою формулу и получает код для вставки на страницу. Выданный на texify.com адрес картинки с некоторых пор не работает за пределами сайта texify.com, так что картинку придётся сохранить, куда-то выложить и только после этого вставить, куда надо. Всё это лишает Texify.com смысла. Аналогичный сервис доступен на mathURL.com. Как и tinyurl.com, он генерирует короткие адреса для картинок с формулами. Плюсы: минимум усилий для вставки одной-двух формул. Не зависит от скрипта (можно использовать, например, в ЖЖ или на форумах). Минусы: зависимость от внешнего веб-сервиса, использование изображений для визуализации.

4. Недавно возможность отрисовывать формулу LaTeX в виде картинки появилась в Google Charts. Ссылка на изображение формируется следующим образом: <http://chart.apis.google.com/chart?cht=tx&chl=формула LaTeX>. К сожалению, Гугл перестал поддерживать этот сервис, и его дальнейшая судьба неизвестна, но пока он продолжает нормально работать и им можно пользоваться.

5. MathJax. Если на сайте требуется вывести множество разных формул и математических символов, то имеет смысл подключить локальную библиотеку MathJax. Эта библиотека работает во всех браузерах, включая старые версии IE, а

также на iPhone, iPad и Android, поддерживает нотацию MathML, TEX и AsciiMath. Для использования MathJax необходимо скачать библиотеку и все требуемые для её работы файлы и скопировать их к себе на сервер. Можно поступить проще и загружать скрипт по сети. Главный JS-файл занимает около 50 Кб, но в процессе работы подгружает разные шрифты и другие скрипты. В итоге получается около пары сотен килобайт. Вроде много, но надо учитывать, что большинство файлов кэшируется при первой загрузке и в следующий раз файлы уже не загружаются.

6.MathML. Это язык разметки на основе XML для представления математических символов и формул в документах WWW. MathML рекомендован математической группой W3C. MathML рассматривает не только представление, но и смысл элементов формулы. Также разрабатывается система разметки математической семантики, призванная дополнить MathML. XML-структура MathML обеспечивает широкую область использования и позволяет быстро отображать формулы в приложениях, таких как браузеры, а также легко интерпретировать их значения в математических программных продуктах. Следует с осторожностью относиться к информации о том, что программный продукт поддерживает MathML, поскольку как правило не сообщается о версии поддерживаемого MathML. На практике же способ довольно громоздкий, неудобный, избыточный и не приспособлен для редактирования вручную.

7.CDF Player это программа для просмотра файлов формата вычисляемых документов разработанная компанией Wolfram Research. CDF Player целиком содержит библиотеку времени исполнения системы Mathematica, что делает возможной динамическую интерактивность файлов в формате вычисляемый документ. В частности содержимое документа может генерироваться в ответ на действие пользователя с помощью любых алгоритмов или визуализационных функций, которые можно описать в системе Mathematica. Это также позволяет CDF Player'у читать блокноты системы Mathematica. CDF Player имеет плагины для браузеров Internet Explorer, Mozilla Firefox, Google Chrome, Opera и Safari, таким образом можно просматривать CDF-файлы на веб-страницах. Клиентские плагины довольно ресурсоемкие, сам же движок требует установки на сервере. К тому же, компания-разработчик недавно сделала движок платным, что сильно ограничивает его применение на некоммерческих веб-ресурсах.

8.jsLaTeX — jQuery-плагин для использования LaTeX-формул на веб-странице. Формулы будут создаваться на специализированном отдаленном сервере и подставляться на странице с помощью jQuery. Следует отметить, что формулы в данной реализации записываются в формате LaTeX. Скрипт берет содержимое указанного контейнера (предполагаем, что текст в середине корректно написан) и выводит соответствующую картинку-формулу. Можно выбирать формат сохраняемого изображения.

9.PhpMathPublisher - плагин, написанный на PHP для визуализации формул путем формирования PHP-скриптом PNG-изображения. Для работы скриптов нужен только PHP с библиотекой GD. Изображения имеют прозрачный фон, что позволяет использовать их на веб-ресурсе с любым оформлением. Довольно прост в использовании, однако требует настроенный сервер.

Характеристики плагинов и их сравнение. После рассмотрения и анализа каждого из вышеперечисленных аналогов был проведен общий сравнительный анализ, позволивший определить единые ключевые параметры и требования ко всем программным продуктам. Результат анализа отображен в таблице 1.

Таблица 1 - Сравнительный анализ особенностей и требований различных плагинов

Плагин	Язык	Требует настроенный сервер	Необходимость использовать специальные шрифты	Формат вывода	Зависимость от внешнего интернет-ресурса
<u>jsTeXrender</u>	javascript	+	+	Изображение	-
<u>jsMath</u>	javascript	-	+	Символы	-
Texify.com	C	-	-	Изображение	+
Google Charts	javascript	-	-	Изображение	+
MathJax	javascript	-	-	Символы	-
CDF Player	Mathematica/C	+	-	Символы	-
jsLaTeX	jQuery	+	-	Изображение	-
PhpMathPublisher	PHP	+	-	Изображение	-
MathML	XML	+	-	Символы	-

На основе данных таблицы 1 можно сделать вывод, что наиболее популярным языком для написания плагинов визуализации математических выражений является javascript. Сервисы Texify.com и Google Charts зависят от внешнего ресурса и требуют интернет-подключение, что делает их малоприспособными для использования на веб-ресурсе, если учитывается возможность автономного просмотра страниц. jsTeXrender, CDF Player, jsLaTeX, PhpMathPublisher и MathML не являются полностью клиентскими приложениями и требуют настроенный сервер. Следует также отметить, что jsTeXrender и jsMath требуют специальные TeX-шрифты для своей работы.

Концепция для создания наиболее универсального плагина для визуализации математических выражений в браузере. Сравнив существующие аналоги и проанализировав их достоинства и недостатки были разработаны следующие требования для создания наиболее универсального плагина для визуализации математических выражений:

1. Язык написания - javascript. Он является наименее ресурсоемким и наиболее универсальным для браузеров, включая их старые версии. Нетипизированность языка облегчает разработчику написание парсера выражений.

2. Независимость от внешних ресурсов. Плагин не должен подгружать скрипты и организовывать обмен данными с интернет-ресурсами для обеспечения работоспособности в автономном режиме.

3. Независимость от настроенного сервера. Плагин должен быть полностью клиентским.

4. Независимость от специальных шрифтов. Разрабатываемый продукт не должен обязывать пользователя устанавливать какие-либо шрифты для корректной работы.

5. Вывод результата с помощью символов. Посимвольный вывод значительно быстрее формирования изображения, так как требует меньше вычислительных ресурсов.

Выводы. Произведен анализ существующих средств визуализации математических выражений и формул в браузерах. Определены их преимущества и недостатки, а также общие для всех плагинов требования. Проведена оценка удобства использования и универсальность существующих решений. Результаты показали, что существует актуальная проблема создания универсального front-end приложения для поставленной задачи. Были выделены требования к такому приложению.

Список литературы

1. Javascript [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2012]. URL: www.wikipedia.org/wiki/JavaScript
2. Математика в картинках [электронный ресурс] // Влад Мержевич: [сайт]. [2013]. URL: <http://htmlbook.ru/blog/matematika-v-kartinkah>

УДК 004

Разработка структуры для web-ресурса «Футбольный портал Донецкой Народной Республики»

Гаранжа А.В., Губенко Н.Е.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна

E-mail: garazh94@gmail.com

Аннотация

Гаранжа А.В., Губенко Н.Е. Разработка структуры для web-ресурса «Футбольный портал Донецкой Народной Республики». Выполнен анализ и сравнение существующих спортивных порталов. Определены популярные разделы сайтов, разработана оптимальная структура для создания ресурса.

Ключевые слова: интернет, портал, структура web-сайта, спорт, футбол.

Постановка проблемы. С развитием сферы информационных технологий все большее количество пользователей активно использует всемирную сеть для получения информации, знаний и коммуникации с другими пользователями в целом и на определенные темы.

Активная пропаганда и современная мода на здоровый образ жизни постепенно наращивает численность населения в Донецкой Народной Республике, интересующегося спортом, в частности футболом. В связи с этим возникает необходимость создания площадки для общения, дискуссий и упрощенного поиска необходимой информации локального масштаба.

Для реализации этого необходимо выполнить следующее:

- проанализировать распространенные спортивные web-ресурсы, выделить их достоинства и недостатки;
- определить популярные разделы сайтов, изучить отзывы посетителей;
- представить созданную с учетом полученных сведений структуру портала.

Цель статьи – проектирование интернет-ресурса о футболе в Донецкой Народной Республике для систематизации информации и объединения болельщиков.

Анализ спортивных порталов. Веб-портал – сайт в компьютерной сети, который предоставляет пользователю различные интернет-сервисы, работающие в рамках этого сайта [1].

Футбол (от англ. foot – ступня, ball – мяч) – командный вид спорта, в котором целью является забить мяч в ворота соперника ногами или другими

частями тела (кроме рук) большее количество раз, чем команда соперника. В настоящее время самый популярный и массовый вид спорта в мире [2].

Существующие в мире и соседних странах спортивные порталы пользуются популярностью. По информации статистического сайта hotlog.ru в первую тысячу самых посещаемых русскоязычных сайтов входят несколько спортивных ресурсов, многие из которых по числу посетителей обходят такие сервисы, как rutube.com, сайты телеканалов «Россия» и «НТВ», торрент-трекер «Торрентино», сайт прогноза погоды «Синоптик» и другие [3].

Место в рейтинге	Название ресурса	Количество посетителей
49	Чемпионат: ведущий сайт о спорте	683 621
64	Всероссийский портал «Спортбокс»	527 548
83	Спортивный портал «СПОРТ-ЭКСПРЕСС»	383 184
189	Футбол на Куличках	155 574
234	Евроспорт Россия	119 223

Рис. 1 – ТОП-5 русскоязычных спортивных интернет-ресурсов

Из данной статистики вытекает актуальность разработки футбольного портала Донецкой Народной Республики.

Определение популярных разделов. Анализ пяти лучших сайтов со спортивной тематикой позволяет извлечь информацию о популярных и необходимых пользователю возможностях для внедрения их в проект собственного портала. Сравнительная характеристика приведена в рисунке 2.

Возможности	Название ресурса				
	Чемпионат	Спорт-бокс	Спорт-экспресс	Футбол на куличках	Евроспорт
Охват нескольких видов спорта	+	+	+	-	+
Интерактивное меню	+	+	+	-	-
Новости	+	+	+	+	+
Мультимедиа	+	+	+	+	+
Форум	-	+	-	+	-
Таблицы	+	+	+	+	+
Зарубежные турниры	+	+	+	+	+
Результаты матчей	+	+	+	+	+
Команды	+	+	+	+	+
Опросы	+	+	+	+	+
Обратная связь	+	+	+	+	+

Рис. 2 – Сравнительная характеристика популярных спортивных ресурсов

Исходя из анализа существующих сайтов и порталов, а также учитывая предложения пользователей, была создана структура «Футбольного портала Донецкой Народной Республики». Целесообразным является внедрение всех 11 пунктов таблицы и дополнительно необходимо добавление таких разделов как: тесты (на знание Правил игры в футбол, истории футбола, команд), документы (протоколы матчей, регламент соревнований, Правила игры в футбол) и вопросы действующим лицам (тренерам, футболистам, арбитрам, футбольному союзу). Структура портала проиллюстрирована на рисунке 3.

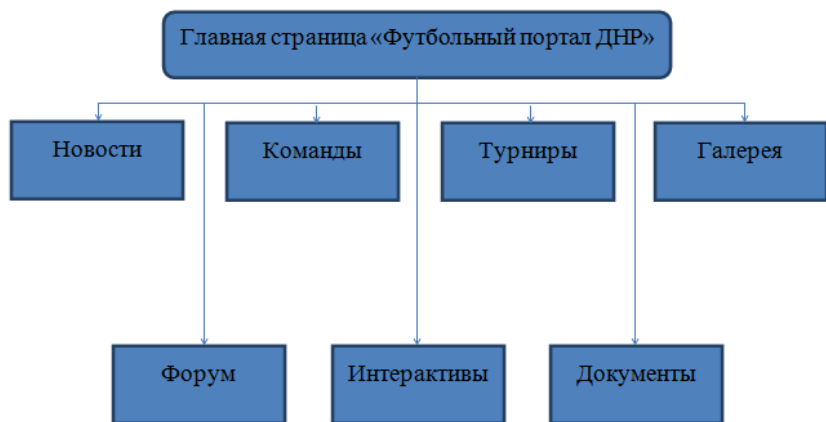


Рис. 3 – Структура web-ресурса «Футбольный портал Донецкой Народной Республики»

Максимальной популярностью пользуются разделы: новости, форум, турниры и интерактивы. Большинство сайтов не имеют взаимодействия с пользователем в интерактивном виде, внедрение которых и станет главной особенностью портала. Посетителям ресурса будет предложено поучаствовать в опросах касательно регламента турниров, политики развития футбола в Республике и множестве других тем; принять участие в голосованиях за лучшего игрока, тренера, исход матча или турнира; пройти тест на знание правил и истории футбола, любимой команды.

Игнорируя относительно большой спрос на различные виды спорта, на портале представлены только футбол и мини-футбол, которые являются родственными, что позволяет сделать ресурс узкоспециализированным. Это является одновременно достоинством и недостатком сайта.

Каждый раздел портала содержит подразделы, которые открываются в интерактивном меню и попасть в них можно из любой страницы.

1. «Новости»: Футбол; Мини-футбол; Мировые новости.
2. «Команды»: Высшая лига; Первая лига; Мини-футбольные команды.
3. «Турниры»: Высшая лига; Первая лига; Кубок; Юноши.
4. «Галерея»: Фото; Видео.
5. «Форум»: Обсуждения; Связь с администрацией; Вопросы действующим лицам.
6. «Интерактивы»: Опросы; Голосования; Тесты.
7. «Документы»: Правила игры; Регламент; Другие.

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 [4].

Выводы. Развитие футбола в Донецкой Народной Республике порождает необходимость заполнения пробелов в информировании населения о состоявшихся и грядущих мероприятиях, создания площадки для коммуникации и контакта с действующими лицами. Разработка web-ресурса «Футбольный портал Донецкой Народной Республики» позволит распространять информацию для заинтересованных лиц об актуальных футбольных событиях в регионе, собрать воедино и структурировать данные о командах и турнирах, обсуждать новости и делиться своим мнением среди болельщиков, а также пропагандировать футбол и здоровый образ жизни среди населения.

Список литературы

1. Веб-портал [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2016]. URL: www.ru.wikipedia.org/wiki/Веб-портал
2. Футбол [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2016]. URL: www.ru.wikipedia.org/wiki/Футбол
3. Хотлог.ру. Рейтинг ТОП-1000: Все русскоязычные сайты [электронный ресурс] // Хотлог.ру: [сайт]. [2016]. URL: www.top1000-ru.hotlog.ru
4. Д. Тидвелл. // Разработка пользовательских интерфейсов – СПб: Питер, 2008 г.
5. А. Уолтер. // Эмоциональный веб-дизайн – М.: Манн, Иванов и Фербер, 2012 г.

УДК 007.92

**Разработка информационной системы для верификации и публикации
студенческих персональных сайтов**

Овсеян Т.Н., Павлий В.А.

Донецкий Национальный Технический Университет, г. Донецк
кафедра компьютерного моделирования и дизайна

E-mail: tigran.raider@mail.ru

Аннотация

Овсеян Т.Н., Павлий В.А. Разработка информационной системы для верификации и публикации студенческих персональных сайтов. Разработана система контроля ошибок при разработке студенческих сайтов. Дополнительно создан интерфейс в виде портала, который выполняет все функции системы.

***Ключевые слова:** информационная система, верификация, интерфейс, студенческие персональные сайты, портал.*

Постановка задачи. В данный момент достаточно актуальной является задача агрегирования различных сайтов в рамках одной платформы.

Цель статьи – описать критерии и типичные ошибки при разработке информационной системы, а также провести анализ ошибок, которые могут допустить студенты и предупредить их при помощи разрабатываемой системы.

Существует большое количество платформ для агрегации, наиболее известной из которых является система uCoz [1]. Данная система позволяет в достаточно короткий срок создать собственный сайт любой сложности – от персональной страницы до интернет-магазина [2]. Для этого в системе имеется большое количество готовых шаблонов, что дает возможность создавать сайты даже не подготовленному пользователю. Внешний вид главной страницы данной системы показан на рисунке 1.

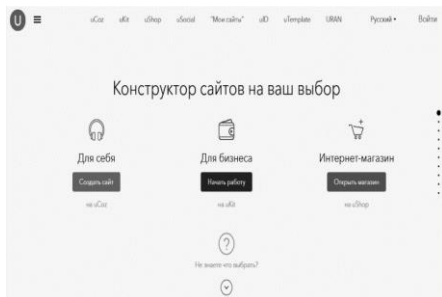


Рис.1 – Главная страница системы uCoz

При всех своих преимуществах система uCoz, являясь универсальной платформой, имеет собственный конструктор сайтов, что не позволяет решать специфические задачи, связанные, например, с верификацией создаваемых ресурсов. Также система не дает возможности разрабатывать серверные сценарии. Фактически её возможности ограничены встроенным конструктором сайтов.

Другим примером подобной системы, рассчитанной в первую очередь на агрегацию персональных сайтов магистров, является портал магистров ДонНТУ [3], разработка которого ведется с 90-х годов. Система имеет удобный интерфейс для навигации по магистерским ресурсам, фотогалерею, также поддерживается несколько языков. Внешний вид главной страницы портала магистров показан на рисунке 2.



Рис. 2 – Главная страница портала магистров ДонНТУ

Большая часть магистерских сайтов, которые поддерживаются данной системой, на этапе разработки могут иметь множество однотипных ошибок. К таким ошибкам относятся: превышение размера магистерского сайта, наличие недопустимых файлов (например, thumbs.db), наличие кириллицы в названиях файлов и т.п. Подобные ошибки можно было бы контролировать автоматически, однако в данной системе подобная возможность отсутствует.

В данной статье предполагается разработать информационную систему для агрегации студенческих персональных сайтов – во многом похожую на портал магистров ДонНТУ, однако, в отличие от него, разрабатываемая информационная система будет контролировать типичные ошибки разработки студенческих сайтов автоматически.

Наиболее типичные ошибки разработки студенческих сайтов и разработка критериев верификации. При разработке системы требовалось учесть удобство её использования преподавателями и студентами. Кроме того,

система в автоматическом режиме должна контролировать максимальное количество возможных типичных ошибок разработки. Принцип действия системы следующий.

Студент выкладывает папку со своим персональным сайтом с использованием File Transfer Protocol (FTP), с выданным преподавателем логином и паролем. Этот логин и пароль дает доступ к временной папке на сервере, которая называется dev. Все остальные каталоги сервера защищены от аписи. Поэтому после подключения необходимо перейти в указанный каталог и скопировать в него папку с сайтом. В каталоге dev допускается также создавать подкаталоги для группировки студентов по группам. В соответствии со стандартами разработки статических сайтов, внутри обязательно должны находиться файл index.html, другие html-файлы, файлы с css-стилями и дополнительные ресурсы в виде картинок, шрифтов и т.п. Кроме того, для обобщения сведений обо всех студентах, в каждой папке студенческого сайта должен находиться текстовый файл meta.txt. Из него и будет формироваться ссылка на сайт студента (за основу берется фамилия, имя и отчество, указанные в файле meta.txt). Этот файл содержит основную информацию о студенте. Кодировка файла – UTF-8(без BOM). Файл состоит из семи текстовых строк. Перевод строки после ввода последней строки не используется. Файл имеет следующую структуру, описанную ниже.

Первая, вторая, третья строки файла содержат фамилию, имя и отчество студента. Здесь допускается использовать русские, прописные и заглавные символы, символ дефиса и пробел (для студентов, имеющих двойную фамилию или имя).

Четвертая строка содержит дату рождения студента, записанную в формате xxxx-xx-xx. Первые 4 цифры – это год, затем следует месяц и число. Если число или месяц состоят из одной цифры, то они должны быть дополнены ведущим нулем. Например, если студент родился 3 апреля 1998 года, то дата должна записываться в следующем виде: 1998-04-03.

Пятая строка файла содержит относительный путь к фотографии студента (относительно главного каталога сайта). Эта фотография в дальнейшем будет использоваться для автоматического формирования галереи студентов. В пути и имени файла фотографии допускается использовать малые латинские буквы, символ точки и символы прямого слеша для указания поддиректории. Сама фотография должна иметь размеры 150 пикселей в ширину и 200 пикселей в высоту. Формат файла изображения – png или jpg. Путь к фотографии не должен начинаться с прямого слеша. Не допускается использование относительных путей для ссылки на родительские директории.

Шестая строка файла содержит адрес электронной почты студента, записанной в общепринятом формате. Пример корректно оформленного файла meta.txt:

Иванов
 Иван
 Иванович
 1998-01-04
 images/photo.jpg
 ivanov@mail.ru

Задача системы заключается в:

- проверке корректности содержимого данного файла;
- проверке корректности имен всех файлов и имен всех вложенных каталогов. Во избежание проблем, связанных с переносом файлов сайта под ОС Ubuntu (на которой работает сервер), имена файлов и каталогов должны использовать только малые латинские буквы и цифры, а также символ нижнего подчеркивания;
 - проверке наличия фотографии студента и корректности её размеров;
 - проверке наличия файла index.html в главном каталоге персонального сайта. Имена файлов для остальных страниц выбираются студентом самостоятельно;
 - проверке отсутствия скрытых и системных файлов, файла thumbs.db и файла .htaccess. Файлы thumbs.db создаются автоматически при разработке и очень часто копируются на сервер, что недопустимо. Файл .htaccess – настроечный файл web-сервера Apache, наличие которого может повлиять на работу сервера;
 - проверке объема сайта, который в общем случае не должен превышать 1.5 мегабайта;
 - проверке имени корневого каталога персонального сайта, которое должно представлять транслитерацию фамилии студента и должно быть записано только малыми символами, включая первый символ. Если в группе присутствуют однофамильцы, то допускается добавление к имени каталога цифры или первой буквы имени.

Таким образом, автоматический контроль указанных ошибок позволяет автоматически верифицировать студенческие персональные сайты и предупреждать наличие посторонних файлов, которые могут повлиять на работоспособность сервера или самого сайта студента.

Разработка интерфейса информационной системы. Для обеспечения удобства был разработан портал, который выполняет все функции системы и обладает удобной навигацией. Внешний вид главной страницы портала показан на рисунке 3.

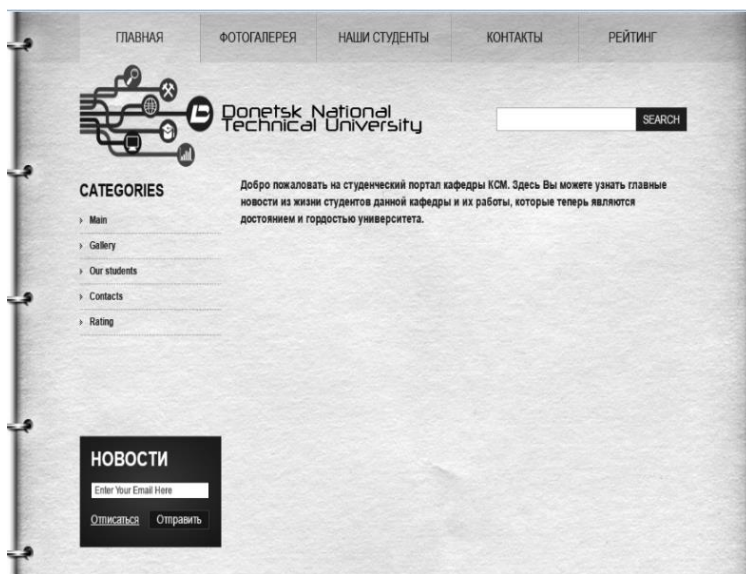


Рис.3 – Главная страница портала

К основным разделам разработанного портала относятся:

- «Главная»: страница с приветственным словом и кратким обзором целей создания сайта. Также находятся такие вкладки как новости и меню на английском языке слева;

- «Фотогалерея»: содержит в себе галерею фотографий студентов на основании тех путей, которые были указаны в meta.txt;

- «Наши студенты»: основная страница, в которой используется описанный выше механизм верификации студенческих сайтов. Также на данной странице имеются подразделы в соответствии с делением студентов на группы. Рядом с фамилией студента может отображаться специальный символ (звездочка), который говорит о том, что студенческий сайт находится в стадии разработки. При нажатии на данный символ будет отображаться перечень контролируемых системой ошибок во всплывающем окне. Пользователь сможет переходить на студенческие сайты посредством щелчка по фамилии студента;

- «Контакты»: контактная информация. Содержит в себе такие пункты, как номера телефонов для связи, рабочий e-mail, физический адрес и API карты для удобного поиска местоположения;

- «Рейтинг»: оценка посещаемости каждой страницы студента, вывод ТОП-посещаемых;

Выводы. В рамках данной статьи разработана система для верификации и публикации студенческих персональных сайтов. Она проверяет и предупреждает наличие всех основных ошибок и недочетов, с которыми сталкиваются студенты при разработке своих сайтов. Также система предупреждает создание «мусора» в виде системных файлов и контролирует объем студенческого сайта. После завершения разработки, система позволяет опубликовать сайт путем его переноса на постоянное место хранения. Предполагается, что использование системы существенно повысит качество и скорость разработки студенческих персональных сайтов, разрабатываемых ими в рамках курса «Web-технологии и web-дизайн».

Список литературы

1. UCoz, конструктор сайтов, система управления сайтом [электронный ресурс] // UCoz copyright [2005-2016] URL: <http://ucoz.ru>
2. UCoz, система управления сайтом [электронный ресурс] // Википедия – свободная энциклопедия. [2014] URL: <https://ru.wikipedia.org/wiki/UCoz>
3. Портал магистров ДонНТУ [электронный ресурс] // Аноприенко А.Я. URL: <http://masters.donntu.org/>
4. Информационная система [электронный ресурс] // Википедия – свободная энциклопедия. [2016] URL: https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0

УДК 004.92

Разработка структуры информационного портала «Traditional art»

Головина Е.О., Губенко Н.Е.,

Донецкий национальный технический университет, г. Донецк

кафедра компьютерного моделирования и дизайна

E-mail: cherrybenjam@gmail.com

Аннотация

Головина Е.О., Губенко Н.Е. Разработка структуры информационного портала «Traditional art». Рассмотрены основные виды внутренней структуры сайтов. определены основные правила дизайна его интерфейса.

Ключевые слова: структура, дизайн интерфейса, информационный портал, веб-портал, навигация портала, интерактивность.

Постановка проблемы. Создание порталов представляет собой разработку ресурсов, предоставляющих для каждого пользователя огромные возможности. Для эффективного использования портала, необходимо разработать простой и удобный интерфейс для пользователя, а также структуру, которая будет полностью зависеть от тематики и целей. В последние годы, разработка дизайна для сайтов и порталов приобрела исключительный интерес у современной молодежи. Но не каждому удастся создать портал с грамотной организацией структуры портала и привлекательным дизайном. Качественный дизайн крайне важен, ведь именно интерфейс определяет удобство пользования вашим сайтом. Необходимо создать правильную структуру и качественный интерфейс для портала «Traditional art».


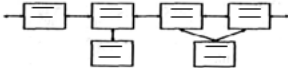
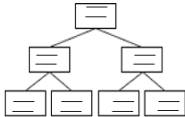
Цель – Разработать структуру и интерфейс портала «Traditional art». Проанализировать виды внутренней структуры сайта. И на основе популярных информационных порталов определить основные правила для разработки дизайна интерфейса портала.

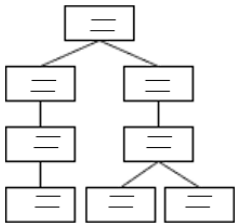
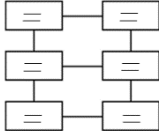
Анализ структуры портала. Порталом называется ресурс, содержащий большое количество информации самых различных направлений. Он совмещает в себе сразу несколько видов сайтов. Основной его целью является предоставление пользователям самой разной и актуальной информации. На страницах портала помимо информации находятся опросы, голосования, форумы, чаты, в общем всё, что необходимо для удобства пользователей и что в дальнейшем поможет развивать портал в правильном направлении. Создавать, поддерживать и обеспечивать работу такого вида сайтов сложно, но зато они могут привлекать практически все категории пользователей и каждому быть интересны.

Перед тем как создать свой портал нужно составить четкую структуру, представив его в виде книги у которой есть оглавление, которое и является навигацией по сайту. Названия основных разделов, расположенные на главной странице должны нести общую информацию о содержимом, а в подразделе – детализировать ее. Таким образом, пользователь, двигаясь по сайту, будет целенаправленно перемещаться от общего к частному.

Идеальная структура сайта имеет один вход и один выход. Пользователь, заходя на главную страницу вашего сайта, находит интересующую его информацию, под которой находится ссылка на страницу, где ее можно найти. Таким образом, пользователь продвигается от главной страницы к внутренним, что позволяет ему не пропустить ту информацию, которая будет ему интересна и полезна. Единственный выход с сайта у пользователя лежит через посещение как минимум двух страниц, что увеличивает значение счетчика ваших посещений. Данная структура сайта строится по принципу «воронки».

Таблица 1 – Виды внутренней структуры сайтов

Название	Вид	Описание
<p>Линейная структура</p>		<p>Если материал выстраивается в логическую цепочку, образуется линейная структура. Линейная цепочка обязана иметь начало и конец. Начинать просмотр сайта с середины обычно лишено смысла.</p>
<p>Линейная структура с ответвлениями</p>		<p>Пользователь последовательно переходит с одной страницы на другую. Если информация, размещенная на какой-то из них, его заинтересовала, и он хочет узнать подробности, то может перейти на ответвление, а потом вернуться обратно на основную «дорожку». Главным преимуществом рассматриваемой структуры является то, что к ней легко перейти с обычного линейного размещения web-страниц.</p>
<p>Древовидная структура</p>		<p>Содержимое каждой страницы кроме первой входит в виде подраздела в страницу более высокого уровня.</p>

<p>Гибридная структура</p>		<p>На практике используются и древовидная и последовательная структуры. Например, оглавление каталога статей начинается с иерархически расположенных страниц, но каждую отдельную главу или статью зачастую имеет смысл разбить на несколько следующих друг за другом кусочков. Обратная конфигурация, когда ветки древовидной структуры вырастают из узлов линейной структуры, встречается реже.</p>
<p>Решетчатая структура</p>		<p>Основана на построении системы навигации сайта, когда между вертикальными и горизонтальными страницами имеется взаимная связь и возможность быстрого перехода с одной страницы на другую без необходимости посещения промежуточных страниц. Подобная структура приводит к излишнему увеличению гиперссылок и применение её ограничено для сайтов с большим количеством страниц.</p>

Выбор основных правил для разработки дизайна. При разработке дизайна интерфейса информационного портала необходимо учитывать такие особенности как структура и удобство навигации портала, стиль текста, интерактивность и авторские права.

Необходимо использовать специальную разметку портала. Создание такой разметки позволяет выделить главные элементы и такой подход даёт возможность понять, что и как будет смотреться на различных устройствах. Этим стоит пользоваться для выделения необходимых элементов.

Очень важно, чтобы весь дизайн портала был доступным и простым. Все популярные порталы, например такие как: Yahoo.com, Bing.com, Google.com используют простой и удобный для пользователей дизайн.

Для разработки дизайна интерфейса информационного портала, необходимо учитывать такие основные правила:

1. Структура и удобство навигации портала.

Пользователь должен легко понимать структуру сайта и свободно ориентироваться в нем, а инструменты навигации сайта должны быть понятны. В хорошем портале не должно быть «пустых» страниц с сообщениями о ведущихся

на них работах по наполнению. Каждая страница портала должна быть оформлена в едином стиле. При подробном анализе портала Yahoo.com можно заметить, что навигация рассчитана на новых пользователей. Для любого, кто зайдет на новый незнакомый сайт, будет понятно, как его использовать.

2. Стиль текста.

Для того чтобы дизайн интерфейса был ясен, необходимо подобрать подходящий шрифт. Также нельзя использовать несколько шрифтов, иначе они будут конфликтовать друг с другом. Следует использовать «безопасные» шрифты, чтобы они поддерживались на всех устройствах. В каждом веб-портале используется не более трех видов шрифтов.

3. Интерактивность.

Предусмотрен форум для пользователей, установлены социальные кнопки, благодаря которым пользователь может поделиться записью в социальной сети. На многих известных информационных порталах используют социальные кнопки.

4. Авторские права.

Портал, использующий чужие материалы, дизайн и информацию, не вызывает одобрения. Поэтому, всегда необходимо согласовывать с авторами желание использовать их материалы. Также, необходимо всегда указывать источник взятой информации, автора, а также должна быть предоставлена ссылка на его ресурс. В веб-порталах, под любым материалом, дана ссылка на источник, либо указан автор.

5. Не использовать флеш-заставки и анимацию в дизайне.

Подобные флеш-заставки затрудняют путь пользователя к содержимому сайта. Дополнительная анимация в дизайне портала, может отвлекать пользователя от необходимой ему информации.

6. Использование оптимального количества цветов.

При использовании большого количества цветов, дизайн портала будет выглядеть грубо.

7. Использовать подбор цветовой гаммы.

Необходимо использовать «безопасные» цвета, которые будут выглядеть на всех устройствах одинаково. При создании изображения для публикации в сети главной проблемой является правильная передача цвета на разных типах мониторов и в разных браузерах. Когда браузер не в состоянии правильным образом передать тот или иной цвет, он подбирает похожий или смешивает несколько соседних цветов. В результате пользователь на своем экране может увидеть совсем не то, что хотел изобразить автор.

При этом в наше время, каждому веб-разработчику, веб-дизайнеру нужно также учитывать то, как портал будет выглядеть на планшетах и смартфонах. Крайне важно, своевременно внести изменения в структуру и дизайн портала.

Выводы. Проанализированы виды внутренней структуры сайта. И на основе популярных информационных порталов определена верная структура и основные правила для разработки дизайна интерфейса портала «Traditional art».

Список литературы

1. <http://techno.bigmir.net/technology/1531714-Kruche-nekuda--TOP-20-samyh-populjarnyh-sajtov-interneta>
2. http://www.i2r.ru/static/255/out_22129.shtml
3. <http://www.6floor.ru/clienthelp3.htm>
4. <http://www.wsformula.ru/content/%D1%83%D0%B4%D0%B0%D1%87%D0%BD%D1%8B%D0%B9-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD>
5. <http://www.designonstop.com/webdesign/tips/9-osnovnyx-pravil-veb-dizajna- kotorye-dizajner-mozhet-inogda-narushat.htm>

УДК 004.42

Создание сайта-портфолио Ерёменко П.С., демонстрирующего возможности работы в React.js, Relay, GraphQL, Babylon.js, Node.js

Ерёменко П.С., Хоруженко А.С.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
pashtet5246@gmail.com, and_riy@ukr.net

Аннотация

Ерёменко П.С., Хоруженко А.С. Создание сайта-портфолио Ерёменко П.С., демонстрирующего возможности работы в React.js, Relay, GraphQL, Babylon.js, Node.js. Проведен анализ современных методов и способов создания сайтов на языке Javascript. Рассмотрены средства настройки связи типа “клиент-сервер”, составлен список.

Ключевые слова: javascript, фреймворки, NodeJS, ReactJS.

Постановка проблемы. JS-framework'и — это инструменты для построения динамических веб/мобильных/настольных приложений на языке Javascript. Как и к любым другим инструментам, разработчики прибегают к использованию js-фреймворков там, где невозможно/очень сложно/очень долго выполнять задачу обычными средствами. В подавляющем большинстве случаев фреймворки используются для написания так называемых Single Page Applications.

С их помощью можно разрабатывать как полноценные сайты, так и функциональные модули (различные онлайн-инструменты). Конечно, полноценные фреймворки лучше подходят для первой задачи, а для второй рекомендуется использовать более легковесные фреймворки или библиотеки.

Фреймворки предоставляют четкую структуру приложения и реализуются с использованием так называемых «паттернов проектирования». Наиболее широко распространены следующие паттерны: MVC (Model-View-Controller), MVP (Model-View-Presenter) и MVVM (Model-View-ViewModel).

Преимущества построения приложения на JS-фреймворке:

- можно легко реализовать SPA (Single Page Application);
- использование js-фреймворка обязывает иметь структуру приложения («скажем решительное «нет» спагетти-коду!»);
- кода становится заметно меньше и он чище, что позитивно отражается на скорости разработки, поддержке и устранении ошибок в коде приложения;
- наличие структуры подразумевает модульность приложения, а это дает возможность проще работать над приложением несколькими разработчикам одновременно;
- и последнее преимущество больше вытекает из использования самого javascript, но при использовании фреймворка значительно усиливается: возможность быстро создать мобильное и/или настольное кроссплатформенное приложение из веб-версии с помощью систем типа PhoneGap или Apache Cordova.

Цель статьи – демонстрация эффективности использования современных технологий создания web-сайтов.

Изложение материалов исследования. Для достижения поставленной цели было решено выбрать и проверить стек технологий на примере реального сайта-портфолио с таким функционалом: просмотр видео, обратная связь, возможность оставлять комментарии и смотреть примеры уже созданных сайтов.

В мире разработки одной из главных проблем является сложность поддержки продукта. Кодовая база в проектах постоянно растёт. Иногда для того, чтобы изменить функционал кода, его нужно поменять в нескольких местах, так как в современных web-приложениях необходимо делать сложные интерфейсы, где на экране одновременно присутствуют более одного потока действий. Подход ООП, или более общий «разделяй на компоненты и властвуй», в жизни работает

только на банальных сайтах. Компоненты, попав в сложный контекст, начинают зависеть не только от своего состояния, но и от других, выходящих за границы локальной модели, факторов: истории взаимодействия, окружающей обстановки, контекстной рекламы, настроения пользователя и т.п.

При разработке сайта-портфолио задачей было избегать или решать данные проблемы, чтобы постараться по максимуму показать преимущества и обойти возможные недостатки выбранных фреймворков.

Для первоначальной постановки цели исследования было необходимо определить используемый стек технологий. Проведён отбор среди набирающих популярность библиотек фреймворков:

- AngularJS;
- Ember.js;
- Backbone;
- Spine.js;
- KnockoutJS;
- Batman.js;
- Babylon.JS;
- Three.js;
- PlayCanvas.js;
- Relay + GraphQL.

Изучив описания каждого из них, было принято решение использовать Relay + GraphQL (спецификация вышла 11 августа 2015 года [1]), React (Facebook представил его как открытое ПО в марте 2015 года), Babylon.js (размещён на GitHub 23 июня 2013, набрал популярность в октябре 2015 года [6]) потому, что данный подход является новым в мире Web-разработки, а также в силу нарастающей популярности данных технологий как в клиентско-серверной части, так и в 3d моделировании (Babylon.js).

Остальные фреймворки также подходят для эффективной разработки, но данным технологиям уже более 3х лет, их уже нельзя считать современными, поэтому такое изучение не соответствует цели исследования.

Поэтому в браузерной части были использованы библиотека для построения пользовательских интерфейсов React[2] и JS-фреймворк Relay [3]), для управления передачей данных в React приложения. Для демонстрации 3D возможностей использован фреймворк Babylon.js[5].

На сервере установлены NodeJS, NoSQL, GraphQL. Relay работает только в связке с GraphQL[4], клиент под который есть только на NodeJS (разработал Райан Дал (англ. Ryan Dahl) в 2009 году после двух лет экспериментирования над созданием серверных веб-компонентов). В качестве базы данных выбран NoSQL. Несмотря на отсутствие собственного опыта работы с ней, судя по отзывам других разработчиков, она хорошо подходит для наших целей.

В проекте применён такой подход, как изоморфный JS, что позволяет использовать один и тот же код и на стороне клиента, и на стороне сервера, что существенно сокращает время его написания.

Выводы. В процессе создания Web-сайта были решены проблемы необходимости одновременного изменения кода в нескольких местах, но обнаружен ряд других проблем, таких как высокий порог вхождения в данные технологии, отсутствие полноценной оригинальной документации и какого-либо перевода на русский язык, что в принципе, вполне ожидаемо для технологий, выведенных на рынок менее 3х лет назад.

В дальнейших планах развития сайта-портфолио – собрать полноценный проект для мобильных телефонов, используя PhoneGap или его аналоги.

Список литературы

1. Facebook. A data query language and runtime. [Электронный ресурс]. – Режим доступа: <http://graphql.org/> .
2. Facebook. A JavaScript framework for building data-driven React applications. [Электронный ресурс]. – Режим доступа: <https://facebook.github.io/relay/> .
3. Facebook. A JavaScript library for building user interfaces. [Электронный ресурс]. – Режим доступа: <http://facebook.github.io/react/> .
4. Facebook. Draft RFC Specification for GraphQL. [Электронный ресурс]. – Режим доступа: <http://facebook.github.io/graphql/> .
5. An open source 3D engine based on WebGL and Javascript.[Электронный ресурс]. – Режим доступа: <https://doc.babylonjs.com/>
6. BabylonJS/Babylon.js [Электронный ресурс]. – Режим доступа: [https://github.com/BabylonJS/ Babylon.js/graphs/contributors](https://github.com/BabylonJS/Babylon.js/graphs/contributors)

УДК 004.056

Методы защиты веб-сайта журнала «САИТ» от атак через интернет.

Бойко В.В., Губенко Н.Е.

Донецкий национальный технический университет, г. Донецк

кафедра компьютерных систем мониторинга

E-mail: LInkor.95@mail.ru

Аннотация

Бойко В.В., Губенко Н.Е. Методы защиты веб-сайта журнала «САИТ» от атак через интернет. В данной статье рассматривается проблема защиты веб-сайта научного журнала. Проанализированы основные типы атак через сеть интернет и некоторые методы борьбы с ними. На основе проведенного анали выделены основные методы защиты веб-сайта журнала от глобальных и локальных атак.

Ключевые слова: веб-сайт, локальная атака, глобальная атака, сервер, антивирус, аутентификация, cookie.

Введение. По данным антивирусной сетки KSN за 2014 год было осуществлено 6 167 233 068 вредносных атак на компьютеры и мобильные устройства. Для проведения атак через интернет злоумышленники воспользовались 9 766 119 уникальными хостами. Следует учитывать, что эти данные справедливы только для пользователей KNS. На самом же деле количество атак приблизительно в 6 раза больше.

В настоящее время количество атак, нацеленных на веб-серверы возрастает по закону Мура, причём их географическое расположение никак не влияет на безопасность и связано с их открытостью к пользователям. Зачастую серверы рассчитаны на обмен информации с пользователями. Злоумышленник может вмешаться в работу такой системы изменив начальную функциональность, однако существуют методы, которые позволяют уменьшить шанс быть атакованным.

Постановка проблемы. Разрабатываемый веб-ресурс предназначен для публикации научных работ, защиты авторских прав писателей и распространения научных работ в сети Интернет. Таким образом, возникает необходимость исследования основных типов атак через интернет и анализа некоторых методов для защиты от этих типов атак.

Цель статьи – анализ и выбор эффективных методов защиты для разработки веб-сайта научно-технического журнала «Системный анализ и информационные технологии в науках про природу и общество» (САИТ) является актуальной.

Изложение материалов исследования. Атаки на веб-серверы можно условно разделить на две категории:

- локальные атаки, которые обычно направлены на кражу информации или перехват управления на отдельном веб-сервере:
- глобальные атаки, которые обычно направлены на несколько веб-сайтов и ставят своей целью заражение всех их посетителей:

С помощью локальных атак злоумышленник может получить доступ к базе данных публикаций журнала, и нарушить их достоверность. С помощью глобальных атак можно заразить публикации вредоносным ПО и заблокировать доступ к ресурсу. Поэтому для создания и использования безопасного веб-сайта журнала следует максимально защитить сервер на котором он размещается и данные пользователей.

Так как некоторые области веб-сайта должны быть доступны только зарегистрированным пользователям, необходим метод проверки подлинности пользователей.

Существует несколько методов аутентификации пользователей:

- Базовая аутентификация, при использовании которой имя пользователя и пароль включаются в состав веб-запроса. Даже если контент с ограниченным доступом не слишком важен, этот метод лучше не использовать, так как пользователь может применять один и тот же пароль на нескольких веб-сайтах.

- Дайджест-аутентификация, поддерживаемая всеми популярными серверами и браузерами, позволяет надежно шифровать имя пользователя и пароль в запросе. Она помогает обеспечить безопасность имен и паролей, что производит соответствующее впечатление на пользователей и снижает вероятность успешной атаки на сервер.

- Протокол HTTPS позволяет шифровать все данные, передаваемые между браузером и сервером, а не только имена пользователей и пароли. Протокол HTTPS (основанный на системе безопасности SSL) следует использовать в случае, если пользователи должны вводить важные личные данные.

При выборе системы аутентификации рекомендуется использовать самый безопасный вариант из имеющихся в наличии.

Планирование является неотъемлемой частью системы безопасности, поскольку оно позволяет сократить ущерб от вирусов, шпионского ПО и других вредоносных программ. Большинство атак можно предотвратить, построив правильную структуру веб-ресурса, которая позволит выстроить более надёжную защиту и выявить успешную атаку, если такая будет. К сожалению, на данный момент нет гарантии, что атака на сайт не удастся, как бы ни была выстроена защита.

В 2007 году сайт MySpace.com подвергся атаке с помощью трояна JS/SpaceStalk-A, который крал информацию, хранящуюся в файлах cookie, и передавал ее на удаленный сервер. Такая информация может содержать конфиденциальные сведения — имена пользователей, адреса предпочитаемых сайтов и пароли.

Одна из важных проблем при проектировании заключается в том, что запрос новой страницы всегда обрабатывается вне контекста предшествующих запросов. Таким образом «запоминание» пользователя сложнее, чем обычно.

Большинство браузеров поддерживают два метода, которые могут использоваться для «запоминания» посетителей: обычные файлы cookie и файлы cookie сеанса.

Файл cookie — это небольшой файл, создаваемый браузером и хранящийся на компьютере пользователя. Его содержимое не регламентируется, но обычно в таких файлах хранится название, дата окончания срока действия и некий объем данных, например: «Count = #» или «Member = #».

Файл cookie сеанса похож на обычный, но при этом он позволяет веб-приложениям хранить данные в памяти.

Обычный файл cookie сохраняется на устройстве пользователя и остается на нем до момента удаления пользователем. Файл cookie сеанса, хранится только на протяжении времени работы устройства и автоматически теряется при закрытии приложения-браузера. Однако, все файлы cookie подвержены манипуляциям извне.

Разработчики часто склонны считать данные из файлов cookie надежными, поскольку считают, что все должно быть в порядке, поскольку они сами разрабатывают соответствующий код. Это мнение ошибочно, так как злоумышленники могут изменить файл cookie (в редких случаях — и данные активного сеанса), чтобы обманом заставить веб-сайт предоставить доступ к закрытой странице.

При проектировании системы никогда нельзя полагаться на надежность пользовательских данных, вводимых посетителями напрямую или поступающих через файлы cookie. Оптимальное решение — считать все данные, хранящиеся на устройстве пользователя, ненадежными.

Веб-сервер журнала сформирован несколькими слоями ПО, каждый из которых подвержен разным видам атак. На рис. 1 представлена основная структура построения ресурса.

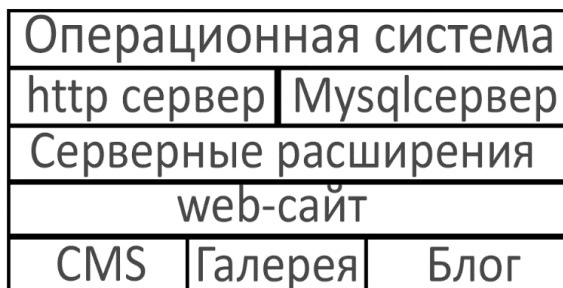


Рис.1 – Структура веб-ресурса журнала.

Основой любого сервера является операционная система. Обеспечить её безопасность можно, вовремя устанавливая последние обновления системы безопасности. Многие системы безопасности устанавливают обновления автоматически.

Злоумышленники склонны автоматизировать свои атаки, перебирая один сервер за другим в поисках сервера, где не было установлено обновление. В связи с этим важно следить за тем, чтобы обновления устанавливались своевременно и надлежащим образом; любой сервер, на котором установлены устаревшие версии обновлений, может подвергнуться атаке.

Следует обновлять систему безопасности на сервере, и всё ПО работающее на веб-сервере. Любое ПО, не относящееся к необходимым компонентам (например, DNS-сервер либо средства удаленного администрирования наподобие VNC или служб удаленных рабочих столов), следует отключить или удалить. Если средства удаленного администрирования все же необходимы, следить за тем, чтобы не использовались пароли по умолчанию или пароли, которые можно легко угадать. Это замечание относится не только к средствам удаленного администрирования, но и к учетным записям пользователей, коммутаторам и маршрутизаторам.

Антивирусное ПО является одним из важнейших критериев. Его использование является обязательным требованием для веб-сервера журнала. Антивирусное ПО становится одним из самых эффективных способов защиты от угроз безопасности. Когда веб-сервер становится целью атаки, злоумышленник без промедления пытается загрузить инструменты взлома или вредоносное ПО, чтобы успеть использовать уязвимость системы безопасности до того, как она будет закрыта. При отсутствии качественного антивирусного пакета уязвимость системы безопасности может долгое время оставаться незамеченной. Следующий по важности компонент программного обеспечения — сам HTTP-сервер, лучшими альтернативами здесь являются IIS и Apache.

Выводы. Для базовой защиты веб-ресурса от таких атак, как глобальные и локальные не следует устанавливать лишние компоненты, так как любой компонент несет с собой отдельную угрозу, и чем их больше, тем выше суммарный риск. Для успешной атаки достаточно единственной брешки в системе безопасности. Своевременные обновления системы безопасности для операционной системы и приложений поддерживают стабильную работу системы.

Список литературы

1. Статья "Обеспечение безопасности веб-сайтов" [Электронный ресурс]. URL: <https://yandex.ru/support/webmaster/protecting-sites/intro.xml>
2. Примеры атак Apache, [Электронный ресурс]. URL: http://www.ossec.net/wiki/index.php/Apache_attack_samples
3. Kaspersky Security Bulletin 2014. Основная статистика за 2014 год [Электронный ресурс]. URL: <https://securelist.ru/analysis/ksb/24580/kaspersky-security-bulletin-2014-osnovnaya-statistika-za-2014-god/>

УДК 004.43

Анализ систем управления содержимым Joomla!, Drupal, WordPress и выбор оптимальной CMS для создания интернет-портала.

Родненко Е.С., Губенко Н.Е.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна

E-mail: j.eun@mail.ru

Аннотация

Родненко Е.С., Губенко Н.Е. Анализ систем управления содержимым Joomla!, Drupal и WordPress и выбор оптимальной CMS для создания интернет-портала. Выполнен анализ бесплатных систем управления содержимым Joomla!, Drupal и WordPress. Определены правила выбора наилучшей платформы при разработке интернет-портала.

Ключевые слова: CMS, системы управления содержимым, joomla, drupal, wordpress, платформа, интернет-портал.

Постановка проблемы. В процессе работы с сайтами, разработчики часто сталкиваются с проблемами потери быстродействия. Таким образом, возникает необходимость определить правильный подход к разработке выбранного сайта и к платформе, обеспечивающей необходимую производительность. Для реализации этого необходимо выполнить следующее:

- проанализировать распространенные CMS, выделить области их применения, достоинства и недостатки;
- найти решение проблем быстрого действия сайтов;
- осуществить выбор платформы, основываясь на требованиях к разработке интернет-портала. Представить особенности структуры CMS Joomla.

Цель статьи – провести анализ распространенных CMS и наиболее точно выбрать нужную платформу для создания интернет-портала.

Система управления содержимым (контентом) (от англ. Content manager system, CMS) – информационная система или компьютерная программа, используемая для обеспечения и организации современного процесса создания, редактирования и управления контентом (т.е. содержимым) [1].

В настоящее время существует множество CMS, с помощью которых создание блогов (личных страниц), веб-порталов и интернет магазинов стало намного удобнее и не требует огромных ресурсов. При создании сайта на движке, важно выбрать оптимальную CMS-систему, при помощи которой можно будет реализовать поставленную задачу. Рассмотрим наиболее популярные и общедоступные системы управления содержимым, такие как: Joomla, Drupal и WordPress.

По рейтингу RUWARD и iTrack за июнь 2015 года, самой популярной системой является WordPress, доля которой составляет 31,64% в общем рейтинге. Популярность этой системы обусловлена тем, что она представляет собой мощную платформу для быстрого создания одностраничных сайтов, проста в использовании, в публичном доступе находятся сотни документаций по использованию на русском, поэтому не требуется быть продвинутым юзером, также в открытом доступе есть большое количество плагинов, шаблонов и цветовых тем. Данная CMS подходит для сайтов где необходимо быстро и легко добавлять информацию, к примеру таких как блоги, информационные ресурсы.

Недостатками данной системы являются:

- нагрузка на сервер;
- ограниченный набор основных функций;
- большое количество некачественных плагинов и тем;
- высокая вероятность взлома;
- отсутствие службы поддержки;
- автоматическая генерация копий страниц и изображений.

На втором месте по рейтингу CMS находится Joomla!, ее доля в общем рейтинге составляет 23,21%. Данная система взяла дружелюбность к пользователю у WordPress и гибкость у Drupal. Платформа с открытым кодом (Open Source CMS), предоставляет широкий функционал, который позволит реализовать сайт любой сложности, но для хорошего результата и освоения необходимо будет потратить некоторое время, однако при возникновении каких либо проблем, в открытом доступе есть масса документаций и порталов, в том

числе и на русском языке. Система постоянно обновляется и повышает качества безопасности, благодаря огромному сообществу. Разработана для создания веб-порталов, иными словами для сообществ, в которых главным являются социальные функции. Система поддерживает протоколы контроля доступа (OpenID, LDAP, Gmail.com), содержит административное управление, поддерживает базы данных, встроенная многоязычность, как и WordPress имеет много плагинов, модулей и шаблонов, однако для индивидуальной настройки требует оплаты.

Недостатки системы:

- при редактировании адреса материала создается копия материала только с новым URL;
- трудности при переходе на более новую версию;
- отсутствие службы поддержки;
- систему нельзя назвать простой в изучении;
- нагрузка на сервер;
- большая вероятность взлома, если своевременно не обновлять платформу.

Пятую позицию занимает Drupal. Данная система используется в 4,47% по общему рейтингу CMS за июнь 2015 года. Drupal – это мощная платформа разработанная на языке PHP, ориентированная на разработчиков и позволяющая создавать сложные сайты, требующая опыта и практики работы с ней. Преимущества Drupal: имеет hooks – php-функции, реализованные в модуле, исполняющие код в ответ на события или условия, произошедшие во время генерации страницы, для избегания почти любых проблем в системе, модули позволяют создавать произвольные типы данных и отображать их, с каждой версией становится проще в использовании, модуль Taxonomy, позволяет создавать множество уровней и типов различного контента, активное и большое комьюнити, повышенная функциональность. Система подходит для создания сайта любой сложности, начиная от блога и заканчивая интернет магазином, файлообменником и порталом.

Недостатками являются:

- сложность системы;
- отсутствие службы поддержки;
- сильная нагрузка на сервер.

Портал позволяет объединить большое количество людей, связанных общими предпочтениями и интересами с помощью одной площадки, а благодаря уникальным, многофункциональным и интересным сервисам расширяются рекламные возможности. Несмотря на множество преимуществ относительно других видов сайтов, интернет-портал выдвигает более жесткие требования к производительности и квалификации разработчика. Для решения данной проблемы важную роль играет грамотный выбор платформы для разработки.

Одной из основных особенностей является наличие плагинов, позволяющих эффективно и с минимальной потерей производительности реализовать задачи, поставленные при создании веб-портала.

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. Также порталы функционируют как точки доступа к информации в интернете или сайты, которые помогают пользователям в поиске нужной информации. Такие порталы представляют информацию из различных источников в единообразном виде. Иногда их называют навигационными сайтами. Как правило, порталы выполняют функции поиска, а также предоставляют доступ к различным интернет-сервисам, например электронной почте, ленте новостей и т. д. Концепция веб-порталов — предоставление максимального количества интернет-сервисов в одном месте для привлечения наибольшего числа пользователей [3].

Исходя из определения, при создании интернет-портала необходимо уделить особое внимание на эффективность взаимодействия модулей движка, обработку видео, фото, быстрый доступ к контенту.

Основными тремя проблемами быстрого действия сайта являются:

- 1) Функциональная часть написана на основе JavaScript и зачастую эти функции написаны неграмотно;
- 2) Перегруженность сайта изображениями большого разрешения;
- 3) Расстояние от пользователя до сервера.

Для решения этих проблем можно воспользоваться следующими вариантами:

1) Проблемы, связанные с функциональной частью, написанной на основе языка JavaScript, можно решить посредством асинхронного режима загрузки. Также не следует перегружать сайт инструментами аналитики и стоит избавиться от внешних обращений к скриптам;

2) Важную роль в решении проблемы оптимизации сайта занимают загружаемые изображения. Инструментов для оптимизации изображений, иными словами для сжатия и преобразования их в нужный формат и размер, существует огромное количество и важно использовать данные инструменты, чтобы не засорять и не перегружать сайт (табл.1);

Таблица 1 – Оптимизация изображений

Проблема	Способ решения	Результат
1. Соответствие экранного размера изображения реальному размеру.	Уменьшение изображения до вставки на сайт	Уменьшение размера изображения в 3 раза, что благоприятно влияет на быстродействие сайта.
2. Ссылки на увеличенные копии изображений.	Вставка на страницу уменьшенных копий изображений, которые раскрываются по клику.	Значительное улучшение загрузки страниц.
3. Формат файлов изображений.	Правильный выбор формата изображения, во избежание расточительства быстродействия.	В случае, если изображение не обладает прозрачным фоном, анимацией или другими специфическими чертами, его вес занимает значительно меньше места и загрузка сайта происходит значительно быстрее.
4. Сжатие изображений	Удаление лишней информации файлов изображений, которая важна в определенных обстоятельствах, но может быть удалена без потери качества при вставке на страницы сайта.	10-20% экономии трафика, в некоторых случаях бывает 5-кратная разница в физическом размере изображения.

3) С проблемой удаленности сервера от клиента частично может помочь CDN (content delivery network – сеть доставки содержания). Она распределит нагрузку на сервер по нескольким узлам. CDN — сеть, состоящая из кеширующих узлов, расположенных в разных географических точках. При обращении к серверу запрос перенаправляется на ближайший к пользователю узел. Таким образом, сокращается сетевой путь, а значит, увеличивается скорость загрузки сайта [7].

Важным аспектом в создании успешного веб-портала является выбор правильной CMS. Из рассмотренных ранее вариантов наиболее оптимальным является Joomla!, обладающая следующими функциональными особенностями, ниже представлены некоторые из них:

1) Наличие JLayout (макетов) - это отличный способ отделить некоторые из сложностей использования различных CSS фреймворков. Но нужно помнить,

что это касается только статичного контента, а не JavaScript взаимодействий, которые требуют знание:

- а) использования DOM-модели фреймверка
- б) доступности вспомогательных функций JavaScript

По сути, JLayout это HTML уровень абстракции (abstraction layer).

2) Использование PHP 5.4, которое позволяет писать код компактнее и с меньшим дублированием, используя трейты. Это и есть практическая причина, стоящая за этим предложением. Такой подход сокращает на 15% ключевые компоненты кода и уменьшает количество ошибок.

3) Использование MySQL позволяет нам: оптимизировать запросы, особенно при работе на больших сайтах; снизить порог вхождения и ошибки в деплойменте, что позволяет быстрее обнаружить ошибки.

4) Улучшенная интеграция Composer. Composer – менеджер зависимостей PHP. Он самостоятельно проверяет установлены ли нужные библиотеки, в случае их отсутствия загружает и устанавливает нужные пакеты.

Выводы. Проанализировав самые популярные системы управления контентом, была выбрана наиболее оптимальная для начинающего разработчика платформа Joomla!, которая идеально подходит для создания веб-порталов, обеспечивая необходимые возможности для оптимизации содержимого.

Список литературы

1. https://ru.wikipedia.org/wiki/Система_управления_содержимым
2. <http://lifehacker.ru/2014/06/16/kakuyu-cms-vybrat-dlya-sajta-sravnenie-wordpress-joomla-i-drupal/>
3. <https://ru.wikipedia.org/wiki/Веб-портал>
4. <https://habrahabr.ru/post/151879/>
5. <https://pingoblog.ru/76-plyusy-i-minusy-cms-joomla.html>
6. <http://joomla.ru/articles/cms-joomla/customize/2486-6-shagov-k-uskoreniyu-sajta>
7. <https://habrahabr.ru/post/145946/>
8. <http://blog.getgoodrank.ru/povyshaem-bystrodejstvie-sajta-chast-i-izobrazheniya/>

УДК 004.427; 004.738.5

**Создание сайта поклонников творчества донецких художников
Мотренко В.В.**

Донецкий национальный технический университет, г. Донецк
Кафедра компьютерного моделирования и дизайна
E-mail: vladislav.051094@gmail.com

Аннотация

Мотренко В.В. *Создание сайта поклонников творчества донецких художников. Рассмотрена задача создания сайта о творчестве донецких художников. В реализованном проекте использована система управления сайтом DataLife Engine 11.0. Предложен ряд функциональных возможностей сайта, учитывающих анализ похожих сайтов о творчестве художников. Созданный сайт дает возможность заинтересованным посетителям получить наиболее полную информацию о творчестве художников.*

Ключевые слова: сайт, средства реализации, анализ, ссылки, права доступа, художники, творчество.

Постановка задачи. В современном мире, когда доступ к интернету имеет каждый второй человек, художники, естественно, обязаны иметь «свой островок» в океане под названием всемирная сеть. Сайт о художниках важен для пропаганды их творчества и раскрутки популярности, - посетитель, получает доступ к их работам – картинам и возможно его заинтересует одна или несколько из них. Через сайт можно информировать посетителя также о новых художниках, об их картинах, их выставках, которые будут вскоре проведены. Веб-сайт – это часть сложного механизма, отвечающего за контакт художников со своими почитателями. Необходимость в нем важна, как точка притяжения публики, что вместе с другими методами популяризует новизну и особенности их творчества. На таком сайте следует размещать как важную, так и интересную информацию о художниках: галереи работ, их биографии, статьи и публикации специалистов о творчестве художников, афиши выставок. Важна также реализация на сайте возможности широкого общения и поклонников творчества художников, и их критиков.

Чтобы обеспечить эффективное выполнение этих задач по успешной реализации сайта необходимо решение ряда проблем связанных с:

- анализом современных систем управления сайтом и выбором из них оптимальной системы, для дальнейшего использования;
- ознакомлением с сайтами-аналогами, и на основе полученных результатов разработкой структуры текущего сайта;

- проектированием макета страниц сайта в соответствии с его структурой, функциями и с психологической особенностью восприятия информации;

- разработкой структуры групп пользователей, прав их доступа к разделам и к управлению содержимым сайта.

Цель статьи – на основе проведенного анализа средств реализации типовых структур сайтов, создать тематический, информационный сайт поклонников творчества донецких художников, с определенным функционалом. При этом критерием отбора художников служат:

- наличие собственных картин,
- размещение работ в публичных местах города,
- участие в выставках или проведение персональных выставок,
- выставление собственных картин на продажу,
- наличие фанатов творчества художника,

Анализ современных средств реализации

Большинство современных веб-сайтов реализовано с помощью CMS (Content Management System) – система управления содержимым (контентом). Они предназначены для обеспечения и организации совместного процесса создания, редактирования и управления сайтом.

Использование CMS дает целый ряд преимуществ:

- простота редактирования содержимого сайта с помощью современных визуальных редакторов, что дает возможность пользователям, не владеющим языками веб-программирования, оперативно и качественно редактировать и содержимое, и разделы сайта;

- широкие возможности для настройки и оптимизации сайта;
- возможность установки дополнительных модулей;
- управление содержимым пользователями с разным уровнем доступа;
- надежная защита;
- автоматическое обновление и следование веб-стандартам.

Из современных систем управления содержимым сайта можно выделить несколько распространенных: WordPress, Joomla, Drupal, DataLife Engine, 1С-Битрикс, MODx, Opencart. Большинство CMS чаще используются для управления новостными сайтами и блогами. Например, для блогов в основном используют WordPress. Систему Joomla целесообразно применять для управления сайтом с гибкой и богатой структурой, используя расширения функций системы. CMS Drupal используется для построения самых разнообразных сайтов: блогов, новостных, информационных и даже для создания форумов. Для создания интернет-магазина подойдут: 1С-Битрикс, MODx и Opencart. Система DLE (DataLife Engine) «заточена» под новостные, информационные сайты и блоги.

Информационный сайт поклонников творчества донецких художников должен иметь гибкую и разнообразную структуру содержимого. Соответственно,

для такого сайта нам подойдут несколько из выше перечисленных систем управления, но свой выбор мы остановим на CMS DataLife Engine, которая является довольно простой в установке и настройке, а также очень гибкой. Данная система управления многофункциональна и соответствует всем требованиям функционала сайта. Она имеет ряд мощных и разнообразных возможностей, благодаря которым можно организовать всю структуру и функционал сайта.

Выбор структуры на основе анализа сайтов-аналогов

Для анализа структуры разделов были выбраны 5 сайтов различных художников. Результаты анализа представлены в таблице 1.

Таблица 1. Структуры разделов сайтов-аналогов

Сайт	Описание	Структура разделов
dkurts.com	Художник Дмитрий Кустанович	Главная – Галерея – О художнике – Отзывы – События – Контакты
kuinji.ru	Художник Архип Иванович Куинджи	Биография – Воспоминания – Статьи – Книги – Галерея – О творчестве – Гостевая книга
kazimirmalevich.ru	Художник Малевич Казимир Северинович	Биография – Книги – О творчестве – Черный квадрат – Галерея – Гостевая
ilyarepin.ru	Картины Ильи Репина	Биография художника – Переписка – Воспоминания – Репин-писатель – О творчестве – Галерея – Гостевая
velaskes.ru	Испанский художник Диего Веласкес	Биография – Книги – О творчестве – Галерея – Гостевая

Так как наш сайт посвящен не одному, а нескольким художникам, то в качестве наиболее характерных разделов были выбраны и добавлены следующие: Главная страница, Список художников, Биографии, Галерея картин, Статьи и публикации, Раздел критики, События, Контакты.

К базовой структуре сайта и ее базовым элементам относят: шапку сайта, которая содержит заголовок, навигационное меню и логотип сайта; боковой блок с дополнительной навигацией, блоком под управление профилем и авторизации; блок контента, содержащий заголовок страницы и название раздела сайта; подвал сайта, в котором находится подпись (копирайт). Разработанный макет структуры страницы интернет сайта представлен на рисунке 1. Как правило, подпись располагают внизу, после содержимого страницы. Для представления ссылок на ресурсы зачастую используют баннеры. Их целесообразно группировать в отдельные блоки интернет страницы. Важной частью страницы является навигационное меню, и его расположение должно не только привлекать внимание посетителей, но быть и удобным. Поэтому целесообразно разместить два типа связанных навигационных меню, в верхней части сайта – в шапке, и на уровне содержимого, для удобного пользования.



Рис.1. Макет структуры сайта

Учитывая особенности физиологии и психологии восприятия человеком информации, просмотр содержимого страницы осуществляется в соответствии с правилами чтения (в большинстве стран мира – слева-направо, сверху-вниз). Следовательно, и располагать основные элементы страницы необходимо по аналогии, в порядке их важности.

Все пользователи сайта делятся на авторизованных (зарегистрированных) и гостей (незарегистрированных). Зарегистрироваться на сайте может любой пользователь, но для управления содержимым, и возможности редактирования его необходима организация прав доступа для отдельных зарегистрированных групп пользователей. Список групп пользователей: Гость, Пользователь, Художник, Союз художников, Критик, Модератор, Администратор. Права доступа групп пользователей приведены в таблицах 2 и 3.
 Таблица 2. Права доступа групп пользователей

Группа	Открытый раздел	Закрытый раздел	Добавление содержимого	Редактирование содержимого	Удаление содержимого	Возможность оставлять комментарии	Панель управления
Гость	+	-	-	-	-	-	-
Пользователь	+	+	+	-	-	+	-
Художник	+	+	+	+	+	+	-
Союз художников	+	+	+	+	+	+	-
Критик	+	+	-	-	-	+	-
Модератор	+	+	+	+	+	+	-
Администратор	+	+	+	+	+	+	+

Таблица 3. Права доступа групп пользователей

Группа	Добавление без проверки модератором	Редактирование своих материалов	Редактирование всех материалов	Удаление своих материалов	Удаление всех материалов	Критика
Гость	-	-	-	-	-	-
Пользователь	-	-	-	-	-	-
Художник	+	+	-	+	-	-
Союз художников	+	+	-	+	-	-
Критик	-	+	-	+	-	+
Модератор	+	+	+	+	+	+
Администратор	+	+	+	+	+	+

Реализация сайта. Для реализации сайта была выбрана система управления - DataLife Engine версии 11.0, которая, на данный момент, является самой последней и содержит новые доработки скрипта. Данная CMS поддерживает современные языки программирования и разметки – HTML, CSS, JavaScript, PHP, MySQL.

И редактирование, и настройка компонентов сайта осуществлено при помощи средств панели управления CMS. В ней происходят все настройки модулей сайта, загрузка дополнительных расширений и их настройка, выбор и корректирование шаблонов страницы, управление материалами страницы, а также настройки пользователей и групп, доступов для них и всевозможные другие манипуляции с системой.

При помощи панели управления добавлены необходимые материалы – новости, статьи, биографии, картины и прочее, добавлена группа баннерных ссылок на дополнительные ресурсы, настроен модуль авторизации пользователей, реализован модуль поиска по сайту, произведена настройка групп пользователей, их прав и прочее. В соответствии с проектом макета страницы реализован и настроен стандартный шаблон оформления. В процессе наполнения и редактирования сайта, системой управления заполняется база данных MySQL, которая хранит всех зарегистрированных пользователей и добавленные материалы. Дополнительным модулем сайта является галерея картин. Для реализации галереи картин использован модуль фотогалереи Марк 1.10, разработанный специально для системы управления DLE начиная с версии 10.4 и выше. Данный модуль полностью оптимизирован и имеет ряд преимуществ: быстрая установка фотоальбомов, универсальный шаблон модуля для любых дизайнов, минимальные требования к серверу, легкая настройка модуля.

Выводы. На основе анализа современных способов управления сайтом и структуры сайтов-аналогов, разработан проект сайта поклонников донецких художников, который реализован на базе системы управления содержимым – DataLife Engine 11.0.

На сайте реализована система поиска по содержимому, галерея картин, блок ссылок на дополнительные ресурсы, модуль авторизации пользователей, предусмотрена настройка групп пользователей и прав доступа, публикации новостей и прочих материалов сайта.

Основными принципами создания информационного сайта для “участия” в творчестве художников являются: системный подход, информационная база, максимальная типизация проектных решений, непрерывное развитие системы, многократное использование массивов информации.

Список литературы

1. Определение CMS. Материал из Wikipedia (Википедия) – свободная энциклопедия. Электронный ресурс. Режим доступа: https://ru.wikipedia.org/wiki/Система_управления_содержимым
2. Общий рейтинг CMS за 2015 год. Материал из Ruward:Track. Электронный ресурс. Режим доступа: <http://track.ruward.ru/cms>
3. Система управления сайтами – DataLife Engine. Материал из официального сайта DLE. Электронный ресурс. Режим доступа: <http://dle-news.ru>
4. Калощко В.В., Аноприенко А.Я. Проектирование и разработка интернет сайта подразделения ВУЗа с использованием средства системы управления содержимым на примере интернет сайта факультета КНТ // Информатика и компьютерные технологии / Сборник трудов IX международной научно-технической конференции 4-6 ноября 2013 г., Донецк, ДонНТУ. – 2013. С. 132-136.

УДК 004

**Анализ и выбор алгоритмов наложения ЦВЗ в мультимедийных файлах
Шевляков А.И., Губенко Н.Е.**

Кафедра «Компьютерное моделирование и дизайн»
Донецкий национальный технический университет
E-mail: name.ass@gmail.com

Аннотация

Шевляков А.И., Губенко Н.Е. Анализ и выбор алгоритмов наложения ЦВЗ в мультимедийных файлах. Рассматриваются вопросы защиты авторского права на мультимедийную информацию, анализируются методы встраивания цифровых водяных знаков для доказательства подлинности фотоизображений.

Актуальность проблемы. В настоящее время выросла потребность в приватизации данных и защите авторского права уделяется большое внимание, так как всевозможные мультимедийные данные могут быть использованы в коммерческих целях. Множество пользователей порталов и социальных сетей могут стать жертвами злоумышленников. Всё чаще появляется случаи незаконного использования чужих видео и фотоматериалов и их владельца приходится доказывать свое преимущественное право на них.

Цель исследования. Анализ и выбор методов для наложения цифровых водяных знаков в мультимедийных файлах форматов AVI и JPEG.

Нарушение авторских прав. Нарушение авторских прав подразумевает несанкционированное правообладателем распространение материала, защищённого авторским правом, такого как программное обеспечение, музыкальные композиции, фильмы, книги, компьютерные игры а в настоящее время и на фотографии. Обладание правами на интеллектуальную собственность защищено законами большинства стран.

Под нарушением авторских прав обычно понимаются следующие действия:

- создание копии продукта и её продажа;
- создание копии продукта и передача её кому-либо ещё;
- в некоторых случаях перепродажа легально приобретённой копии.

Распространённость нарушений авторских прав

Нарушение интересов обладателей авторских прав (которые как правило не являются непосредственными авторами произведений интеллектуальной собственности) широко распространено во многих странах, в том числе в России, Украине, Китае, Казахстане, Мексике и Индонезии.[2] Особенности защиты

мультимедийного продукта существенно зависят от типа формата в котором он публикуется.

Особенности формата AVI и JPEG. Кратко остановимся на особенностях форматов файлов JPEG и AVI.

JPEG (*Joint Photographic Experts Group*, по названию организации-разработчика) - один из популярных графических форматов, применяемый для хранения фотоизображений и подобных им изображений. Алгоритм JPEG позволяет сжимать изображение, как с потерями, так и без потерь (режим сжатия lossless JPEG). Поддерживаются изображения с линейным размером не более 65535×65535 пикселей.[1]

AVI-файлы — особый случай файлов RIFF. (сокращенно от Resource Interchange File Format). Этот формат, изначально предназначавшийся для обмена мультимедийными данными, был Microsoft совместно с IBM. Данный формат является наиболее распространенной формой представления видео на персональных компьютерах. В зависимости от формы представления видеоданных файлы AVI бывают различных стандартов. Основой RIFF-формата является chunk. Chunk - это порция данных с сигнатурой, по которой определяется тип данных.[3]

Выбор формата мультимедиа для защиты авторского права при помощи наложения ЦВЗ. Рассматривать методы наложения ЦВЗ я буду на примере формата JPEG. Итак, внедряемый в защищаемую фотографию ЦВЗ должен отвечать двум противоречащим друг другу критериям: робастности (устойчивости к различным внешним воздействиям) и скрытности (обеспечения наименьших искажений изображения по сравнению с оригиналом)[4]. Для проверки авторских прав на цифровое изображение осуществляют извлечение встроенной информации, которое при недостаточной робастности ЦВЗ к атакам может стать невозможным.

Из существующих форматов цифровых изображений для сохранения полноцветных фотографий хорошо подходят форматы JPEG и JPEG2000, поскольку сохраненные в этих форматах изображения имеют высокие показатели качества при сравнительно небольших размерах файла. Недостаток данных форматов заключается в том, что для уменьшения размера графических файлов применяется алгоритм их сжатия с потерями, а значит, часть информации при сохранении таких изображений будет необратимо утеряна, и, следовательно, существует вероятность искажения или потери встроенных ЦВЗ.

Для повышения робастности вложения встроенных ЦВЗ к сжатию или масштабированию графических файлов в стегоалгоритмах стараются применять те же преобразования, что и в алгоритмах сжатия этих файлов.[5] Для формата JPEG таким алгоритмом является дискретное косинусное преобразование, для JPEG2000 – вейвлет преобразование.

Графический формат JPEG2000 имеет ряд преимуществ по сравнению с форматом JPEG. Это лучшее качество изображения при равной степени сжатия,

оптимизация качества кодирования, а также возможность сжатия без потерь. Поэтому при выборе формата сохранения цифровой фотографии лучше отдать предпочтение более современному JPEG2000.

Алгоритм наложения ЦВЗ

При разработке нового метода необходимо учесть плюсы упомянутого выше существующего, т.е. встраиваемый ЦВЗ должен быть устойчив к зашумлению изображения, его фильтрации и сжатию с потерями. В то же время маркировка областей изображения должна производиться таким образом, чтобы при процедуре извлечения ЦВЗ для проверки подлинности изображения можно было понять, производилась ли его обрезка, а также были ли удалены или заменены его фрагменты.

Для достижения поставленной цели можно использовать следующий алгоритм.

1. Исходное изображение, которое можно представить как матрицу $f(n,m)$, разбивается на одинаковые блоки a_{ij} размером 64 x 64 пикселя (Рис. 1).

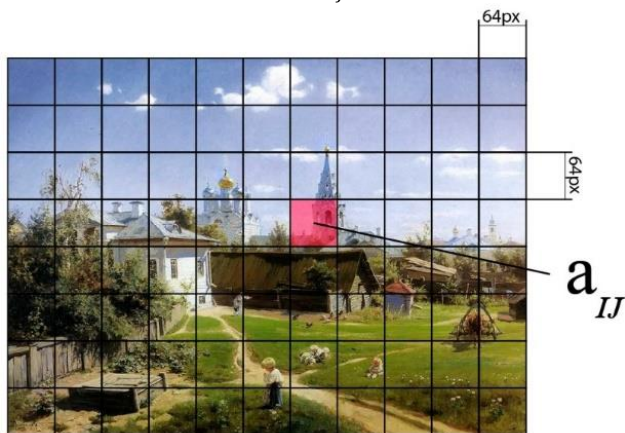


Рис. 1. Изображение, разбитое на блоки

2. Выбирается алгоритм расположения ЦВЗ, являющихся матрицей $w_{(m,n)}$ размером 4 x 4 пикселя, в этих блоках. Целесообразно встраивать в каждый блок по 4 ЦВЗ, отступая от края не менее чем на 3 пикселя, но располагая их несимметрично относительно этого блока (рис. 2).

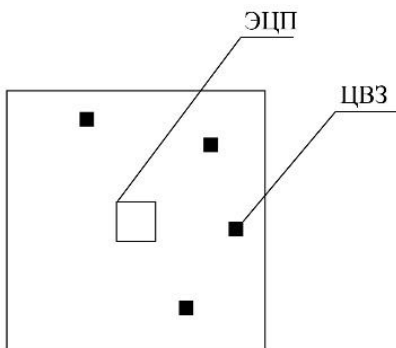


Рис. 2. Пример встраивания ЦВЗ и ЭЦП в блок изображения

3. ЦВЗ встраиваются в каждый блок изображения согласно выбранному алгоритму.

Каждую ЭЦП необходимо встраивать чётко посередине каждого блока, не пытаясь выбрать наиболее удачные для робастности участки. Это связано с тем, что полное или частичное отсутствие в проверяемом изображении этих центральных меток будет свидетельствовать о том, что данная фотография подвергалась каким-либо воздействиям. При этом другие встраиваемые ЦВЗ должны быть максимально робастными к атакам, чтобы можно было говорить о нарушении авторских прав на это изображение.

Отличие встраиваемых ЭЦП друг от друга необходимо для определения применения к изображению процедуры кадрирования.

Вернёмся к алгоритму.

4. Выбирается алгоритм нанесения в блоки различных ЭЦП (рис. 3).

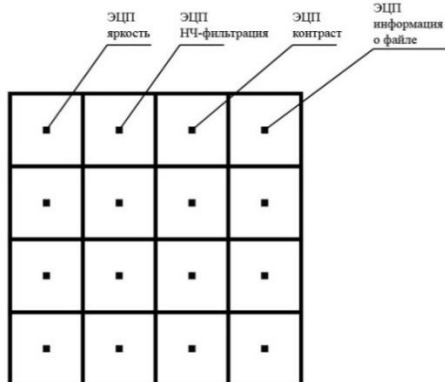


Рис. 3. Пример алгоритма нанесения ЭЦП в блок изображения

5. Составляются ЭЦП в зависимости от их назначения.
 6. ЭЦП встраиваются в блоки изображения в соответствии с выбранным алгоритмом.
 7. ЭЦП каждого блока корректируется в зависимости от соседних блоков.
- Следующим шагом изображения подвергается обратному вейвлет-преобразование, в результате чего становится меченым.

Выводы. Благодаря разложению исходного изображения на блоки, асимметричным блокам маркировки ЦВЗ и наличием в центре каждого блока подписи расположенной в соответствии с выбранным алгоритмом, при проверке подлинности изображения, можно определить, подвергается ли он резке, фильтрования или другим атакам. Поэтому мы в своей работе по защите мультимедийных продуктов остановимся именно на этом подходе.

Список литературы

1. Особенности формата JPEG. [Электронный ресурс] Режим доступа: <https://ru.wikipedia.org/wiki/JPEG>
2. Пиратство. [Электронный ресурс] Режим доступа: http://ru.wikipedia.org/wiki/Нарушение_авторского_права
3. Особенности формата AVI [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/Audio_Video_Interleave
4. Цзянь Ван. Исследование устойчивости цифровых водяных знаков и логотипов, внедряемых в статические изображения: автореф. СПб, 2010. 20с.
5. Стеганография, цифровые водяные знаки и стеганоанализ / Аграновский А.В. [и др.]. М.: Вузовская книга, 2009.

УДК 7.013

Принципы композиционного построения в современных плакатах Жиленкова Е.Ю., Харитонова В.В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
E-mail: zhilenkova18@gmail.com

Актуальность исследования. В современном обществе существует очень высокая потребность в плакатах, постерах, афишах. Это легкий и быстрый способ донести информацию до любого человека, поэтому они должны быть максимально простыми, четкими и понятными. В этом дизайнерам помогают принципы композиционного построения.

Цель. Проанализировать основные законы композиции, проследить применение принципов композиционного построения в современном плакате.

Основная часть. Гармония в композиционном плане есть согласованность, соразмерность частей (элементов) и целого. Этот поиск и есть выражение того общего характера формы, который обуславливает достижение наиболее целостного и глубокого от нее впечатления. Согласованная в частях, гармоничная форма убеждает, выглядит совершенной, собранной, красивой. Гармоничность – важнейший, не зависящий от вкуса признак выразительной композиции. Один из средств достижения – симметрия - асимметрия.

Симметрия – асимметрия

Эта пара средств определяет расположение элементов композиции относительно главной оси. Если оно одинаково, то композиция выступает как симметричная, если в нем есть небольшое отклонение в ту или иную сторону – как дисимметричная. При значительном таком отклонении она становится асимметричной.

Существуют три основных вида симметричной композиции: зеркальная, осевая и винтовая. Зеркальная симметрия образуется при одинаковом расположении элементов относительно главной оси, проходящей по центру горизонтальной или вертикальной композиционной плоскости.



Рис. 1 – Пример симметрии
 Лаборатория «Начала» —
 история джаза



Рис.2 – Пример симметрии
 Ian Jepson

Симметрия обеспечивает предельно четкое зрительное равновесие композиционной формы. Всякое ее нарушение ведет к тому, что эта форма приобретает неуравновешенный характер. Однако дисимметричная, как и асимметричная композиции сохраняют целостность в том случае, когда фактическая их неуравновешенность устранивается общим зрительным равновесием формы.



Рис. 3. – Зрительное равновесие Xavier Esclusa Trias / eight8

Нюанс – контраст

Эта пара средств гармонизации характеризует степень сходства и различия композиции. Она может быть выявлена только при сравнении элементов по одному композиционному свойству, например размеру или геометрическому виду.

Тождество. Означает не просто схожесть, а аналогию элементов. Оно может быть полным (абсолютным) и частичным.

Полное сходство выражается в одинаковости элементов по всем их композиционным свойствам, например размеру, цвету, пластике и т. д. На его основе строятся так называемые раппортные композиции.



Рис. 4– Тожество элементов Pierre Mendell

Частичное сходство означает отношение в целом одинаковых элементов, имеющих небольшое различие по какому-то одному признаку. Выражается оно в композиционной связи аналогичных по размеру, расположению, конфигурации, фактуре и несколько отличающихся по цвету элементов.

Нюанс. Характеризуется слабым различием элементов композиции по основным композиционным признакам. Например, в фактурных поверхностях он представляет крупную и мелкую зернистость, в цвете – оранжево-красные и красные оттенки и т. д.

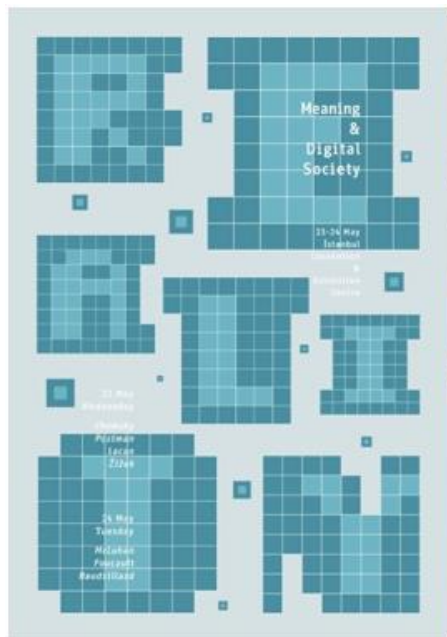


Рис.5 – Ньюанс в цвете
Geray Gencer



Рис.6 – Ньюанс элементов
João Machado

Контраст. Представляет собой резкое отличие элементов композиции. Сдержанный контраст проявляется при резком различии второстепенных элементов. Обостренный контраст имеет место при резком отличии главного элемента от второстепенных элементов. Он может быть выражен в размерах, расположении, цвете, пластике формы.

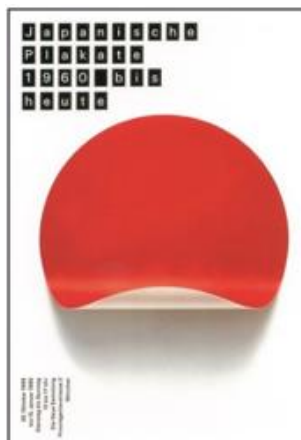


Рис. 7– Контраст в цвете
Pierre Mendell

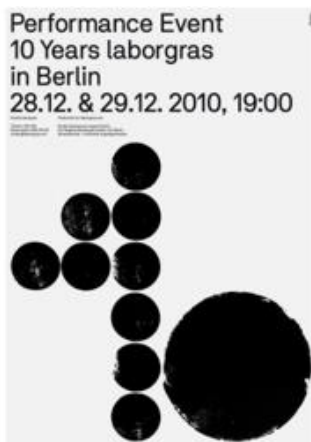


Рис. 8– Контраст в размере
Neubau Berlin

Полярность. Характеризуется тем крайним состоянием контраста, при котором наиболее ярко проявляется полное различие элементов по всем их композиционным свойствам.

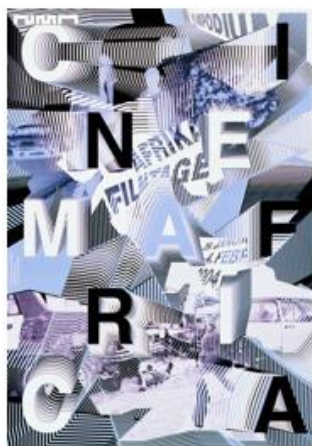


Рис. 9 – Использование
полярности
Ralph Schraivogel



Рис. 10 – Использование
полярности
Андрей Логвин

Метр – ритм

Метр – простейший порядок, основанный на повторении равных элементов. Повтор облегчает восприятие формы, делает ее четкой и ясной. Однако при большой протяженности метрическая композиция может выглядеть монотонной.



Рис. 11– использование метра
Ian Jepson

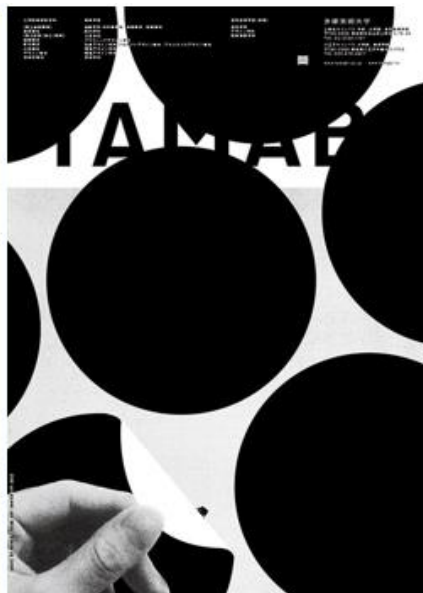


Рис.12– использования метра Kenjiro
Sano — Made by Hands

Ритм – более сложный, чем метр, порядок чередования элементов композиции. Он основан на неравномерном измерении их свойств. Это изменение может касаться как самих элементов, так и интервалов между ними. При постоянном их изменении образуется непрерывное множество, которое может носить разный характер – резко или плавно изменяющийся. Ритм, может развиваться в разных координатных направлениях.

Наиболее характерный прием построения ритма – изменения величины элементов. Другой распространенный прием – изменение интервалов между элементами.

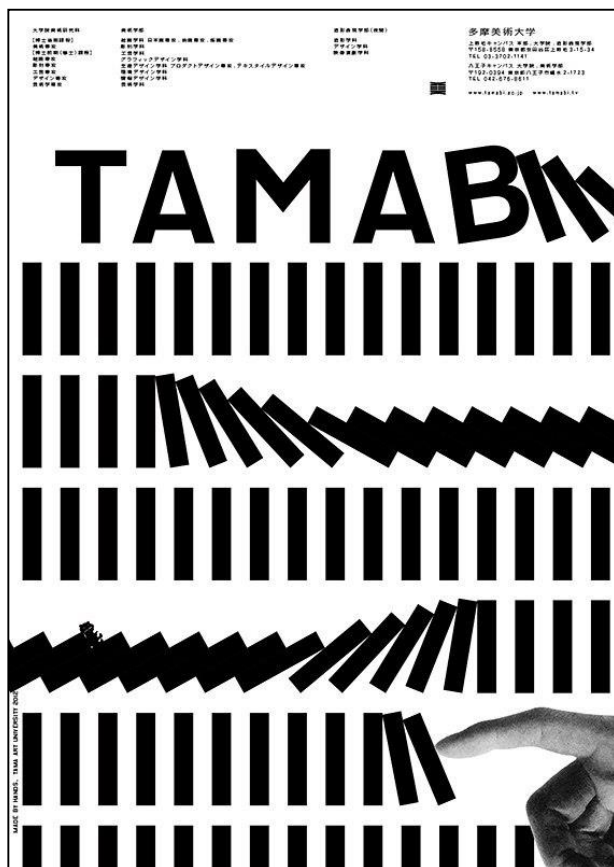


Рис.13 – Пример использования ритма Kenjiro Sano — Made by Hands

Статика – динамика

Данная пара средств гармонизации используется для выражения степени стабильности композиционной формы. Такая стабильность оценивается чисто эмоционально, по тому впечатлению, которое форма производит на зрителя. Это впечатление может исходить как из физического состояния формы – стабильного или динамичного, связанного с движением объекта в целом или его частей, так и чисто композиционного (формального). По степени зрительной и физической стабильности формы можно разделить на следующие четыре вида.

К первому виду относятся зрительно и физически статичные формы. К ним можно отнести: квадрат, прямоугольник, параллелепипед, положенный на

широкое основание, куб, пирамиду и т.п. Композиция, составлена из подобных форм, носит монументальный, предельно статичный характер.

Второй вид представляют физические статичные, но зрительно динамичные формы, оцениваемые так по впечатлению их некоторой неуравновешенности. Эта оценка касается стационарных форм, устремленных, например, в одном направлении, с нарушенной симметрией и другими специфичными для динамичных композиций свойствами.

Третий вид представляют зрительно статичные, но физически частично динамичные формы. Они имеют устойчивую основу, в которой «двигаются» отдельные элементы. Часто в практике дизайнера такое «движение» обусловлено особенностями функционирования объектов, реальным движением в них отдельных деталей.

Последний вид – зрительно и физически полностью динамичные формы. Они типичны для форм многих современных движущихся дизайн-объектов, прежде всего различных транспортных средств.



Рис. 14– Зрительно и физически статичные формы
Miriam König — Bergwerk



Рис. 15 – Физические статичные, но зрительно динамичные формы
Neubau Berlin

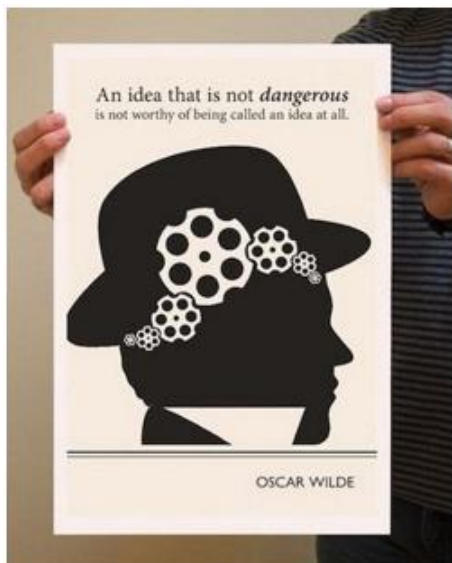


Рис.16 – Зрительно и физически полностью динамичные формы
Kazumasa Nagai



Рис.17 – Зрительно статичные, но физически частично динамичные формы
Kazumasa Nagai

Выводы. В ходе исследования были проанализированы основные законы композиции, такие как: симметрия – асимметрия, нюанс – контраст, метр – ритм, статика – динамика. Применение основных законов композиционного построения делает плакат ярким, запоминающимся, собранным, понятным для любого человека.

Список литературы

1. Устин, В. Б. Композиция в дизайне. Методические основы композиционно-художественного формообразования в дизайнерском творчестве: учебное пособие. – 2-е изд., уточненное и доп./ В. Б. Устин. – М.: АСТ: Астрель, 2007. – 239, [1] с.: ил.
2. Posterino – искусство плаката [электронный ресурс] URL: <https://new.vk.com/posterino>
3. CROCUS posters [электронный ресурс] URL: <https://new.vk.com/crocuspsters>

УДК 004.92

Виды инфографики

Давыденко Д.П., Харитонов В.В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерного моделирования и дизайна
E-mail: darinochka.davidenko@yandex.ru

Аннотация

Давыденко Д.П., Харитонов В.В. Виды инфографики. Выполнен краткий анализ самых распространенных видов инфографики. Рассмотрены некоторые виды классификаций инфографики.

Ключевые слова: инфографика, визуальная коммуникация.

Постановка проблемы. Сегодня для современного общества характерно возрастание информационной активности, что проявляется в резком увеличении темпов и объемов производства и потребления информации. Информация, представленная визуально, воспринимается легче и быстрее.

Поэтому так актуальна инфографика как средство визуальной коммуникации. Для эффективного использования визуальных образов важно правильно ориентироваться в многочисленных видах информационной графики, у каждого из которых, имеются свои особенности и задачи.

Цель статьи – провести краткий анализ самых распространенных видов инфографики, обозначить области их применения и привести примеры.

Анализ видов инфографики. Инфографика (от лат. informatio — осведомление, разъяснение, изложение - греч. γραφικός — письменный, от γράφω — пишу) — это графический способ подачи информации, данных и знаний[2]. Инфографика дает возможность представить большой объем самой разной информации в сжатом и удобном для восприятия виде. Часто графический компонент выступает в качестве дополнения к текстовой информации. Инфографика может комбинировать фотографии, текстовые блоки, таблицы, диаграммы, реконструкцию событий. Она может быть представлена в виде комиксов, карикатур, эмблем, иллюстраций, анимации, видео.

Ключевыми форматами инфографической коммуникации являются:

- статические изображения с фиксированной информацией, предназначенной для чтения и просмотра.
- движущиеся изображения (анимация или видео-формат) с фиксированной информацией, которую пользователь смотрит, читает, слушает.
- интерактивные интерфейсы с фиксированной или обновляемой информацией, которую пользователь может искать, преобразовывать

отображаемое содержимое, выбирая, что именно и с каким сюжетом развития должно быть визуализировано.

– По типу источника информации различают 3 основных вида инфографики:

– аналитическая инфографика — графика, подготавливаемая по аналитическим материалам.

– новостная инфографика — инфографика, подготавливаемая под конкретную новость в оперативном режиме,

– инфографика реконструкции — инфографика, использующая за основу данные о каком-либо событии, воссоздающая динамику событий в хронологическом порядке.

Существует множество различных классификаций инфографики по различным критериям. Рассмотрим некоторые из них.

По характеру визуализации различных типов информационных данных инфографику можно разделить на количественную и качественную.

Для визуального представления количественных (числовых) данных используются количественная инфографика: графики, диаграммы, гистограммы и номограммы, которые, в свою очередь подразделяются на подгруппы (точечные, линейные круговые и т.п.).

Для изображения совокупности (например, иерархий) объектов и других качественных данных применяется качественная инфографика [3]:

– организационные диаграммы, отражающие структуру объекта (например, темы дисциплины, состава устройства или организации), функции и оптимальные связи подчиненных структур с целью оптимизации работы объекта;

– диаграммы трендов, определяющие стратегию развития процессов. На такой диаграмме все показатели представлены в наглядной форме, облегчающей планирование и принятие решений;

– планы-графики, четко определяющие последовательность решения задач в ходе того или иного процесса, которые, в частности, используются для планирования учебного процесса;

– технологические диаграммы, на которых четко представлены особенности и стадии технологических процессов;

– диаграммы и схемы компьютерных сетей и телекоммуникаций, используемые как на стадии разработки технических концепций формирования компьютерных сетей, так и в дальнейшем процессе их функционирования;

– разнообразные схемы, при помощи которых акцентируется внимание на наиболее важных моментах при обмене информацией;

– рисунки и схемы, алгоритмы, наглядно и доступно представляющие тот или иной процесс или ситуацию;

– графы и деревья, визуализирующие переходы или связи понятий, событий, процессов;



Рис.2 – Инфографика “Истина в воде”

Выводы. В данной работе мы сделали лишь краткий обзор различных видов инфографики. Инфографика как средство визуальной коммуникации будет становиться в дальнейшем все более актуальной и постоянно развиваться. Несколько лет назад была распространена только статичная инфографика, затем появилась динамичная анимированная инфографика, а сейчас развивается интерактивная инфографика. Возможности информационной среды развиваются настолько быстро, что можно предположить зарождение интеллектуальной инфографики.

Список литературы

1. Lankow, Ritchie, Crooks, 2012, INFOGRAPHIC FORMATS: CHOOSING THE RIGHT VEHICLE FOR YOUR MESSAGE.
2. Инфографика [электронный ресурс] // Википедия – свободная энциклопедия: [сайт]. [2012]. URL: <https://ru.wikipedia.org/wiki/Инфографика>
3. Никулова Г. А. Средства визуальной коммуникации — инфографика и метадизайн /Г. А. Никулова, А. В. Подобных // международный электронный журнал КНИТУ

УДК 004.92

Некоторые правила применения цветовых комбинаций в Web-дизайне

Касияненко Д.И., Харитонова В.В.

Донецкий национальный технический университет, г. Донецк

Кафедра компьютерного моделирования и дизайна

DashaKasiyanenko@gmail.com

Аннотация

Касияненко Д.И., Харитонова В.В. Некоторые правила применения цветовых комбинаций в Web-дизайне. В статье рассмотрен выбор цветов при разработке информационных систем. Проанализированы ассоциации, которые вызывает цвет у пользователя и их соответствие интерфейсу программного продукта. Также рассматриваются решения данной проблемы, правильный выбор цветов и их сочетаний.

Ключевые слова: цвет, интерфейс, сочетания цветов.

Актуальность исследования. Цвет имеет уникальные психофизиологические свойства воздействия, поэтому играет важную роль при создании веб-сайтов. Так как современный человек длительное время проводит на просторах интернета нужно, чтобы сайт не только выглядел привлекательно для глаз, но и своей цветовой гаммой не наносил вред здоровью пользователя.

Цель исследования. Рассмотреть основные свойства воздействия цвета на человека, сформулировать правила использования цветовых комбинаций при разработке дизайн-макетов сайтов.

Психологические, физиологические и физические факторы восприятия цвета. Для целенаправленного использования цвета необходимо знать устойчивые связи между цветом и психологической реакцией человека. В этом аспекте установлен ряд закономерностей – по предпочтительности цветового тона, по образной ассоциативности цветов, по гармоничности цветосочетаний.

Цвета, окружающие каждого из нас, оказывают воздействие на организм, нервную систему и психику человека. Ведущим фактором в формировании цветовых значений является объективный характер цветового воздействия на человека, как об этом свидетельствуют данные психофизиологических и психологических исследований.

Большая интенсивность цвета, действующая на человека длительное время, утомляет зрение, яркая окраска надоедает и раздражает.

Привлекательность тенденции ярких цветов в том, что ее можно использовать многими разными способами в разных стилях и ко множеству различных элементов.

Посмотрим на лучшие примеры реализации, которые только выиграли от использования ярких цветов в трех разных областях.

Кнопки. Так же как слова, яркий цвет может акцентировать внимание на кнопки или другие элементы, использующиеся в интерфейсе, как для привлечения внимания, так и для облегчения использования.

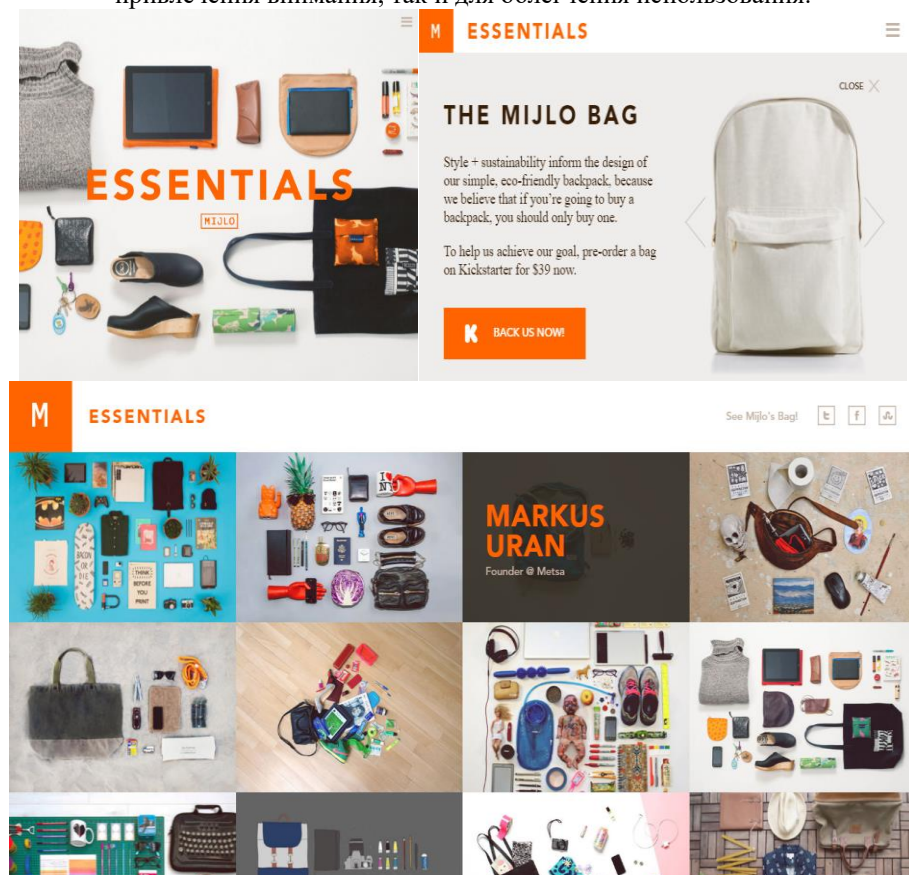


Рис.1 – Выделение ярким цветом ключевых элементов

Ярко-оранжевая кнопка на сайте, который представлен выше, выделяется на простом фоне и изображении товара. Комбинирование цветов фона и товара вместе приводят к тому, что нужно сделать кнопку яркой, которая будет выглядеть более привлекательно.

2. Выделение цветом элементов навигации. Сила организации материалов с использованием цвета была известна давно и ее можно увидеть в цветных закладках в книгах или при организации картотек в ящиках. Его применение в веб навигации кажется логическим развитием, так как разными цветами легко разделить разные секции. Это станет особенно полезно для сайтов со сложным контентом, где каждое усилие должно быть направлено на упрощение навигации.



Рис. 2 - Фото: <http://colouredlines.com.au/>

Например, Colored Lines на примере, использует спектры радуги для разделения восьми отдельных страниц их сайта. Это не только упрощает объёмную навигационную систему, а так же представляет идентификацию бренда, с учетом названия компании.

Использование неоновых цветов. Неоновые цвета могут довольно нестандартными и яркими и добавить дизайну необходимых в некоторых случаях качеств. Но такие цвета очень сложны для восприятия. Прибегая в дизайне к таким решениям, следует понимать, что долго смотреть на такое произведение никто не сможет - попросту будут болеть глаза.



Рис. 3 – Использование неоновых цветов

Проблема в том, что буквы для чтения слишком яркие, что вызывает проблемы с комбинированием темного и светлого фона. Неоновый фон часто используют для отвлечения внимания от основного сообщения в дизайне.

Решение: следует уменьшить яркость цветов чтобы они получились более тёмными и лучше выглядели на экране.

Это одна из самых наиболее распространенных ошибок. Это допускается по разным причинам, начиная от неправильно настроенного монитора, до недостатка практики. Но использование светлых элементов на светлом фоне практически всегда нечитабельно.



Рис. 4 – Светлые элементы на светлом фоне

Использование многоцветного фона. Уже давно известно, что радужные темы не работают. В них нарушаются почти все правила теории выбора цвета для дизайна. Такой фон на некоторое время захватывает, но надоедает.



Рис. 5 – Использование многоцветного фона

Решение: если разработчику все же необходимо уместить множество цветов в одном дизайне, то следует это сделать на основе блоков или карточек, где цвета могут ассоциироваться с определёнными функциями и элементами интерфейса. Это позволит повысить гибкость цвета, создаст ощущение организации и движения.



Рис. 6 – Многоцветный фон на основе блоков

Использование контрастных и нюансных цветовых сочетаний. Ещё один совершенно провальный приём, которым можно серьёзно усложнить жизнь пользователям вашего дизайна, это использование темных по цвету элементов на темном фоне.



Рис. 7 – Не верное использование цветов

Как и "светлое на светлом", эта ошибка часто случается из-за недостатка опыта либо ошибочного представления о том, что на всех мониторах одинаковая цветопередача.

Решение: Следует использовать контрастные сочетания цветов

Не нужно использовать чистый черный цвет. Придайте ему реальности. В качестве фона всегда используйте комбинацию черного с другими цветами.



Следует помнить, что чисто черный цвет используется исключительно для печати. Его использование в дизайне - это не очень хорошая практика.

Решение: Использовать такой цвет необходимо по минимуму и разбавлять его акцентами другого цвета.

Выводы. В настоящем исследовании мы рассмотрели вопросы использования различных цветовых комбинаций для фона и элементов сайта, с целью вызвать те или иные ощущения у пользователя. Так же были обозначены некоторые правила сочетания цвета в веб-дизайне. Цвет – неотъемлемая часть дизайна веб-приложений, и, в независимости от текущего тренда, разработчики макетов приложений обязаны учитывать особенности психофизиологического влияния цвета на человека с целью создания безопасного и гармоничного продукта.

Список литературы

1. Тренды в Web-дизайне/ Интернет-ресурс. – <http://webformyself.com/trendy-veb-dizajna-na-2015-i-2016-gody-besposhhdnye-cveta/>
2. Какие цвета не использовать в веб-дизайне?! Интернет-ресурс. - <http://skillsup.ru/blog/interfejsyi/kakie-czveta-ne-ispolzovat-v-veb-dizajne.html>

УДК 519.683.8

Разработка обучающего пакета программ работы с контейнерными классами

Ольшевский А.И., Стась А.В.

Донецкий национальный технический университет

E-mail: a_olshevskiy@mail.ru

Аннотация

Ольшевский А.И., Стась А.В. Разработка обучающего пакета программ работы с контейнерными классами Рассмотрена классификация по типам основных структур данных контейнерных классов. Описаны классы и различные конструкторы для создания этих структур. Определен подход формирования поддерживающий разные наборы операций, среди которых есть совпадающие операции. Проанализирована эффективность разных методов соответствующих операций.

Общая постановка проблемы. Для высшего образования первостепенную актуальность приобретает задача использования возможностей информационных технологий в обучении. Программирование – это искусство мыслить и достигать успеха. Процесс реформирования знаний рассматривается в контексте создания таких педагогических технологий, которые обеспечивали бы переход от формально-дисциплинарного к проблемно-активному типу обучения [1].

Язык С++ захватил пальму первенства по количеству приверженцев среди программистов. Ведь он является преемником языка Си – огромного количества наработок, готовых библиотек и примеров. Огромный плюс – это кроссплатформенность. Можно создавать программы любой сложности для Windows, Linux и MacOS.

Постановка задачи. Объектом разработки является шаблоны классов, часть библиотеки С++, в которую входят контейнерные классы, а также алгоритмы и итераторы стандартной библиотеки шаблонов (STL — Standard Template Library).

Цель проекта – накопление и систематизация информации для обучения пользователей возможностью стандартной библиотеки шаблонов содержащей контейнеры, реализующие основные структуры данных, для проектирования и написания программ. Использование контейнеров позволяет значительно повысить надежность программ, их переносимость и универсальность, а также уменьшить сроки их разработки.

0 алгоритмы работают с заданным диапазоном элементов контейнера.

Итераторы – это объекты, которые в той или иной степени действуют подобно указателям. Они позволяют циклически опрашивать содержимое

контейнера практически так же, как это делается с помощью указателя при циклическом опросе элементов массива.

Программная реализация

Обучающий пакет программ предоставляет пользователям теоретическую информацию по типам основных структурам данных контейнерных классов, примеры программ реализации алгоритмов и итераторов при работе со стандартной библиотекой шаблонов.

Смысл одноименных операций для различных контейнеров одинаков, основные операции применимы ко всем типам контейнеров. Стандарт определяет только интерфейс контейнеров, поэтому разные реализации могут сильно отличаться по эффективности.

Векторы (vector), двусторонние очереди (deque) и списки (list) поддерживают разные наборы операций, среди которых есть совпадающие операции. Они могут быть реализованы с разной эффективностью.

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.

Выбор контейнера позволяет изучить теоретическая информация по этому виду объекта и обеспечить выбор своего набора действий над данными (программную реализацию). Выбор вида контейнера зависит от того, что требуется делать с данными в программе.

Теоретическая информация по типу контейнера. Например: вектор – это структура, эффективно реализующая произвольный доступ к элементам, добавление в конец и удаление из конца; двусторонняя очередь эффективно реализует произвольный доступ к элементам, добавление в оба конца и удаление из обоих концов; список эффективно реализует вставку и удаление элементов в произвольное место, но не имеет произвольного доступа к своим элементам и т.п.

Векторы, двусторонние очереди и списки поддерживают разные наборы операций, среди которых есть совпадающие операции. Они могут быть реализованы с разной эффективностью.

В теоретической части приводиться схемы организации данных, как обеспечить доступ к элементам и за какое время, как будет организованно выделение память под очередной элемент или перераспределение памяти при удалении.

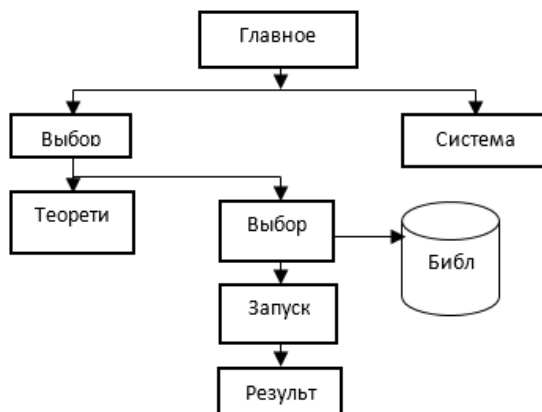


Рис. 1. Общая структура вызовов элементов программы и их взаимодействия

Библиотека программ содержит пакет обучающих программ работы с контейнерными классами, которые могут пополняться.

Программист может создавать собственные контейнерные классы на основе имеющихся в стандартной библиотеке.

Выбор программной реализации позволяет выбрать программу из набора, запустить ее и получить результат. Например работы с вектором: в файле находится произвольное количество целых чисел; программа считывает их в вектор и выводит на экран в том же порядке.

Выводы. Были определены цели, которые можно достичь, используя обучающий пакет программ, и сформулированы требования, которые должны выполнять программы реализующие методы и операции при работе с основными структурами данных контейнерных классов. Так же, был проведен анализ эффективности использования операций различными способами. Были выделены их недостатки и преимущества, которые будут учитываться при разработке программного обеспечения.

Список литературы

1. Юрков Н. К. Интеллектуальные компьютерные обучающие системы / Юрков Н. К. [Электронный ресурс]. – Режим доступа : http://www.mtas.ru/upload/library/MONOGRAFIYA_IKOS_2010.pdf

2. Павловская Т. А. С/C++. Программирование на языке высокого уровня / Павловская Т. А. – СПб.: Питер, 2003. – 461 с.

3. Шилдт Г. С++: базовый курс / Шилдт Г. – [3-е изд.] ; [пер. с англ.]. – М.: Вильямс, 2010. – 624 с.

Секция № 8
Компьютерная
инженерия



УДК 004.2

Система контроля микроклимата жилого дома

Селеменев В.Д., Теплинский С.В.

Донецкий национальный технический университет, г. Донецк

кафедра компьютерной инженерии

E-mail: aladik17@list.ru

Аннотация

Селеменев В.Д., Теплинский С.В. Система контроля микроклимата жилого дома. Произведен анализ платформы Arduino, изучены основные ее характеристики и дополнения. Выполнен выбор аппаратных средств и разработана система контроля микроклимата жилого дома на основе платы Arduino Uno.

Ключевые слова: Arduino Uno, микроконтроллер, роутер, датчик температуры, датчик влажности, датчик давления, плата расширитель Ethernet.

Постановка проблемы. Системы типа «умный дом» в настоящее время набирают популярность, но из-за высокой стоимости их не каждый может себе позволить. В связи с этим была выбрана для реализации система контроля микроклимата жилого дома. Для создания данной системы необходимо выполнить анализ составляющих платы Arduino, определить оптимальные аппаратные средства для системы контроля микроклимата жилого дома, представить структуру системы и реализовать ее.

Цель статьи – выполнить анализ платформы Arduino и произвести выбор аппаратных средств для системы контроля микроклимата жилого дома. Реализовать данную систему с использованием плат и компонентов Arduino.

Анализ платформы. Arduino – это набор программируемых микропроцессорных плат, которые ориентированы на получение данных об окружающем мире. Кроме того, существует огромное множество дополнительных компонентов, которые расширяют возможности стандартных плат. Такие компоненты можно разделить на датчики и платы расширений (шилды). С помощью датчиков и плат расширений мы можем получать информацию, обрабатывать ее и выдавать результат. Проекты Arduino могут быть как самостоятельными, так и взаимодействовать с другими устройствами или программным обеспечением, работающем на персональном компьютере, телефоне с ОС Android, iOS или через Internet.

Любую плату Arduino можно собрать самому или же купить готовое устройство. Оригинальную плату можно приобрести не более чем за 25\$, а ее копию от 2\$ и выше.

Плат Arduino большое количество. Они предназначены для различных целей, следовательно, имеют разнообразные характеристики. В основном они отличаются между собой объемами памяти, моделью микроконтроллера, количеством входов/выходов и размерами самой платы.

Все особенности программирования под платформу Arduino сводятся к тому, что имеется набор библиотек, включающий в себя некоторые функции и объекты[3]. Код программы для Arduino называется скетч. В скетче обязательно должны присутствовать две основных функции `setup()` и `loop()`, без них программа не от компилируется. Упрощенно цикл программирования Arduino можно представить в следующей последовательности:

- Подключение платы через USB-порт к компьютеру;
- Написание скетча программы;
- Выгрузка этого скетча на плату через USB-соединение;
- Выполнение платой написанного скетча.

Для тестирования программ можно использовать программу VBB 4.2.9 (Virtual Breadboard), которая эмулирует работу с платами Arduino. Эта программа позволяет писать код прямо в эмуляторе (не используя IDE) и тут же посмотреть, как он работает. Так же, программа включает в себя отладчик и достаточный набор компонентов для постройки различных схем. Существует еще программа эмулятор Fritzing, она подобна VirtualBreadboard. Есть так же онлайн версия эмулятора, которая открыта для использования.

Таким образом, используя библиотеки, функции и объекты для программирования Arduino реализуется необходимая система контроля.

Необходимые аппаратные средства для реализации системы контроля микроклимата жилого дома. Чтобы система правильно функционировала необходимо выбрать аппаратные средства, которые будут соответствовать поставленной задаче и будут оптимальны по функциональности. Для системы контроля микроклимата жилого дома были выбраны плата Arduino Uno, датчик температуры DS18B20, датчик влажности DHT22, датчик давления BMP180, плата расширитель Ethernet Shield W5100, два резистора номиналом 4.7кОм и 10кОм.

Структура системы контроля микроклимата жилого дома изображена на рис. 1.

Рассмотрим более детально каждый элемент. Начнем с основной платы Arduino Uno, которая изображена на рис. 2 с описанием составляющих элементов.

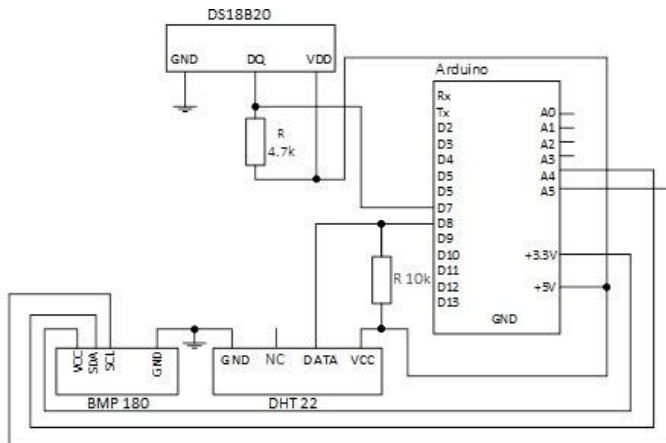


Рис. 4 – Структура системы контроля микроклимата жилого дома

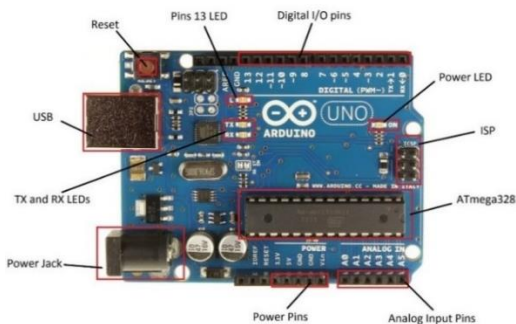


Рис. 5 – Плата Arduino Uno с описанием составляющих элементов

Arduino Uno – это устройство на основе микроконтроллера ATmega328. Данная платформа может работать при внешнем питании от 6 В до 20 В. Если напряжение питания ниже 7 В платформа может работать нестабильно. Если использовать напряжение выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Далее рассмотрим датчик температуры (см. рис. 3). Микросхема DS18B20 это термометр с цифровым выводом, работающий с точностью $\pm 0.5^{\circ}\text{C}$. Рабочий диапазон от -55 до $+125^{\circ}\text{C}$. Данные считываются через последовательную шину. Время получения данных: максимальное 750мс, минимальное: 94мс. Напряжение питания: 3–5,5В. Потребляемый ток при бездействии: 750нА. Потребляемый ток при опросе: 1мА.



Рис. 6 – Датчик температуры DS18B20 в корпусе и без корпуса

Данный датчик продается как в корпусе, так и без него. Для контроля температуры был выбран датчик в корпусе, так как это более практично в использовании, например, можно измерять температуру воды.

Следующим рассмотрим датчик влажности и температуры. DHT22 – это цифровой датчик температуры и влажности. Основные параметры датчика:

- Диапазон измерения влажности: 0 ~ 100% с погрешностью $\pm 2\%$
- Диапазон измерения температуры: $-40 \sim 125^{\circ}\text{C}$ с погрешностью $\pm 0,5^{\circ}\text{C}$
- Напряжение питания: 3,3 - 6В.
- Частота измерений до 0.5Гц (одно измерение за 2 секунды).
- Размер корпуса: 15.1мм x 25мм x 7.7мм.
- Максимально потребляемый ток – 2.5мА при преобразовании (при запросе данных).

Датчик DHT22 с описанием ножек изображен на рис. 4.

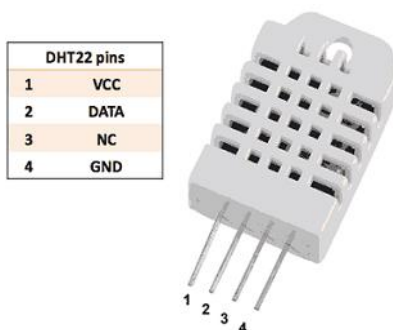


Рис. 7 – Датчик влажности и температуры DHT22

Для работы с датчиком в среде Arduino необходимо установить дополнительную библиотеку DHT.

В качестве датчика атмосферного давления используется BMP180 (см. рис.5). Этот датчик является более экономным в плане электропитания и более точным, чем его предшественник BMP085[1].

Рассмотрим основные характеристики данного датчика:

- Диапазон: 300гПа..1100гПа (от +900мм.рт.ст. до -500мм.рт.ст.);
- Напряжение питания: от 3.3В и 5В;
- Низкое энергопотребление: 3мкА (режим ультранизкого энергопотребления);
- Точность: 0.06гПа (0.5мм.рт.ст.);
- Интерфейс подключения: I2C;
- Возможность измерения температуры в диапазоне: -40..+85 °С;
- Время отклика: 5мс;
- Ток в режиме ожидания: 0.1мкА.

Кроме того, с помощью преобразования абсолютного давления в высоту может использоваться как датчик высоты. Позволяет достичь точности определения высоты порядка 1м.

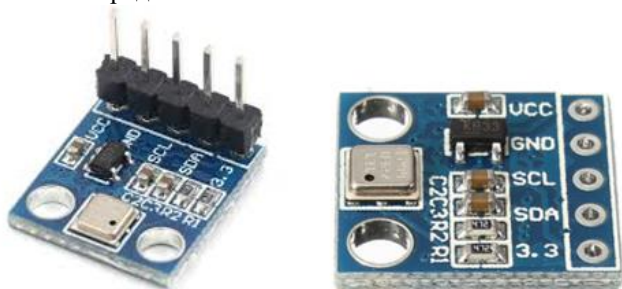


Рис. 8 – Датчик атмосферного давления BMP180

Данный датчик имеет 5 контактов:

- VCC для подключения 5В питания;
- GND для подключения к земле;
- SCL и SDA для подключения к шине I2C;
- 3.3 для подключения 3.3В питания.

Для получения доступа к сети Интернет используется плата расширитель Ethernet Shield W5100[2]. Она дает возможность управлять устройством дистанционно через интернет, получать информацию от различных онлайн-сервисов и обрабатывать ее, а также запустить небольшой веб-сервер. Плата расширитель Ethernet изображена на рис. 6. Плата построена на базе Ethernet-контроллера Wiznet W5100. Она совместима с Arduino Uno, но для работы с ней необходимо подключить библиотеку Ethernet. На плате также имеется слот для

microSD-карты объёмом до 2 ГБ для хранения файлов и организации сетевого доступа к ним. Питается плата расширитель Ethernet через основную плату Arduino Uno от внешнего источника питания.

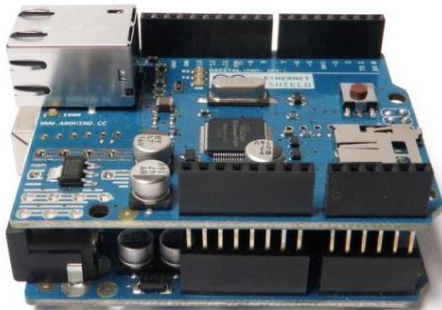


Рис. 9 – Плата расширения Ethernet Shield и Arduino Uno

Разработка системы контроля микроклимата жилого дома. Данная система необходима для того, чтобы контролировать показатели температуры, влажности и давления в жилом помещении[4]. Принцип действия заключается в следующем: данные считываются с датчиков, далее они передаются на плату Arduino Uno, после этого Arduino отправляет данные на плату расширения Ethernet, которая выступает в качестве сервера, далее формируем веб-страницу, на которой отображаются данные. После этого, пользователь может обратиться к ip-адресу роутера с указанием порта и получить доступ к необходимой веб-странице. Таким образом, имеется возможность удаленно контролировать микроклимат в жилом помещении. Результаты работы системы представлены на рис.7

В дальнейшем планируется усовершенствовать данную систему, чтобы непосредственно влиять на изменения температуры в доме.

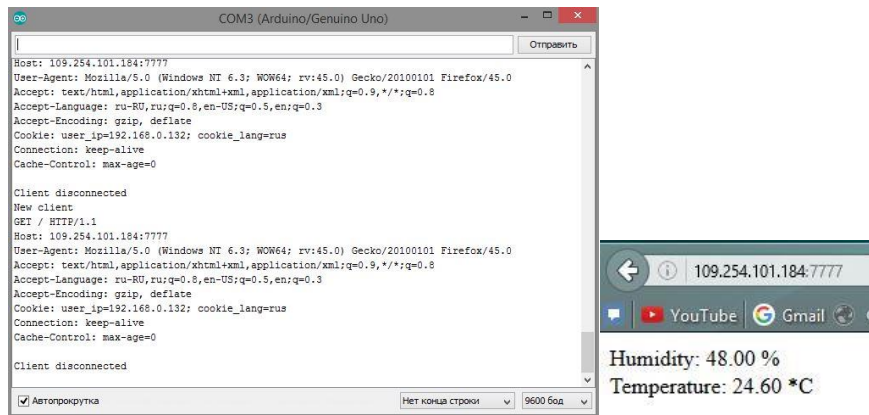


Рис.10 – Результаты работы системы контроля микроклиматом жилого дома

Выводы. Проведен анализ плат платформы Arduino и реализована система контроля микроклимата жилого дома. Успешно произведены снятия и обработка показаний с датчиков, вывод результатов на экран персонального компьютера через сеть Интернет, организованной с помощью платы расширения Ethernet. Полученные результаты совпадают с реальными данными.

Список литературы

1. Паяльник. [электронный ресурс] // Сайт Паяльник: [сайт]. [1999-2016]. URL: <http://cxem.net>
2. Амперка [электронный ресурс] // Амперка: [сайт]. [2010-2016]. URL: <http://amperka.ru>
3. У. Соммер. «Программирование микроконтроллерных плат Arduino/Freeduino». БХВ-Петербург, 2012 г. – 256 с.
4. Arduino-Diy. [электронный ресурс] // Ардуино: [сайт]. [2015-2016]. URL: <http://arduino-diy.com>

УДК 681.3

Разработка системы ориентирования в пространстве с использованием модуля GY-531

Горин Н.А., Струнилин В.Н.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерная инженерия
MegaNikolai@yandex.ua

Аннотация

Горин Н.А., Струнилин В.Н. Разработка системы ориентирования в пространстве с использованием модуля GY-531. Выполнена разработка электронного прибора отслеживающего при помощи акселерометра и гироскопа положение объекта и генерирующего сигналы управления для корректировки работы двигателей.

***Ключевые слова:** микроконтроллер, акселерометр, гироскоп, микросхема, драйвер.*

Постановка проблемы. При взаимодействии роботизированной техники с окружающими объектами возникает необходимость в системах ориентирования в пространстве. Эти системы необходимы так же в тех случаях, когда оператор какой-либо движущейся машины или устройства не может достоверно оценивать перемещение управляемого объекта в пространстве и его взаимодействие с другими объектами. С развитием научно-технического прогресса данная проблема становится всё актуальнее. 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, исследование и разработка систем ориентирования в пространстве является актуальной.

Цель статьи – разработка электронного прибора, отслеживающего при помощи акселерометра и гироскопа положение объекта в пространстве, генерирующего управляющие сигналы для корректировки работы двигателей.

Выбор платформы. Разработка системы ориентирования в пространстве выполнена на платформе ArduinoUno (рис.1). Данная платформа построена на основе микроконтроллера ATmega328 [1]. Платформа имеет 14 цифровых входов/выходов (шесть из которых могут использоваться как выходы широкополосной импульсной модуляции), шесть аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру

посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи [1]

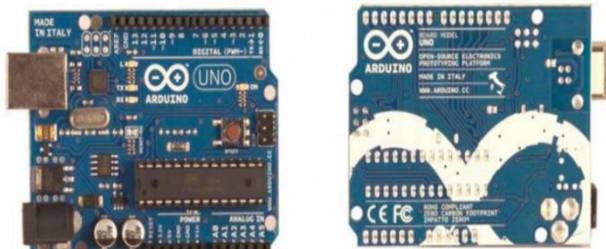


Рис.1 – Внешний вид ArduinoUno

Платформа программируется посредством программного обеспечения Arduino. Язык программирования устройств Arduino основан на C/C++ [2, 3].

Для программирования микроконтроллера используется среда разработки Arduino 1.0.5. Микроконтроллер ATmega328 поставляется с встроенным загрузчиком, облегчающим запись новых программ без использования внешних программаторов.

Обоснование выбора модуля GY-531. Микросхема MPU6050 содержит акселерометр, гироскоп и температурный сенсор. MPU6050 является главным элементом модуля GY-531 (рис.2). Помимо этой микросхемы на плате модуля расположена необходимая обвязка MPU6050, в том числе подтягивающие резисторы интерфейса I2C, а также стабилизатор напряжения на 3,3 вольта с малым падением напряжения (при питании в 3,3 вольта на выходе стабилизатора будет 3 вольта) с фильтрующими конденсаторами. Также на плате имеется SMD светодиод с ограничивающим резистором, выполняющие функцию индикации питающего напряжения.

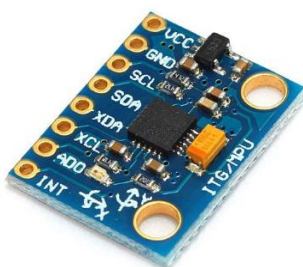


Рис. 2 - Внешний вид GY-531

Математический аппарат. Данные, приходящие с гироскопа имеют следующий вид:

$$gyro_x_scaled = \frac{d}{dt} \theta_x^{gyro} ;$$

$$gyro_y_scaled = \frac{d}{dt} \theta_y^{gyro} ;$$

$$gyro_z_scaled = \frac{d}{dt} \theta_z^{gyro} .$$

gyro_x_scaled, gyro_y_scaled и gyro_z_scaled – это переменные, в которые библиотечная функция MPU6050_ReadData() заносит значения, считанные с гироскопа с учётом калибровки.

В дальнейшем будем рассматривать все расчёты на примере оси x. Для расчета угла необходимо проинтегрировать переменную “gyro_x_scaled”.

Для реализации дискретного интегрирования, будем использовать метод Эйлера как один из самых популярных алгоритмов. Математически интегрирование методом Эйлера можно записать следующим образом:

$$\theta_x^{gyro}(t_n) = gyro_x_scaled * T + \theta_x^{gyro}(t_{n-1}) ;$$

$$T = t_n - t_{n-1} ,$$

где $n=\{1, 2, 3 \dots\}$ это количество итераций.

Предполагаем, что начальные углы относительно осей x, y, z после калибровки равны 0, 0 и 90 градусов соответственно, так что для итерации $n=0$:

$$\theta_x^{gyro}(t_0) = 0^{\circ}, \theta_y^{gyro}(t_0) = 0^{\circ}, \theta_z^{gyro}(t_0) = 90^{\circ} .$$

Из-за неидеальной калибровки гироскопа, “gyro_x_scaled” никогда не равна нулю и со временем “angle_x_gyro” изменяет свои значения. Для решения данной проблемы, проводится расчет угла с помощью акселерометра и полученные значения сравниваются с углом гироскопа. Так как модуль MPU6050 располагается горизонтально, ускорение по оси z равно 1g (то есть, 9,81 м/с²). Используем этот вектор ускорения и его проекцию на ось y для расчета угла между осями x и y.

Угол, который рассчитывается с помощью акселерометра, рассчитывается по зависимости:

$$\theta_x^{accel} = \tan^{-1} \frac{accel_x_scaled}{\sqrt{accel_y_scaled^2 + accel_z_scaled^2}} ;$$

$$\theta_y^{accel} = \tan^{-1} \frac{accel_y_scaled}{\sqrt{accel_x_scaled^2 + accel_z_scaled^2}} .$$

Основными проблемами при определении угла наклона с помощью акселерометра являются: сильная зашумленность сигнала и очень сильная чувствительность к вибрациям, без которых ни один механизм не работает. Более того, при перемещении MPU6050 вдоль одной из осей координат, полученные значения угловых скоростей будут мешать расчету угла. Для лучшего результата углы с гироскопа и акселерометра обрабатываются с помощью фильтра:

$$\theta_x = Filter_gain * \theta_x^{gyro} + (1 - Filter_gain) * \theta_x^{accel}$$

Окончательно уравнение для определения угла наклона принимает вид:

$$\theta_x^{gyro}(t_n) = gyro_x_scaled * T + \theta_x(t_{n-1})$$

В дальнейшем использовался коэффициент усиления 0,95. В зависимости от динамики системы, можно его повышать, но не до единицы, так как значения будут сильно отклоняться от истинных.

Принцип работы. Разработанная система ориентирования в пространстве должна определять угол подъёма или спуска движущегося объекта и в зависимости от этого генерировать управляющие сигналы для увеличения/уменьшения напряжения, подаваемого на двигатели. Если объект преодолевает подъём, система определяет угол подъёма и увеличивает напряжение, подаваемое на двигатели, чтобы не утратить скорость движения. Если объект преодолевает спуск, система определяет угол спуска и уменьшает напряжение на двигатели чтобы предотвратить неконтролируемый разгон объекта.

Моделирование производилось в утилите Virtual Bread Board. Virtual Bread Board – это симулятор макетной платы Arduino для моделирования цифровых схем. Утилита позволяет писать код для платформы Arduino непосредственно в эмуляторе и одновременно проверить его работоспособность на компьютере без самого контроллера Arduino.

Схема подключения модуля GY-531 для получения системы ориентирования в пространстве представлена в табл.1. [4]

Таблица 1 – Схема подключения модуля GY-531

Выход/вход модуля GY-531	Выход/вход платформы ArduinoUno
VCC	3,3V
GND	GND
SCL	A5
SDA	A4
ADO	GND
INT	D2

Следует отметить, что в данном проекте использовался акселерометр и гироскоп одновременно, чтобы повысить точность полученных данных. Это обусловлено разной спецификой работы акселерометра и гироскопа, а также определяемых ими параметров.

Далее необходимо разработать систему для управления двигателем. В данной системе двигатель будет управлять той же платформой при помощи микросхемы драйвера двигателей L293D. Система управления двигателем (рис.3) моделировалась отдельно от модуля GY-531 при помощи макетной платы.

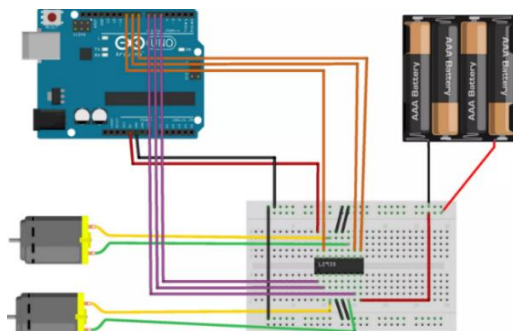


Рис.3 – Схема управления двигателями

Выводы. Выполнена разработка и моделирование электронного прибора, определяющего положение объекта в пространстве при помощи акселерометра и гироскопа. Устройство определяет углы подъёма или спуска и генерирует сигналы управления для корректировки работы двигателей. Полученная система ориентирования в пространстве простая в применении и дешёвая при сборке. Меняя микросхемы драйвера двигателей возможно управление данной системой электродвигателями любой мощности. Для программирования микроконтроллера используется язык C. Среда разработки транслирует программу на язык Assembler – понятный для микроконтроллера.

Список литературы

1. Платформа Arduino Uno [электронный ресурс] // Официальный сайт Ардуино в России: [сайт]. [2016]. URL: <http://arduino.ru/Hardware/ArduinoBoardUno>.
2. Официальный сайт Arduino [электронный ресурс]: [сайт]. URL: <http://arduino.cc>.
3. Подбельский В. В. Программирование на языке Си: Учеб. пособие. – 2-е доп. изд./ В. В. Подбельский, С. С. Фомин. – М.: Финансы и статистика, 2004. – 600 с., ил.
4. Arduino Playground [электронный ресурс] // Энциклопедия знаний по MPU-6050: [сайт]. [2016]. URL: <http://playground.arduino.cc/Main/MPU-6050>.

УДК 004.65

Исследование работы стандартного оптимизатора реляционной СУБД и применение основных методов лексической оптимизации запросов в рамках АСУ ДонНТУ.

Скиба В.Е., Краснокутский В.А., Меренкова Л.Л.

ГОУВПО Донецкий национальный технический университет
кафедра компьютерной инженерии
E-mail: skibaviktor@mail.ru

Аннотация

Скиба В.Е., Краснокутский В.А., Меренкова Л.Л. Исследование работы стандартного оптимизатора реляционной СУБД и применение основных методов лексической оптимизации запросов в рамках АСУ ДонНТУ. Изучена технология работы стандартного оптимизатора в реляционной СУБД на примере Microsoft SQL Server. Рассмотрены основные подходы в области оптимизации запросов, в частности, технология лексического преобразования для увеличения эффективности работы СУБД и уменьшения времени выполнения запроса.

Общая постановка задачи. Вопрос оптимизации запросов всегда является актуальным как для разработчиков баз данных, так и для программистов, которые с ними работают. В современных СУБД содержится компонент, называющийся оптимизатором, работу которого необходимо рассмотреть и выбрать возможные пути, как изменить запрос или БД таким образом, чтобы оптимизатор тратил меньше времени и ресурсов на построение плана выполнения запроса и его реализацию.

Существуют несколько различных подходов и методов оптимизации запросов, но первоочередным является лексическая оптимизация на уровне языка SQL с целью анализа текста запроса для выявления избыточности.

Исследование работы оптимизатора. Оптимизатор [1] – подкомпонент системы управления базой данных, который принимает входной аргумент в виде текста запроса и рассматривает возможные стратегии исполнения запроса, после чего выбирает из них наиболее эффективную. Выбранная стратегия и будет планом исполнения запроса. Оптимизатор принимает решение, учитывая такие факторы, как размер таблиц, из которых проводится выборка, существующие в таблицах индексы и всевозможные логические операторы (AND, OR, NOT), используемые в предложении WHERE.

Основной задачей оптимизатора является выборка наиболее эффективного плана выполнения запроса. Эта задача разбивается на 4 основных этапа, которые представлены на рисунке 1.

1) Синтаксический разбор (parsing). При выполнении синтаксического анализа, проверяется синтаксис запроса и запрос преобразовывается в дерево. После этого, выполняется проверка всех объектов базы данных, на которые ссылается запрос. (Например, проверка наличия в базе данных столбцов, указанных в запросе). После завершения работы создается окончательное дерево запроса.

2) Компиляция запроса (query compilation). На этом этапе дерево запроса компилируется оптимизатором.

3) Оптимизация запроса (query optimization). Оптимизатор в качестве входных параметров на данном этапе принимает скомпилированное дерево запроса и исследует различные стратегии обращения к данным, прежде чем решить, как обрабатывать данный запрос. Чтобы отыскать оптимальный план выполнения запроса, оптимизатор сначала выполняет анализ запроса, в процессе которого ищет аргументы поиска и операции соединения. Затем оптимизатор решает, какие индексы использовать. Индекс представляет собой структуру данных, которая позволяет получать быстрый доступ к одной или нескольким строкам данных. Компонент Database Engine, при поиске определенной строки из таблицы обращается к индексу, чтобы узнать ее физическое местонахождение. [1] Далее, если в запросе есть операции соединения, оптимизатор выбирает методы соединения и метод обработки.

4) Выполнение запроса (query execution). Созданный план выполнения сохраняется как основной и затем выполняется.

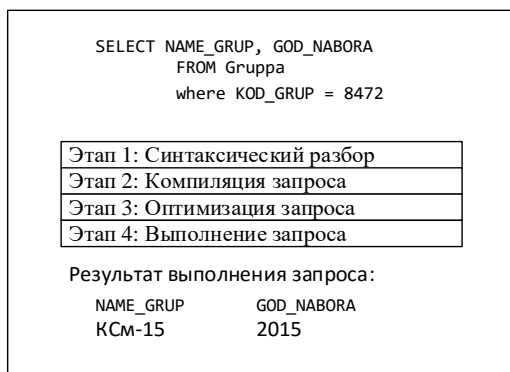


Рис. 1. Основные этапы обработки запроса.

Лексическая оптимизация запросов. Для выполнения лексической оптимизации запросов единственным источником информации является сам текст запроса как лексическая конструкция. Иные сведения о базе данных и ее структуре в анализе не используются. Процесс лексической оптимизации

включает в себя анализ ограничения запроса, сравнение содержащихся в нем условий с целью выявления избыточности.

Лексическая оптимизация может устранять неоптимальности запроса тремя различными способами:

- 1) Сокращение запроса - удаление избыточных условий, например, дублирующих одно другое или тождественно истинных, обработка которых является излишней с точки зрения выполнения запроса.
- 2) Усовершенствование – действие, при котором в некотором роде усложняется структура запроса, но это ведет к оптимизации его выполнения.
- 3) Преобразование запроса – изменение запроса, не относящееся ни к сокращениям, ни к усовершенствованиям запроса, и играющее роль, служебную по отношению к иным формам оптимизации запроса, например, стандартизация формы представления запроса.

Лексическая оптимизация путем сокращения запросов. Сокращение запросов является наиболее распространенным действием и предполагает собой повышение лаконичности и уменьшение условий выборки при сохранении семантической целостности. Если учесть то, что каждое условие в запросе требует определенных затрат ресурсов на обработку, то любые семантические повторы в ограничении выборки приводят к росту задержек на обработку запроса. Соответственно, устранение подобных случаев избыточности на стадии написания запроса до его компиляции и выполнения может существенным образом оптимизировать данный запрос.

Основные идеи лексической оптимизации путем сокращения запросов сводятся к следующим:

- Исключение избыточных операций;
- Упрощение выражений, использующих пустые отношения и тривиальные условия;
- Вынос общих выражений «за скобки».

Пример использования сокращения запросов путем использования реляционной алгебры для уменьшения логических выражений можно представить формулой[2]:

$$\begin{aligned} & ((a \text{ AND } b) \text{ AND } c \text{ OR } (((a \text{ AND } b) \text{ AND } (c \text{ AND } d)))) = > \\ & (a \text{ AND } b \text{ AND } c) \text{ OR } (a \text{ AND } b \text{ AND } c \text{ AND } d) \end{aligned} \quad (1)$$

Далее формулу (1) можно еще упростить получив в итоге:

$$(a \text{ AND } b \text{ AND } c) \quad (2)$$

В ходе исследования был проведен эксперимент, в котором в качестве условий выборки были представлены аналогичные формулам (1) и (2) логические условия. Оба запроса выполнялись на центральном сервере ДонНТУ. Для достоверности результатов, каждый запрос выполнялся по 5 раз, так как загруженность центрального сервера в разные моменты времени может отличаться и это может оказать существенное влияние на время выполнения. Условия запросов представлены на рисунке 2.

```

select student.GOD_NABORA, gruppа.NAME_GRUP,* from student
inner join gruppа on STUDENT.K_GRUP = Gruppа.KOD_GRUP
where ((student.K_FAK = 30 and PR_SOST = 217) and student.GOD_NABORA >= 2010 or
(((student.K_FAK = 30 and PR_SOST = 217) and (student.GOD_NABORA >= 2010 and student.GOD_NABORA < 2016))))

select student.GOD_NABORA, gruppа.NAME_GRUP,* from student
inner join gruppа on STUDENT.K_GRUP = Gruppа.KOD_GRUP
where student.K_FAK = 30 and PR_SOST = 217 and student.GOD_NABORA >= 2010

```

Рис. 2. Запрос до и после применения лексической оптимизации путем сокращения.

При выполнении двух данных запросов, имеется одинаковый набор данных, что свидетельствует о правильном применении законов реляционной алгебры. Временные параметры, которые необходимы для определения эффективности оптимизации сведены в таблицу 1. Результаты получены средствами отладчика запросов MS SQL Management Studio.

Таблица 1 – Результаты выполнения запроса до и после оптимизации.

Шаг Запрос	№1, мс	№2, мс	№3, мс	№4, мс	№5, мс
До оптимизации	112	123	133	138	128
После оптимизации	113	115	109	114	104

Из результатов работы определено, что среднее время выполнения неоптимизированного запроса составляет 126.8мс, в то время как время выполнения оптимизированного запроса составило 111мс. Можно сделать вывод, что трансформация формулы (1) в (2) привела к уменьшению времени выполнения запроса на 12.5%.

Лексическая оптимизация путем усовершенствования запросов. Технология усовершенствования запросов базируется на усложнении структуры запроса, включении в него новых временных таблиц, использование которых позволяет сократить расходы на обработку исходного запроса. Большинство алгоритмов, решающих эту задачу, относятся к технологии «магических множеств» [2].

Алгоритм магических множеств относится к числу алгоритмов лексической оптимизации запроса. Он означает перезапись запроса,

исключающую генерацию нерелевантных связей путем вычисления дополнительных таблиц, которые содержат связывания, используемые для ограничения таблицы, и действуют как фильтры.

Перезапись исходного запроса с использованием этих фильтров включает следующие основные этапы:

- анализ и поиск неоптимальных связей в рамках реализации запроса;
- создание дополнительных (магических) таблиц;
- интеграцию магических таблиц в уже существующее ограничение и изменение описания существующих условий.

Лексическая оптимизация путем преобразования запросов.

Критерием классификации алгоритма как относящегося к числу алгоритмов простого преобразования является его функциональное предназначение[2]. Если итогом работы некоторого алгоритма должно стать обретение запросом более оптимальной формы с точки зрения того или иного критерия (например, числа условий в ограничении), то алгоритм может быть отнесен к числу алгоритмов сокращения или усовершенствования. Если же алгоритм, преобразует запрос к иному виду, к примеру, меняя порядок операций, то в данном случае речь идет о простом преобразовании запроса.

К технологиям преобразования нужно отнести алгоритмы стандартизации запросов, приведения их к универсальной стандартной форме в целях дальнейшей обработки. Наиболее распространенным видом преобразования запросов является перенос логических условий вниз по дереву запроса. Важным аспектом является перемещение селективных операций ниже конструктивных операций, таких, как соединение и декартово произведение, чтобы можно было выполнить селективные операции как можно раньше, для меньших ресурсозатрат.

Выводы. При выборке многосвязных данных большого объема в реляционных базах данных, вопрос оптимизации запросов является весьма актуальным, так как при большой загрузке сервера время выполнения запросов и затраты ресурсов могут играть большую роль.

При исследовании стандартного оптимизатора на примере Microsoft SQL Server были выявлены основные особенности и принципы его работы. Также рассмотрены основные методы лексической оптимизации запросов, проведен эксперимент оптимизации путем сокращения, который показал уменьшение времени выполнения запроса на 12,5%.

Список литературы

1. Microsoft SQL Server 2012. Руководство для начинающих: Пер. с англ. – СПб.:БХВ-Петербург, 2013, - 816 с.: ил.
2. Обзор развития методов лексической оптимизации запросов. Труды Института системного программирования РАН Том 23. 2012 г. Стр. 195-214.

УДК 004.383.3

**Разработка и исследование алгоритмов управления модуля
энергосбережением системы "умный дом"**

Каховский Д.А., Завадская Т.В.

ГОУВПО "Донецкий национальный технический университет" кафедра
компьютерной инженерии

E-mail: Harbinger1994d@yandex.ru

Аннотация

Каховский Д.А., Завадская Т.В. Разработка и исследование алгоритмов управления модуля энергосбережением системы "умный дом". Спроектирована аппаратная часть модуля энергосбережения системы "умный дом". Разработаны алгоритмы анализа энергетического потребления помещения и его климатических показателей в реальном времени.

Общая постановка проблемы. В современной тенденции перехода на альтернативные источники электроэнергии, а так же, учитывая высокую стоимость энергетического сырья, появилась проблема неэффективного использования электроприборов и отопительных систем. В связи с высокой стоимостью и труднодоступностью фирменных систем автоматического управления электропитанием жилых помещений, появилась задача разработки относительно дешевой системы "умный дом", с возможностью дистанционного управления и мониторингом состояния в режиме реального времени. Важная цель работы - реализовать систему "умный дом" с минимальными материальными затратами.

Исследования

Проектирование системы можно разделить на 2 части:

- Аппаратную.
- Программную.

Аппаратная часть системы будет представлена платформой Arduino (рис.1) с расширениями и физическими модулями. Arduino – это инструмент для проектирования электронных устройств, более тесно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры. В данной работе будет использована версия платформы Arduino UNO – самая популярная версия базовой платформы Arduino с USB-интерфейсом и возможностью подключения большого разнообразия плат расширения. Эта платформа предназначена для физических расчетов (physical computing) с открытым программным кодом, построенная на простой печатной плате с современной средой для разработки программного обеспечения [1].



Рис. 1 - Внешний вид платформы Arduino UNO

Arduino использует микроконтроллер Atmega328, который имеет 32 Кб флеш памяти. Этого будет вполне достаточно для выполнения возложенной на платформу задачи. Остальная обработка будет распределена на веб-ресурс.

Микроконтроллеры Arduino отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов и соединяется с компьютером через USB-интерфейс [2].

Для осуществления дистанционного управления устройством, необходимо подключить устройство к сети Интернет. В ходе проектирования будет использована плата расширения Arduino Ethernet.

Так как система должна быть универсальна и интуитивно понятна любой группе пользователей, то необходимо использовать неинвазивный метод подключения системы к электросети помещения. Для определения общего энергопотребления помещения будет использован датчик переменного тока CST-013-000 100А (рис.2) с внешним подключением (монтаж происходит поверх силового провода). Такие же датчики будут использованы для определения работы электроприборов.



Рис. 2 - Внешний вид датчика переменного тока CST-013-000

Управление приборами осуществляется с использованием плат реле Arduino Relay 10A, с напряжением питания 5В, что соответствует напряжению питания используемого микроконтроллера. На реле будут поступать управляющие сигналы по средствам радиоканала на частоте 433 МГц. Такой метод подключения позволяет избавиться от использования схем согласования, организовать полностью дистанционное управление, а так же упрощает начальную установку системы в любое помещение.

Набор датчиков для климатического мониторинга помещения подбирается индивидуально, в зависимости от поставленной задачи и параметров помещения, но стандартно в системе используются датчики температуры.

В целях автоматизации системы, а так же что бы предотвратить человеческий фактор – система должна отслеживать наличие людей в помещении. В жилых домах эта функция будет реализована проверкой наличия ключа в замочной скважине входной двери. При отсутствии ключа в течении 30 минут, система отключит все включенные приборы освещения.

В итоге на аппаратную часть возложена задача опроса всех типов датчиков, передача управляющих сигналов по радиоканалу, а так же отправка полученных данных на веб-ресурс, с целью мониторинга и дальнейшей обработки.

Программная часть системы представлена в программном коде микроконтроллера (Arduino скетч) и веб-ресурсе, который позволит дистанционно управлять системой "умный дом" и проводить анализ всех показателей системы.

Веб-ресурс должен иметь систему защиты. Для этого необходимо установить модуль авторизации пользователя. Это позволит избежать злоумышленного вмешательства в важные данные веб-ресурса. Схема обмена информацией системы с веб-ресурсом представлена на рисунке 3.

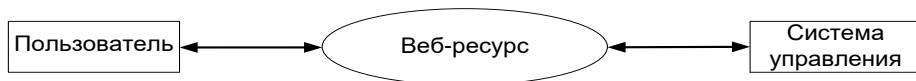


Рис.3 - Схема обмена данными

В работе будут использованы инструменты СУБД phpMyAdmin. Они позволят удобно передавать данные между объектами системы, защитить данные при помощи авторизации и хранить все показатели в базе данных веб-ресурса, с целью их дальнейшего использования. Остальные обработки данных будут реализованы с помощью PHP и JavaScript.

Алгоритм будет анализировать климатические показатели помещения, автоматически регулируя отопительные приборы или же рекомендовать пользователю дистанционно провести изменения в системе, а так же предотвращать использование осветительных приборов, если в помещении нет людей.

Важное значение будет иметь интерфейс пользователя. Он должен быть максимально интуитивно понятен и прост для обычного пользователя. Сайт управления системой будет доступен как на стационарных ПК, так и на любых видах портативных устройств.

Для удобного представления информации - будут использованы таблицы показателей датчиков, дневные диаграммы температуры и т.д.

Выводы. Разрабатываемая система управления исключит вероятность того, что неиспользуемые в данный момент приборы будут потреблять электроэнергию. Это позволит улучшить энергосбережение используемого помещения и получить экономическую выгоду. А так же даст возможность из любой точки мира осуществлять мониторинг состояния помещения, такие как температура помещения, общее электропотребление, и наличие активных электроприборов.

Список литературы

1. Официальный сайт Ардуино/ Интернет-ресурс. – Режим доступа: <http://arduino.ru/About>
2. Википедия - свободная энциклопедия, статья "Arduino"/ Интернет-ресурс.– Режим доступа: <https://ru.wikipedia.org/wiki/Arduino>
3. Сообщество "Arduino, AVR, и просто интересные проекты" Интернет-ресурс.– Режим доступа: <http://devictor.blogspot.com/>

УДК 004.414

Разработка базовой модели квадрокоптера на базе FPGA Мохначёв В.В., Зинченко Ю.Е., Коваль А.В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
E-mail: vlad230596@mail.ru

Аннотация

Мохначёв В.В., Зинченко Ю.Е., Коваль А.В. Разработка базовой модели квадрокоптера на базе FPGA. Выполнен анализ возможности изготовления квадрокоптера. Был дан обзор его ключевых блоков. Предложена конфигурация построения базовой модели и система для её тестирования.

Ключевые слова: квадрокоптер, FPGA, микроконтроллер, STM32, визуализация данных, Qt.

Постановка проблемы. С каждым днём в новостных источниках появляется всё больше и больше информации о мультикоптерах, а, начиная с 2015

года, обладателей таких устройств обязали проходить регистрацию в США и России. Всё это свидетельствует о стремительном росте популярности дронов. На сегодняшний день они уже нашли своё применение в различных сферах жизни человека, и число новых областей их применений растёт очень быстро. На сегодняшний день это одно из самых актуальных направлений робототехники, о чём свидетельствуют международные технические форумы и выставки[1].

Таким образом, возникает необходимость в наличии устройств такого класса для современного общества. Готовые устройства весьма дорогие, что заставляет задуматься о возможности самостоятельного изготовления квадрокоптера. Поскольку это сложное устройство, в ходе исследования было принято решение разбить процесс изготовления на несколько итерационных этапов.

На первом этапе стоит задача выбора наиболее оптимальных компонентов для квадрокоптера, а также изготовление и тестирование базовой модели и в первую очередь важнейшего компонента – система позиционирования и система передачи данных.

Целью данной статьи является проведение анализа ключевых блоков конструкции квадрокоптера, выбор оптимальных составляющих для базовой модели, и разработка архитектуры средств тестирования и отладки (верификации).

Общая конструкция квадрокоптера. Обобщённая структура квадрокоптера [3] приведена на рисунке 1. Она включает в себя основные элементы, которые обеспечивают работоспособность устройства и дополнительные, которые расширяет его функционал. К основным элементам можно отнести схему управления, аккумулятор, датчик положения (гироскоп), контроллеры моторов, моторы и радиопередатчик. К дополнительным элементам обычно относят датчик GPS, датчик высоты, камеру и видеопередатчик.

Схема управления является главным элементом системы, она принимает команды пользователя, а также опрашивает датчики и посылает сигналы контроллерам моторов, которые непосредственно управляют мощными моторами. Видео-поток с камеры поступает в управляющий элемент и после минимальной обработки транслируется через передающее устройство пользователю. Также осуществляется запись данного видео на флеш-память. Аккумулятор обеспечивает питанием все модули устройства.

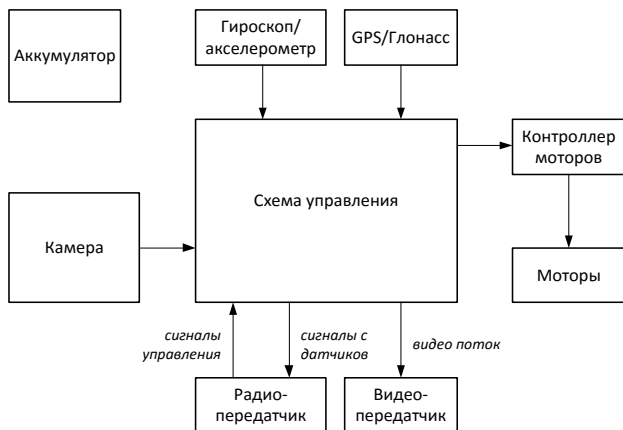


Рис. 1 – Общая структура квадрокоптера

Выбор оптимальной конфигурации. Под оптимальной конфигурацией понимается набор компонентов, которые должны иметь минимальную цену и соответствовать заявленным требованиям.

Рама. Корпус квадрокоптера является весьма важным элементом, к нему предъявляются следующие требования: лёгкость, прочность и удобство монтажа элементов. На рисунке 2 представлен один из самых простых, но эффективных вариантов рамы. Его и предлагается использовать. Конструкция состоит из двух перпендикулярных алюминиевых лучей, которые крепятся на пластине. Данная пластина может быть изготовлена из разных материалов: начиная от фанеры, заканчивая углепластиком. Но наиболее доступным является текстолит, имея при этом хорошие свойства (высокая прочность и лёгкость), поэтому именно он используется в базовой версии квадрокоптера.

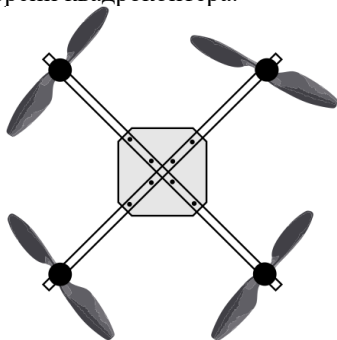


Рис. 2 – Рама квадрокоптера

Ядро. Ранее [4] было доказано, что для построения ядра квадрокоптера наиболее лучшим образом подходит FPGA. Для тестирования и отладки используется отладочная FPGA-плата семейства Altera Cyclone II [2]. Проектирование квадрокоптера предполагается осуществлять на базе FPGA-лаборатории ДонНТУ [5, 6].

Моторы и пропеллеры. При выборе моторов важнейшей характеристикой является скорость вращения и создаваемая тяга. Для небольших квадрокоптеров нужна высокая скорость вращения винтов и низкая тяга, в то время как для аппаратов, предназначенных для перевозки грузов, наоборот. Для базовой версии будем использовать моторы с 1000 kv (оборотов в минуту на один вольт) и пропеллеры типа 1045 – что является компромиссом между максимальной скоростью движения и грузоподъемностью.

Контроллеры моторов. Исходя из максимальных нагрузок (10-12А) с запасом рекомендуется использовать рассчитанные на 30А контроллеры. Помимо того они способны отдавать 2А постоянного тока для питания системы контроля.

Аккумулятор. Согласно спецификации, которую предоставляет производитель моторов, необходимое напряжение питания 11.1 В. В качестве аккумулятора решено использовать несколько батарей (каждая по 1500 mAh) взамен одного аккумулятора повышенной ёмкости. Преимущество данного подхода в том, что увеличивается надёжность системы (неисправность одной из батарей не выводит систему из рабочего состояния, а лишь снижает полётное время), и это при том, что цена, размер и масса аналогичных конструкций (один большой аккумулятор или несколько меньших) одинаковы.

Радиопередатчик. В качестве передатчика команд и состояния системы решено было использовать SI4432. Это радиопередатчик, который работает на частоте 433 МГц и имеет заявленную дальностью работы в 1 Км. Помимо этого модуль имеет размер 2см*1.5см и вес несколько грамм, что является несомненным преимуществом для его использования на летательном аппарате.

Контроллер управления. Управление в дальнейшем должно осуществляться с джойстика, привычного для многих людей. Также явным преимуществом будет возможность попробовать управлять виртуальной моделью квадрокоптера, не подвергая опасности дорогостоящий аппарат. Для взаимодействия джойстика, радиопередатчика и компьютера решено использовать микроконтроллер архитектуры ARM класса STM32 как наиболее современный и мощный контроллер за сравнительно небольшую стоимость.

Тестирование устройства. Для тестирования ставятся следующие задачи: получение данных с датчика положения (гироскопа), передача их контроллеру управления (на базе STM32), связь микроконтроллера и компьютера по USB, визуализация данных в 3D-модели. Предлагаемая система отображена на рисунке 3.

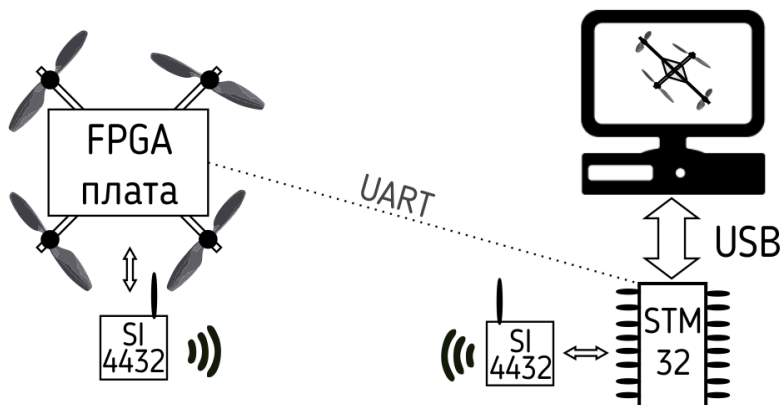


Рис. 3 – Система тестирования базовой модели

В качестве начального этапа предполагается использование UART в качестве канала передачи данных между квадрокоптером и управляющим контроллером. В дальнейшем планируется переход на радиопередатчики типа SI4432. Для визуализации данных была выбрана среда Qt. Данный выбор обусловлен хорошей поддержкой как работы с 3D-графикой, так и работы с периферийными устройствами. Помимо того, Qt имеет очень широкий набор библиотек и развитое сообщество, это обеспечивает более быструю и лёгкую работу со средой.

Выводы. Произведён анализ ключевых блоков конструкции квадрокоптера, выбрана оптимальная конфигурация составляющих для базовой модели, а также предложена архитектуры среды для её тестирования. Результаты анализа показали, что существует возможность самостоятельно изготовить квадрокоптер, не прибегая к покупке дорогостоящего готового аппарата. В работе была представлена лишь базовая модель квадрокоптера, которая имеет очень много векторов развития, разработка планируется в ближайшем будущем.

Список литературы

1. 7 лучших дронов на CES 2016 [электронный ресурс]. URL: <https://geektimes.ru/company/robhunter/blog/269478/>
2. Ресурс Altera. [электронный ресурс]. - Режим доступа : URL <https://www.altera.com/products/fpga/cyclone-series/cyclone-ii/support.html>
3. Принцип работы квадрокоптера. [электронный ресурс]. - Режим доступа: URL <http://quatrocopter.com/printsip-raboty-i-kvadrokoptera>
4. В.В. Мохначёв, Ю.Е. Зинченко, А.В. Коваль. Анализ возможности разработки квадрокоптера на базе FPGA // Сборник трудов V ежд. научн-техн.

конф. студентов, аспирантов и молодых ученых «Компьютерная и программная инженерия – 2015». – Донецк: ДонНТУ. – 2015. – С 4.

5. Зинченко Ю., Калашников В., Хайдук С., и др. FPGA-технологии проектирования и диагностика компьютерных систем в ДонНТУ / Сборник научных трудов VI Междунар. научн.-практ. конф. «Современные информационные технологии и ИТ-образование». - Москва: МГУ, 2011. - Т. 1. 787 С. 422-429

6. Зинченко Ю.Е., Калашников В.И., Дяченко О.Н., и др.. Современные проекты FPGA-лаборатории ДонНТУ / Материалы международной научно-практической конференции «Инновационные перспективы Донбасса», том 5. «Компьютерные науки и технологии, г. Донецк, 20-22 мая 2015 г. – Донецк: ДонНТУ, 2015. С 60-66

УДК 004.2

Система приема посетителей жилого дома

Бурлака М.В., Теплинский С.В.

Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии
maria.burlaka@bk.ru

Аннотация

Бурлака М.В., Теплинский С.В. Система приема посетителей жилого дома. *Выполнен анализ платформы Arduino, определены основные характеристики ее составляющих. Произведен выбор аппаратных средств и разработана система приема посетителей жилого дома на основе плат Arduino.*

Ключевые слова: *микроконтроллер, плата Arduino Uno, плата расширитель Ethernet Shield, модуль камеры OV7670.*

Постановка проблемы. В последнее время мы можем наблюдать, как набирают популярность различные системы для создания «умного дома». Но в большинстве, если использовать готовые системы, то их стоимость измеряется в тысячах долларов, что не делает «умный дом» доступным любому желающему. В связи с этим, возникла идея проекта приема посетителей жилого дома на основе доступной и сравнительно не дорогой платформы Arduino. Для реализации данного проекта необходимо выполнить следующее:

- проанализировать составляющие платформы Arduino;
- выбрать оптимальные аппаратные средства для системы приема посетителей жилого дома;
- представить структуру системы в целом и реализовать ее.

Цель статьи – провести анализ платформы Arduino, разработать на основе данной платформы систему приема посетителей жилого дома, определить аппаратные средства наиболее подходящие для данной системы.

Анализ платформы Arduino. Arduino – это открытая программируемая аппаратная платформа для работы с различными объектами с помощью компонентов, расширяющих возможности платы. Проекты Arduino могут быть как самостоятельными, так и взаимодействовать с программным обеспечением, работающем на персональном компьютере, телефоне с ОС Android или через Internet. Любую плату Arduino можно собрать вручную или же купить готовое устройство. Программное обеспечение является кроссплатформенным, в то время как большинство подобных систем ориентировано на работу только в Windows.

Arduino упрощает процесс работы с микроконтроллерами и предоставляет широкие возможности для изучения микроконтроллеров и проектирования различных управляющих систем на базе большого набора недорогих модулей (плат). Программное обеспечение Arduino имеет большое количество библиотек, которые упрощают написание программного кода для устройства. Следует отметить, что программное обеспечение Arduino имеет открытый исходный код, благодаря этому опытные программисты могут изменять и дополнять его.

Существует несколько разновидностей плат Arduino. Основные типы: Arduino Nano, Arduino Uno (Standart), Arduino Mini, Arduino Leonardo, Arduino Pro, Arduino Pro Mini. Так же существуют платы расширители. Например, некоторые из них: Ethernet, GSM, TFT LCD-экран, Wireless, Wi-Fi.

Программирование Arduino. Программы для Arduino пишутся на обычном C/C++, дополненным простыми и понятными функциями для управления вводом/выводом на контактах. Для удобства работы с Arduino существует бесплатная официальная среда программирования «Arduino IDE»[2], которая не требует установки дополнительного программного обеспечения. Это упрощает работу пользователю с данной платформой. С помощью неё загрузка новой программы в Arduino становится очень быстрой, для этого необходимо подключить плату к компьютеру через USB. Так же возможна работа и через Visual Studio, Eclipse, другие IDE или командную строку, но для этого требуется использование дополнительных плагинов. Для тестирования программ можно использовать программу VBB 4.2.9 (Virtual Breadboard), которая эмулирует работу с платами Arduino.

Выбор аппаратных средств для реализации системы приема посетителей жилого дома. Для оптимальной работы системы и ее правильной функциональности необходимо выбрать аппаратные средства, которые будут соответствовать поставленной задаче. Для системы приема посетителей жилого дома были выбраны следующие аппаратные средства:

- Плата Arduino Uno
- Модуль камеры OV7670
- Плата расширитель Ethernet Shield W5100
- Четыре резистора номиналом 4.7 кОм и 10 кОм

Данные элементы были выбраны по ряду причин: низкая себестоимость, доступность и удобство в эксплуатации. Рассмотрим более детально каждый элемент.

Основой системы приема посетителей является плата Arduino Uno, которая представлена на рис. 1.

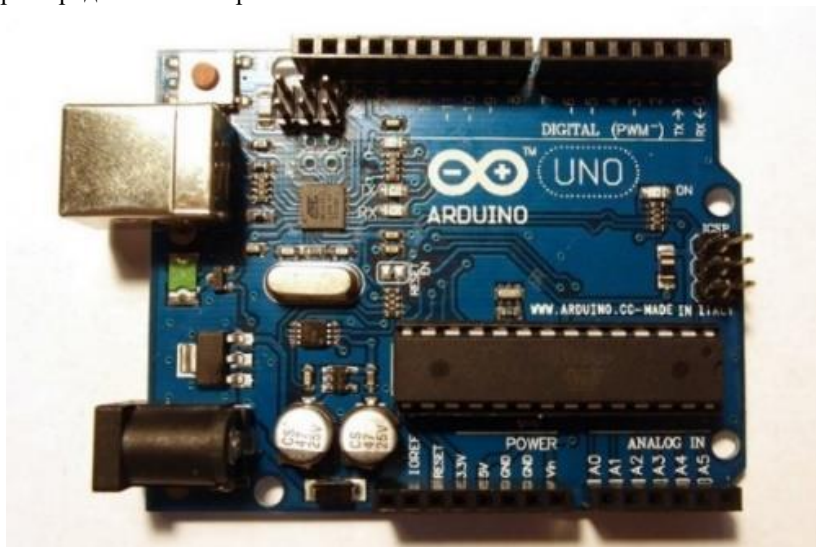


Рис. 11 – плата Arduino Uno

Эта плата была выбрана из множества других так как является эталонной моделью платформы Arduino. Ее так же называют стандартной. Рассмотрим ее краткие характеристики, которые представлены в табл. 1.

Таблица 1 – Краткие характеристики Arduino Uno

Название	Описание
Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (ATmega328)
EEPROM	1 Кб (ATmega328)
Тактовая частота	16 МГц

Arduino Uno – это устройство на основе микроконтроллера ATmega328. ATmega328 оснащен 32 Кб флеш-памятью (память программ), 0,5 Кб из которых отведено под загрузчик. Эта память постоянна и не предназначена для изменения по ходу работы устройства. Её предназначение — хранение программы и сопутствующих статичных ресурсов. Также имеется 2 Кб SRAM-памяти (память данных), которые используются для хранения временных данных вроде переменных программы. SRAM-память очищается при обесточивании. Ещё имеется 1 Кб EEPROM-памяти для долговременного хранения данных[1].

Arduino Uno может питаться через USB или от внешнего источника питания, таких как блок питания или аккумуляторная батарея.

Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, тогда платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Далее рассмотрим плату расширитель Ethernet Shield W5100, которая изображена на рис. 2.

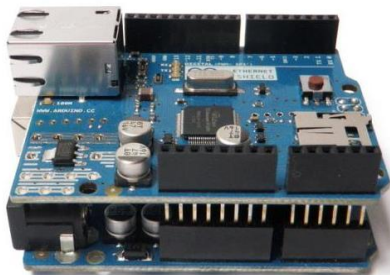


Рис. 12 – Плата расширения Ethernet Shield и Arduino Uno

Плата расширитель Ethernet позволяет устройству на базе Arduino получить доступ к сети Интернет. Это дает возможность управлять устройством дистанционно через интернет, получать информацию от различных онлайн-сервисов и обрабатывать ее, а также запустить небольшой веб-сервер.

Для подключения платы расширения к Arduino предусмотрен специальный разъем[3], представляющий собой металлические выводы ("папа") с одной стороны платы и гнезда ("мама") – с другой стороны. Такая конструкция позволяет подключить к Arduino сразу несколько плат расширения, разместив их одну над другой. Для соединения с сетью используется стандартный разъём RJ45. На плате также имеется слот для microSD-карты объемом до 2 Гб для хранения файлов и организации сетевого доступа к ним. В частности, для системы приема посетителей на microSD возможно хранение снимков с камеры.

Следующим рассмотрим модуль камеры OV7670, изображенный на рис.3.



Рис. 13 – Модуль камеры OV7670

Модуль камеры OV7670 формирует изображение с максимальным разрешением 640 x 480 и может выдать его со скоростью до 30 кадров в секунду. Видеоинтерфейс использует синхроимпульсы по кадрам VSYNC, по строкам HREF и по пикселям PCLK. Данные пикселя, представляющие собой закодированную информацию об его цвете, выдаются по 8-ми разрядному параллельному интерфейсу D7-D0 по тактам PCLK. Один пиксель не равен

одному байту. В случае кодировки RGB данные о цвете пикселя выдаются в двух байтах. Для OV7670 существуют множество настроек.

Данная камера была выбрана из соотношения цена-качество. Существуют и другие камеры для работы с платформой Arduino, которые по каким-то параметрам лучше, а по каким-то хуже выбранной OV7670.

Структура системы приема посетителей жилого дома. Данная система необходима для того, чтобы контролировать людей, которые нажимают кнопку дверного звонка. Принцип действия заключается в следующем: человек нажимает кнопку дверного звонка, после чего сигнал нажатия кнопки подается на плату Arduino Uno, а затем она передает сигнал камере, которая делает снимок посетителя. Таким образом, хозяин дома может увидеть кто пришел к нему, не открывая дверь и даже не подходя к ней. Особенно данная система удобна для частных домов. Схема подключения модуля камеры к плате Arduino Uno изображена на рис. 4.

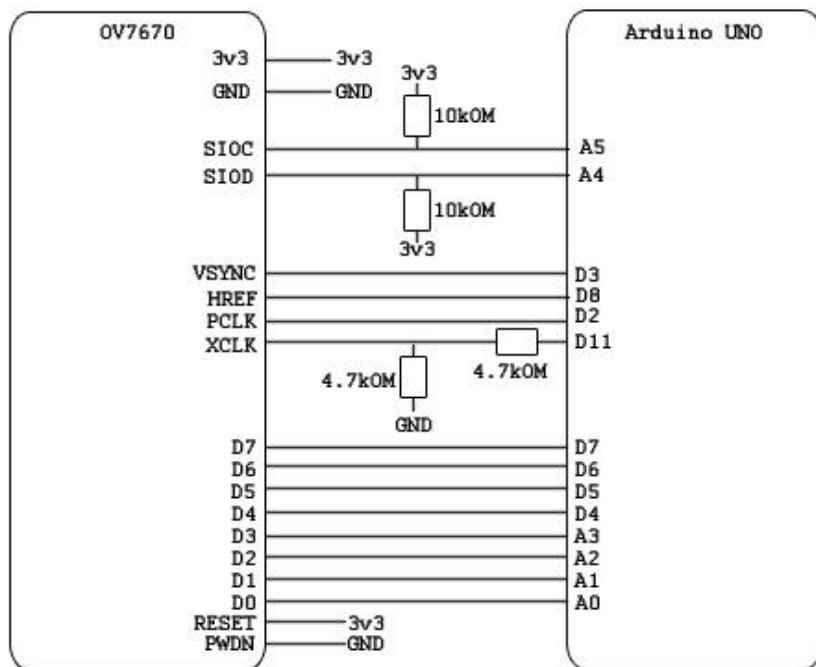


Рис. 14 – Схема подключения камеры OV7670 к плате Arduino Uno



Рис. 15 – Фотография снятая с модуля камеры OV7670

После подключения камеры к Arduino необходимо выполнить ряд настроек для получения изображения на устройстве. На данном этапе разработки проекта удалось получить черно белое изображение, но модуль этой камеры позволяет в конечном итоге получить цветное изображение с приемлемым качеством. Полученное изображение представлено на рис. 5.

В дальнейшем планируется разработать оповещение о приходе гостей на электронную почту хозяина дома. Для этого понадобится выше описанная плата расширитель Ethernet.

Выводы. Проведен анализ плат платформы Arduino и реализована система приема посетителей с применением необходимых компонентов и плат, которые оптимально подходят для данной системы по ряду причин. Изучены и применены на практике основы программирования под Arduino в среде разработки «Arduino IDE». На практике проведена проверка работоспособности данной системы на текущем этапе разработки.

Список литературы

1. Амперка [электронный ресурс] // Амперка: [сайт]. [2010-2016]. URL: <http://amperka.ru>
2. У. Соммер. «Программирование микроконтроллерных плат Arduino/Freeduino». БХВ-Петербург, 2012 г. – 256 с.
3. Паяльник. [электронный ресурс] // Сайт Паяльник: [сайт]. [1999-2016]. URL: <http://cxem.net>

УДК 004.457

Особенности восстановления удаленных файлов и каталогов в файловой системе FAT32

Костенко Д.А. (4 курс, каф. КИ), Теплинский С.В., к.т.н., доц.
Донецкий национальный технический университет, г. Донецк
кафедра компьютерной инженерии

Аннотация

Костенко Д.А., Теплинский С.В. Особенности восстановления удаленных файлов и каталогов в файловой системе FAT32. Выполнен анализ методики восстановления файлов и каталогов в файловой системе FAT32. Определены основные особенности восстановления файлов и каталогов.

Ключевые слова: файловая система FAT32, восстановление файлов.

Постановка проблемы. Потеря важной информации – неприятное событие, которое может повлечь как финансовые потери, так и потерю личной информации. Такое случается довольно часто: случайно удалили папку, отформатировали съемный или логический диск, компьютер заразился вирусом и так далее.

На данный момент проблема восстановления удаленных данных в файловой системе FAT32 решена. Однако она не перестает быть интересной для изучения. Создано множество различных программ восстановления удаленных файлов. Разработчики данных утилит используют различные подходы к решению проблемы восстановления. Соответственно, и результаты их работы отличаются.

Цель статьи – провести анализ методики восстановления файлов и каталогов в файловой системе FAT32, используемой в разработанной программе. Доказать ее работоспособность и эффективность.

Особенности восстановления файлов. При удалении файла, соответствующая ему запись в файловой системе помечается как удалённая, а занимаемые им кластеры, соответственно, получают статус свободных, цепочка кластеров теряется [2]. Кластеры удаленного файла содержат его данные, до тех пор, пока они не будут перезаписаны данными другого файла. Данные самой файловой записи также не изменяются до перезаписи. На рисунке 1 приведен результат удаления файла.

Просто восстановить всю цепочку кластеров, принадлежащих удалённому файлу невозможно. Но можно попытаться восстановить содержимое файла, прочитав данные из известного начального кластера. Если файл небольшой и его содержимое умещается в единственном кластере – прекрасно. Если – нет, то можно попробовать определить, какие именно кластеры принадлежали конкретному файлу. Можно просто последовательно скопировать

объем данных, соответствующий размеру файла, игнорируя состояние выделения кластеров [3]. Однако разумнее считать данные только из свободных кластеров, так как это дает возможность восстановить даже некоторые фрагментированные файлы.

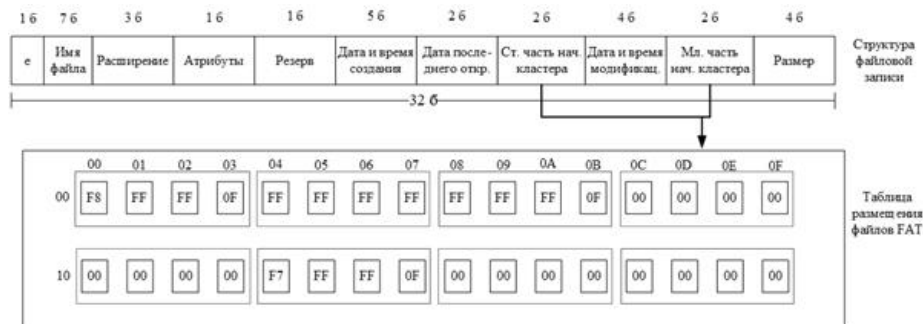


Рис. 1 – Результат удаления файла

В разработанной программе восстановление файлов осуществляется посредством анализа информации о файлах и папках – это самый первый метод, используемый в программах восстановления данных, так как при его успешном применении восстанавливаются файлы с оригинальными именами, путями, отметками даты/времени, и сами данные [1].

Пользователю предоставляется возможность выбора диска, подлежащего сканированию. Все удаленные файлы, файловые записи которых не были перезаписаны выводятся в список. Пользователь может выбрать файлы, которые необходимо восстановить. Также, предоставлена возможность выбора диска для сохранения восстановленных файлов (диски с файловой системой FAT32). По нажатию кнопки «Восстановить файлы» программа по очереди сканирует каждую файловую запись, определяя начальный кластер его содержимого. Так как размер кластера диска, содержащего удаленный файл, может не совпадать с размером кластера диска, на который файл восстанавливается, то размер считываемого кластера всегда равен размеру кластера диска, на который восстанавливаются файлы. Это позволяет корректно восстановить содержимое файла на другой носитель. Далее на диске, на который восстанавливаются файлы, создается файл с именем, указанным в структуре файловой записи. В кластер, относящийся к вновь созданному файлу, записывается информация из кластера восстанавливаемого файла. Если размер файла больше размера кластера, то данная операция выполняется в цикле необходимое количество раз. При чем, не зная цепочки кластеров удаленного файла, поиск «следующего» файла в цепочке осуществляется посредством поиска ближайшего свободного кластера. Одновременно формируется цепочка кластеров для восстановленного файла,

которая записывается в обе копии таблицы FAT. Заключительным шагом является изменение файловой записи восстановленного файла, в соответствии с параметрами удаленного файла (размер, дата и время создания и последнего изменения, атрибуты файла).

Данная методика позволяет восстанавливать не фрагментированные и слабо фрагментированные файлы любых размеров.

Особенности восстановления каталогов. Каждый каталог хранит файловые записи, содержащихся в нем файлов и каталогов. Для корректного восстановления иерархии каталога в разработанной программе предусмотрен механизм запоминания пути, по которому был расположен каждый из вложенных элементов каталога.

При отметке каталога на восстановление, осуществляется проход по всему содержимому этого каталога и каждый элемент выбранного каталога добавляется в список восстановления. При этом, если программа обнаруживает вложенный каталог, то и его содержимое обрабатывается таким способом. В результате, при восстановлении на выбранном диске создаются файлы и каталоги по указанному пути, восстанавливая иерархию. Также, организована возможность восстановления одного или нескольких определенных файлов в папке. В таком случае их путь не запоминается, и они будут восстановлены в корне выбранного диска.

Вывод. В результате проделанной работы разработана программа восстановления удаленных файлов в файловой системе FAT32. Программа восстанавливает файлы любых размеров и форматов. Предусмотрено восстановление полной иерархии каталога и отдельных его элементов. Программа разработана на языке C# с использованием функций WinAPI. Результаты работы программы совпадают с результатами работы утилиты R-Studio.

Список литературы

- 1.Харт, Джонсон, М. Системное программирование в среде Win32, 2-е изд. : Пер. с англ. : - М. : Издательский дом «Вильямс»,2001. – 464 с. : ил. – Парал. тит. англ.
2. Гордеев А.В, Молчанов А.Ю. Системное программное обеспечение. – СПб.: Питер, 2002. – 736 с.: ил.
3. Сенкевич Г. Е. Искусство восстановления данных. — СПб.: БХВ-Петербург, 2011. – 304 с.: ил. – (Аппаратные средства)

УДК 004.42

Исследование оптимальных запросов при выборке многосвязных данных больших объемов в реляционных базах данных.

Иванов В.Ю., Завадская Т. В., Меренкова Л.Л.

ГОУВПО "Донецкий национальный технический университет"

кафедра компьютерной инженерии.

E-mail: navi.rakstyle@gmail.com

Аннотация

Иванов В.Ю., Завадская Т.В., Меренкова Л.Л. Исследование оптимальных запросов при выборке многосвязных данных больших объемов в реляционных базах данных. Рассмотрен ряд вопросов, связанных с оптимизацией выполнения запросов в реляционных системах управления базами данных (СУБД). Рассмотрена работа оптимизатора СУБД, а так же два типа оптимизации - оптимизация по синтаксису и по затратам на языке SQL.

Общая постановка проблемы. Рассматривая оптимизацию в реляционных СУБД, имеют в виду аспект оптимизации запросов, то есть такой способ выполнения запросов, когда по начальному представлению запроса путем его синтаксических и семантических преобразований вырабатывается процедурный план выполнения, наиболее оптимальный при существующих в базе данных управляющих структурах. Соответствующие преобразования начального представления запроса выполняются специальным компонентом СУБД – оптимизатором, и оптимальность производимого им плана запроса может носить достаточно условный характер. План оптимален в соответствии с критериями, заложенными в оптимизатор, но при этом возможны отклонения от реальной оптимальности.

Несмотря на то, что оптимизаторы запросов современных реляционных СУБД различаются по сложности и принципам создания, все они следуют одним и тем основным этапам в выполнении оптимизации запроса, представленных на рис.1.



Рис.1. Последовательность выполнения запросов в реляционных СУБД

На *первой фазе* запрос, заданный на языке SQL:

- подвергается лексическому и синтаксическому анализу (оптимизатор разбивает запрос на его синтаксические компоненты, проверяет ошибки в синтаксисе);

- вырабатывается его внутреннее представление, отражающее структуру запроса и содержащее информацию, которая характеризует объекты базы данных, упомянутые в запросе (отношения, поля и константы).

На *второй фазе* запрос во внутреннем представлении подвергается логической оптимизации. Оптимизатор применяет правила для преобразования запроса в формат, оптимальный с точки зрения синтаксиса.

- Эквивалентные, преобразования после проведения которых получается внутреннее представление, эквивалентное начальному.

- Семантические: получаемое преобразованное представление не является эквивалентным начальному, но гарантируется, что результат выполнения преобразованного запроса совпадает с результатом запроса в начальной форме.

После выполнения второй фазы обработки запроса его внутреннее представление остается непроцедурным, хотя и является более эффективным, чем начальный запрос.

Третья фаза обработки запроса состоит в:

- Построение альтернатив (оптимизатор разрабатывает альтернативы для его выполнения, в соответствии с его внутреннем представлением, полученным на второй фазе).

- Оптимизатор выбирает лучший план выполнения запроса (из набора эвристических правил или вычисляя стоимость для каждой альтернативы выполнения).

При оценках используется статистическая информация о состоянии базы данных, доступная оптимизатору. Из полученных альтернативных планов выбирается наиболее дешевый.

На *четвертой фазе* по выбранному оптимальному плану выполнения запроса формируется выполняемое представление плана. Он может быть программой в машинных кодах, если система ориентирована на компиляцию запросов в машинные коды, или быть машинно-независимым, но более удобным для интерпретации, если система ориентирована на интерпретацию запросов. В нашем случае это непринципиально, поскольку четвертая фаза обработки запроса уже не связана с оптимизацией.

Пятая фаза обработки запроса заключается в его реальном выполнении [1].

Исследования. Существуют два принципиально разных подхода к оптимизации запросов. В первом подходе оптимизатор использует информацию о механизмах реализации путей доступа, то есть метод оптимизации основан на синтаксисе (на правилах). Во втором подходе основой является статистическая

информация о распределении данных, где используется метод оптимизации, основанный на стоимости (на издержках) [2, с.140]. Рассмотрим их подробнее.

1. При использовании метода оптимизации, основанный на синтаксисе, план составляется на основании существующих путей доступа и их рангов. Ранжирование путей доступа выполняется на основании знаний о правилах и последовательности осуществления этих путей [2, с.140].

Такой метод оптимизации учитывает иерархическое старшинство операций. Если для какой-либо операции существует более одного пути ее выполнения, то выбирается тот путь, чей ранг выше по старшинству, так как в большинстве случаев он выполняется быстрее, чем путь с более низким рангом. План выполнения запроса формируется из выбранных путей доступа с максимальными по старшинству рангами (рис. 2).

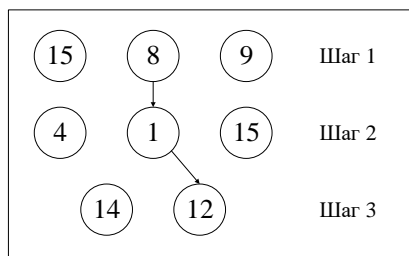


Рис.2. Построение плана выполнения запроса в методе оптимизации по синтаксису

2. При использовании метода оптимизации, основанного на стоимости, оптимизатор сначала строит несколько возможных планов выполнения запроса. При этом он применяет некоторые эвристики. Эти правила позволяют сузить пространство поиска оптимального плана благодаря тому, что неэффективные планы не рассматриваются [2, с.142].

Для каждого из построенных планов рассчитывается его стоимость. Она определяется на основании статистики данных в таблицах, к которым обращается команда, и связанными с ними кластерами и индексами. Статистика касается распределения значений данных в таблице и может включать в себя количество записей в таблице, минимальные и максимальные значения данных по столбцам и другие параметры. Стоимость – это оценка ожидаемого времени выполнения запроса с использованием конкретного плана выполнения. Оптимизатор может учитывать количество необходимых ресурсов памяти, стоимость операций ввода-вывода, времени процессора и оперативной памяти, необходимой для выполнения плана.

Для сравнения планов выполнения для оптимизатора в качестве критериев оптимизации может выступать:

- Наилучшая общая производительность системы. Она достигается путем выборки из таких планов, которые требуют меньше всего ресурсов. Это позволит увеличить степень параллельности работы системы и повысить общую производительность, но при этом время выполнения отдельных запросов увеличится.

- Минимальное время реакции – время, необходимое для обработки первой строки. Этот критерий чаще применяется при работе в интерактивном режиме, но только для запросов, которые не содержат агрегирующих функций и не требуют сортировки данных результата.

- Минимальные затраты времени на обработку всех строк, к которым обращается данная команда. Этот критерий используется обычно при работе в пакетном режиме или когда невозможно выдавать результат по частям.

СУБД, использующие метод оптимизации по стоимости, могут учитывать распределение данных и характеристики хранения таблиц, распределение значений в столбце. Последняя характеристика может быть определена с помощью гистограмм. Все множество значений столбца упорядочивается и разбивается на N интервалов (рис.4). Гистограмма помогает определить равномерность или неравномерность распределения данных. На рис.4,б представлено неравномерное распределение данных для некоторого столбца F. В словарь-справочник данных записываются полученные значения столбца (1,4,4,5,6,10,11,20,35,60,100). При анализе запроса с условием ($F < 5$) система сможет с помощью этой гистограммы определить, что через индекс придется выбрать не менее 15% записей таблицы.

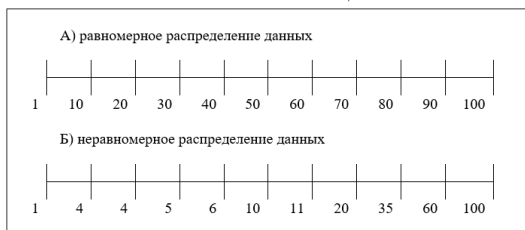


Рис.4. Примеры равномерного (а) и неравномерного (б) распределения значений

Существуют различные подходы к порядку сбора статистики. Некоторые СУБД постоянно собирают статистическую информацию, но это может повлиять на быстрдействие системы. Другие позволяют осуществлять сбор статистики периодически, например, в период минимальной загрузки системы. Третьи предлагают администратору специальные средства для сбора статистики, которые запрашиваются интерактивно по его команде.

Гистограммы полезны только в тех случаях, когда они отражают актуальное распределение данных. Если распределение меняется, гистограмму нужно обновлять; если изменение данных не изменяет их распределения, то

обновление гистограммы не требуется. Построение гистограммы бесполезно в следующих случаях:

- значения столбца распределены равномерно;
- столбец не используется в предикатах запросов;
- значения столбца уникальны и используются только в предикатах эквивалентности.

Выводы. В результате исследования было определено, что оптимизаторы реляционных СУБД представляют собой наиболее сложные компоненты систем. Эффективность системы в целом во многом определяется тем, насколько мощной является оптимизация. Цель оптимизации состоит в обеспечении как можно более быстрого получения ответа на запрос с минимальным числом обращений к БД. Из рассмотренных типов оптимизации можно сделать вывод, что оптимизация по синтаксису использует тот факт, что в языке SQL эффективность запроса зависит от того, как он сформулирован. В данном случае оптимизация зависит от квалификации программиста. При оптимизации основанной на стоимости происходит сбор сведений о БД — числе таблиц, числе строк, типе данных в каждой строке, доступности индексирования для конкретного столбца. Оптимизатор использует эту информацию для выработки наилучшего плана обработки запросов.

Преимущества метода оптимизации основанного на стоимости: задача определения наилучшего способа выполнения запроса перекладывается с пользователя на процессор БД.

К недостатку можно отнести, то что нахождение оптимального метода может занять много времени.

Для реализации оптимальных запросов как по синтаксису, как и по стоимости необходимо выполнять действия в следующем порядке:
Вычисление выражений и условий, содержащих константы.

1. Преобразование сложной команды в эквивалентную ей с использованием соединения (проводится не всегда).
2. Если команда выполняется над представлением, то оптимизатор объединяет запрос на создание представления и запрос к этому представлению в одну команду.
3. Выбор метода оптимизации.
4. Выбор путей доступа к таблицам, к которым обращается запрос.
5. Выбор порядка соединения (если в запросе соединяются несколько таблиц, то оптимизатор определяет, какие две таблицы будут соединяться первыми, какая таблица следующей будет подключаться в результате и т.д.).
6. Выбор операции соединения для каждой команды соединения.

Список литературы

1. Jhingram A. A Performance Study of Query Optimization Algorithms on a Database System Supporting Procedural Objects // Proc. 14th Int. Conf. Very Large Data Bases, Los Angeles, Calif., Aug.-Sept. 1988. Los Altos, Calif., 1988.- С. 88-99.
2. И.П. Карпова – Базы данных: Учебное пособие. – СПб.: Питер, 2013. – 240 с.: ил. – (Серия «Учебное пособие»).

УДК 004.652

Исследование оптимальной структуры базы данных в рамках автоматизированной системы приемной комиссии ДонНТУ

Дерksen Е.А., Завадская Т.В., Меренкова Л.Л.

ГОУВПО "Донецкий национальный технический университет" кафедра компьютерной инженерии
E-mail: derksen.evgenij@yandex.ua

Аннотация

Дерksen Е.А., Завадская Т.В., Меренкова Л.Л. Исследование оптимальной структуры базы данных в рамках автоматизированной системы приемной комиссии ДонНТУ. Рассмотрено влияние структуры базы данных на производительность выполнения разных транзакций. Определен подход к проектированию базы данных для автоматизированной системы приемной комиссии ДонНТУ.

Общая постановка проблемы. На этапе проектирования базы данных (БД) следует учитывать такой аспект, как производительность выполнения транзакций. Важно определить подход к проектированию структуры БД, для обеспечения максимального быстродействия в рамках решения конкретной задачи (введение, хранение, редактирование, обработка информации в рамках автоматизированной системы (АС) «Приемная комиссия»).

Исследования.

Проектирование базы данных выполняется в три этапа [1]:

- 1) Концептуальное проектирование.
- 2) Логическое проектирование.
- 3) Физическое проектирование.

В данной работе рассматривается этап физического проектирования, так как одна из важнейших задач данного этапа - дать гарантию того, что база данных обеспечивает требуемый уровень производительности.

Обычно, производительность базы данных измеряется в терминах производительности транзакций. Для решения задачи повышения производительности нужно сформировать транзакции, которые могут существовать в базе данных, но в большинстве практических случаев сформировать изначально полностью прописанные транзакции к базе данных является весьма сложной организационной задачей. К тому же, часть проблем, связанная с производительностью, может быть выявлена лишь на стадии тестирования и опытной эксплуатации базы данных [2].

Прежде чем определить набор и тип используемых транзакций, следует оценить тип приложений, для которых разрабатывается БД. Перед этим следует уточнить термины транзакция и запрос. В теории баз данных под транзакцией понимают одну из команд SQL: SELECT, INSERT, UPDATE, DELETE. Однако, в зависимости от типа приложений, термин транзакция трактуется более свободно как элементарная логически завершенная единица работы (так называемая бизнес-транзакция), которая может включать несколько команд вставки, удаления или модификации. В зависимости от того, какие команды SQL используются, транзакции разделяют на транзакции только для записи, только для модификации, только для чтения, только для удаления. Транзакции только для чтения называют запросом.

Исходя из такого деления транзакций по типам обработки с учетом частоты транзакций каждого типа, можно выделить основные классические типы приложений баз данных:

OLTP-системы (On-Line Transaction Processing) - это такое приложение, которое содержит в основном транзакции вставки, обновления и удаления, с высокой частотой преимущественно транзакций обновления. Для таких систем характерен высокий уровень параллелизма, который в данном случае означает, что много пользователей используют базу данных одинаковым образом.

DSS-системы (Decision Support System) - это такое приложение, которое работает с очень большой базой данных в режиме "только чтение". Обычно используется набор фиксированных простых запросов или неогламентированные запросы пользователей.

OLAP-системы (On-Line Analytical Processing) - это приложение, которое обеспечивает аналитическую обработку данных, включающую математический, статистический или иной анализ данных. Такие системы нельзя отнести полностью либо к OLTP-, либо к DSS -системам. Они располагаются где-то между ними. В рамках OLAP систем выделяют так называемые ROLAP системы (Relational OLAP), т.е. OLAP -системы, использующие реляционные базы данных. Типичные OLAP-системы разрабатываются обычно под многомерные модели данных.

VCDB-системы (Variable Cardinality Database) - это такое приложение обработки данных, для которого база данных растет или сжимается в размерах периодически в зависимости от характера обработки данных. Обычно размер этих

баз данных постоянно растет. Кардинальность относится к числу строк в таблицах базы данных в текущий момент времени.

Для того чтобы определить тип приложений, используемых в АС «Приемная комиссия», следует рассмотреть реализуемый ими функционал. АС «Приемная комиссия» включает в себя следующие функции:

- 1) Хранение данных всех абитуриентов университета в единой базе.
- 2) Персонализированный доступ к информации с разграничением прав пользователей в соответствии с должностными полномочиями.
- 3) Регистрация анкетных данных, паспортных данных, фотографий и заявлений абитуриентов.
- 4) Редактирование данных анкет и заявлений абитуриентов, подавших документы
- 5) Учет документов, поданных абитуриентом.
- 6) Формирование и контроль плана набора.
- 7) Формирование необходимого пакета документов для абитуриента (расписки, заявление, анкета и т.д.).
- 8) Формирование списка рекомендованных к зачислению в вуз.
- 9) Зачисление абитуриентов (формирование списков, протоколов и приказов на зачисление).
- 10) Формирование статистической отчетности для анализа данных об абитуриентах, поданных заявлениях и результатах вступительных испытаний в разрезе факультетов, специальностей, форм обучения, видов финансирования и др.

Исходя из этого можно сделать вывод, что отдельные части АС «Приемная комиссия» являются как OLTP-системами, так и ROLAP-системами. Следовательно, набор транзакций будет состоять из транзакций вставки, обновления и удаления, а также запросов для выборки и анализа данных. Для построения списка транзакций, используемых в БД, необходимо определить основные объекты модели данных АС «Приемная комиссия»:

- 1) Персона – описывает модель персональных данных абитуриента;
- 2) Дело – описывает модель данных заявления абитуриента;
- 3) Объект справочной информации – описывает модель данных единицы справочной информации.

Объект «Персона» является составным и состоит из нескольких подобъектов:

- 1) Фото.
- 2) Паспортные данные.
- 3) Адреса и контактные данные.
- 4) Документы об образовании.

На основе функционала АС «Приемная комиссия» и основных объектов модели данных можно составить список основных транзакций БД:

- 1) Добавление новой персоны.
- 2) Добавления фото для персоны.
- 3) Ввод паспортных данных персоны.
- 4) Редактирование паспортных данных персоны.
- 5) Добавление адресных и контактных данных персоны.
- 6) Редактирование адресных и контактных данных персоны.
- 7) Добавление документа об образовании персоны.
- 8) Редактирование данных документа об образовании персоны.
- 9) Добавление нового дела.
- 10) Редактирование данных дела.
- 11) Получение справочной информации.
- 12) Получение списка дел с привязкой к данным персоны.
- 13) Получение статистики о поданных делах с группировкой по различным параметрам.

Поскольку АС «Приемная комиссия» предусматривает одновременную работу большого количества пользователей, выполняющих как однотипные, так и разные операции - следует оптимизировать производительность параллельного выполнения транзакций.

Для построения оптимальной структуры БД нужно учитывать особенности системы управления БД (СУБД), которая обеспечивает работу данной БД. В данном случае используется СУБД MS SQL Server 2012.

Исходя из документации MS SQL Server 2012 для повышения производительности транзакций вставки и обновления данных выдвигаются следующие требования [3]:

- Короткие транзакции для минимизации долгосрочных блокировок и улучшения параллелизма.
- Высокая нормализация базы данных.
- Минимизация исторических и статистических данных или полное их отсутствие.
- Минимизация использования индексов.
- А для запросов на получение данных из БД:
- Обильное индексирование.
- Денормализация базы данных.

Из приведенных выше требований к структуре БД можно сделать вывод, что невозможно оптимизировать структуру БД для обеспечения максимальной производительности как транзакций вставки, так и запросов на выборку данных. Для решения данной проблемы есть несколько вариантов:

- 1) Использовать разные базы данных для обслуживания разных транзакций.
- 2) Разработать структуру обеспечивающую средние показатели производительности для различных транзакций.
- 3) Определить приоритеты и частоту использования для каждого типа транзакций, и разработать структуру обеспечивающую максимальную производительность наиболее приоритетных и часто используемых транзакций.

Реализация первого варианта потребует дополнительных аппаратных затрат для обслуживания двух БД вместо одной. Второй вариант кажется оптимальным, однако требует большого количества времени и трудозатрат на разработку оптимальной структуры БД. Таким образом было принято решение использовать третий вариант.

Наиболее приоритетными задачами для АС «Приемная комиссия» являются ввод, учет, хранение данных абитуриентов и заявлений поданных ими, а самым важным этапом проведения приемной компании ДонНТУ является прием дел абитуриентов, в ходе которого частота ввода новых данных и редактирования имеющихся будет максимальной. Следовательно, структура БД должна быть ориентирована на обеспечение максимальной производительности транзакций записи и обновления.

Однако АС «Приемная комиссия» реализует еще один немаловажный функционал – публичный web-сервис, который отображает в реальном времени конкурсное положение абитуриентов. Web-сервер обеспечивающий работу данного сервиса генерирует большое количество запросов на выборку многосвязных данных по различным параметрам. Для обеспечения максимальной производительности выполнения данных запросов следует использовать механизм представлений, реализованный в MS SQL Server.

Представление — виртуальная (логическая) таблица, представляющая собой поименованный запрос (синоним к запросу), который будет подставлен как подзапрос при использовании представления. В отличие от обычных таблиц реляционных баз данных, представление не является самостоятельной частью набора данных, хранящегося в базе. Содержимое представления динамически вычисляется на основании данных, находящихся в реальных таблицах.

При использовании представлений выигрыш в производительности достигается за счет использования механизма кеширования запросов, реализованного в MS SQL Server, который позволяет сократить время выполнения одинаковых запросов. Также данная СУБД позволяет создать индексы для представления, не затрагивая при этом исходные таблицы.

Выводы. Из всего вышесказанного можно сделать вывод, что для создания оптимальной структуры БД АС «Приемная комиссия» следует обеспечить хранение основных объектов модели данных в различных таблицах,

избегать избыточности данных с помощью нормализации, минимизировать использование индексов, а для выборки многосвязных данных следует использовать представления.

Список литературы

1. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика — 3-е изд. — М.: «Вильямс», 2003. — 143 с
2. Создание физической модели базы данных. Учет влияния транзакций // НОУ Интуит [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/1095/191/lecture/4985>
3. Сравнение оперативной обработки транзакций // Microsoft TechNet [Электронный ресурс]. – Режим доступа: [https://technet.microsoft.com/ru-ru/library/ms187669\(v=sql.105\).aspx](https://technet.microsoft.com/ru-ru/library/ms187669(v=sql.105).aspx)

УДК 004.42

Перспективы применения технологий искусственного интеллекта при разработке системы поддержки принятия решений в составе программного комплекса автоматизации формирования нагрузки преподавателей кафедры

Прищепа В.В., Краснокутский В.А., Меренкова Л.Л.

Донецкий национальный технический университет

кафедра компьютерной инженерии

E-mail: prishepa_vladimir_94@mail.ru

Аннотация

Прищепа В.В., Краснокутский В.А., Меренкова Л.Л. Перспективы применения технологий искусственного интеллекта при разработке системы поддержки принятия решений в составе программного комплекса автоматизации формирования нагрузки преподавателей кафедры. Рассмотрены проблемы и задачи разработки приложения для автоматизации процесса формирования нагрузки преподавателей, из которых вытекает необходимость применения технологий искусственного интеллекта. Предложены несколько путей решения. На основании, поставленной задачи, выбран вариант с использованием искусственной нейронной сети, описана ее структура, алгоритм обучения.

Общая постановка проблемы. Разрабатываемое приложение должно предоставлять пользователю, которым является ответственный человек на

кафедре, удобный графический интерфейс для распределения дисциплин между преподавателями.

Принято решение о включении в данное приложение подсистемы, которая должна давать рекомендации по предоставлению дисциплины тому или иному преподавателю на основе анализа некоторого набора данных о преподавателе и дисциплине.

С целью исключения ситуации повторения системой ошибок, допущенных ранее, она должна быть обучаемой.

Пути решения проблемы. Теория принятия решений является широкой областью знаний и предполагает творческий подход при разработке систем поддержки принятия решений (СППР). Если система обладает интеллектуальными свойствами, то говорят об интеллектуальной СППР (ИСППР). Такие системы не действуют в соответствии с жестким алгоритмом, а используют подход, основанный на адаптации к текущим условиям работы и, как правило, это эвристический подход формирования решения на основе заданных входных параметров.

Для обеспечения указанных свойств могут использоваться методы, основанные на:

- применении эвристических алгоритмов, например, генетический алгоритм;
- применении искусственных нейронных сетей (ИНС).

Оба подхода являются довольно популярными при разработке систем искусственного интеллекта. Учитывая специфику задачи и необходимость предотвращения повторения ошибок, а значит обеспечить обучение системы, разработчик склоняется к применению СППР с использованием ИНС.

Выбор архитектуры и класса сети

С помощью ИНС решаются широкий круг задач:

- распознавание образов и классификация;
- кластеризация;
- прогнозирование;
- аппроксимация;
- анализ данных;
- принятие решений и управление.

Во всех, описанных выше, классах задач на практике успешно применяются перцептроны (рис. 1) – это одна из первых моделей нейронной сети, предложенная Френком Розенблаттом в 1957 году [4]. Перцептрон обладает высокой степенью универсальности, успешно обучается и может решать широкий круг сложных задач.

Описанная задача, относится к классу задач принятия решений. При ее решении нет необходимости использовать сложные модели нейронных сетей,

таких как сеть Элмана, которая используется для решения задач в реальном времени, к примеру, управление движущимся объектом.

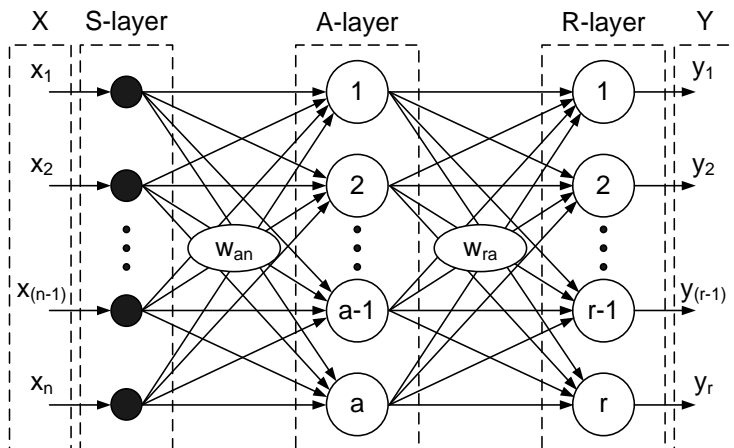


Рис. 1. Двухслойный перцептрон с n входами, скрытым слоем из a нейронов и r выходами

На рисунке 1 изображена классическая структура перцептрона. Набор входных данных – вектор X , результат – вектор Y , S-layer (от англ. sensor) – входы сети, их количество зависит от решаемой задачи, A-layer (от англ. association) – ассоциативный или скрытый слой, выбор необходимого количества нейронов этого слоя является основной задачей проектирования нейронной сети, R-layer (от англ. reaction) – выходной слой, или слой реагирования, количество выходов сети зависит от конкретной задачи. Каждый чувствительный элемент S-слоя связан с каждым элементом A-слоя синапсами (межнейронными связями), w_{an} – отдельно взятый синапс из множества между слоями S и A, аналогичным образом происходит коммуникация нейронов между слоями A и R.

Перцептрон может быть и однослойным (если убрать слой A и сенсоры подключить к выходному слою), скрытых же слоев может быть несколько, на практике применяются сети с одним, или двумя скрытыми слоями.

Перцептрон был открыт в однослойном виде, такая структура позволяла очень просто его обучать с помощью правил Хебба и дельта-правила формулы (1) и (2), но с его помощью решался узкий круг задач [4].

$$\Delta w_{ij} = \eta \delta_i x_j, \quad i \in [1; r], \quad j \in [1; n] \quad (1)$$

$$\delta_i = (e_i - y_i) y_i (1 - y_i), \quad i \in [1; r]; \quad (2)$$

Формула (1) описывает изменение весовых коэффициентов при обучении сети с использованием правил Хебба, в ней используются следующие обозначения:

- Δw_{ij} – изменение весового коэффициента соответствующего синапса;
- δ_i – ошибка i -го выхода сети;
- η – коэффициент скорости обучения;
- r – количество выходов сети;
- n – количество входов сети.

Формула (2) описывает вычисление ошибки выхода сети (дельта-правило), в ней используются следующие обозначения:

- δ_i – ошибка i -го выхода сети;
- e_i – ожидаемое значение выхода сети (expected);
- y_i – фактическое значение выхода;
- r – количество выходов сети.

Построение многослойных сетей резко расширяло круг решаемых задач, но вставала проблема их обучения. Обучение однослойного персептрона проходило путем подачи входных воздействий и вычисления погрешности на выходе (дельта-правило), после чего происходила корректировка весовых коэффициентов по правилам Хебба. Этот простой и элегантный метод невозможно применить к скрытым слоям сети, так как невозможно вычислить ошибку, не зная какие значения выходных сигналов скрытого слоя являются правильными [2].

Эффективным способом обучения многослойных персептронов является алгоритм обратного распространения ошибки [1]. Идея состоит в том, чтобы погрешность выходных сигналов скрытых слоев выражать через произведение суммы ошибок выходного слоя и весовых коэффициентов соответствующих синапсов [2].

$$(e_j - y_j) = \sum_{i=1}^r \delta_i w_{ij}, \quad j \in [1; a]; \quad (3)$$

Используя формулы (1) и (2), подставив (3) в (2), имеем для скрытого слоя [2]:

$$\Delta w_{ij} = \eta \delta_i x_j, \quad i \in [1; a], \quad j \in [1; n]; \quad (4)$$

$$\delta_j = y_j (1 - y_j) \sum_{i=1}^r \delta_i w_{ij}, \quad j \in [1; a]; \quad (5)$$

Обозначения в формулах (3), (4), (5) имеют тот же смысл, что и в формулах (1), (2).

То есть, метод обучения многослойных сетей по алгоритму обратного распространения ошибки является модификацией метода обучения однослойного персептрона, учитывающей особенности архитектуры многослойной сети.

ИНС классифицируются по следующим параметрам:

- тип входной информации:
 - аналоговая (вход ИНС принимает информацию в форме чисел);
 - двоичная (вход ИНС принимает информацию в двоичном виде);
- способ обучения:
 - обучение с учителем (Производится на основе набора исходных данных и корректных результатов для них. Правильно спроектированная и обученная сеть обладает свойством обобщения, что означает корректную реакцию на наборы входных данных, которых не было в обучающей выборке.);
 - обучение без учителя (Пригодно для задач, в которых известны множества объектов, и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между ними, то есть при отсутствии правильных ответов.);
 - обучение с подкреплением (Частный случай обучения с учителем, но учителем выступает некая среда, или ее модель, она реагирует на принятые решения, формируя сигналы подкрепления.);
- по характеру настройки синапсов:
 - с фиксированными связями (весовые коэффициенты константны);
 - с динамическими связями (в процессе обучения происходит настройка весовых коэффициентов связей между нейронами).

В рамках решаемой задачи информация о преподавателе и дисциплине может быть представлена в разных видах (как в булевом, так и в числовом). По типу входной информации проектируемая ИНС аналого-цифровая. Обучение сети базируется на основании выборки за последние годы, это способ обучения с учителем. Так как взаимосвязь между наборами входных данных и выходных не известна, то перед обучением весовые коэффициенты необходимо задать случайными малыми значениями. Результатом обучения должна стать сеть со значениями весовых коэффициентов, которые обеспечат корректный выходной вектор при определенном входном. Данную сеть можно классифицировать как сеть с динамическими связями.

Выводы. Современные технологии искусственного интеллекта позволяют решать широкий круг задач, и они легко адаптируются для рассматриваемой задачи. С учетом некоторых особенностей и необходимости обучения подсистемы, было принято решение реализовать ее на основе искусственной нейронной сети и возможностей архитектуры многослойного персептрона. Более точные модели потребуют усложнения математических вычислений, что повлечет за собой повышение требований разрабатываемого программного обеспечения к характеристикам компьютеров.

Список литературы

1. Джонс М. Т. Программирование искусственного интеллекта в приложениях / М. Тим Джонс; Пер. с англ. Осипов А. И. - М.: ДМК Пресс, 2004. - 312 с.
2. Ясницкий Л. Н. Введение в искусственный интеллект: учебное пособие для студ. высш. учеб. заведений / Л. Н. Ясницкий – 2-е изд., испр. – М.: Издательский центр «Академия», 2008. – 176с.
3. Ясницкий Л. Н. Использование методов искусственного интеллекта в изучении личности серийных убийц / Л.Н. Ясницкий, С.В. Ваулева, Д.Н. Сафонова, Ф.М. Черепанов // Криминологический журнал Байкальского государственного университета экономики и права. — 2015. — Т. 9, № 3. — С. 423–430.
4. Хайкин С. Нейронные сети: полный курс, 2-е издание. : Пер. с англ. – М. : Издательский дом «Вильямс», 2006. – 1014 с.

УДК 004.351

Вопросы организации аналого-цифрового преобразования в устройствах на базе микроконтроллера ADuC841

Плахин С.В., Яременко П.А., Краснокутский В.А.

Донецкий национальный технический университет
кафедра компьютерной инженерии
e-mail: pavel.yaremenko@gmail.com

Аннотация

Плахин С.В., Яременко П.А., Краснокутский В.А. Вопросы организации аналого-цифрового преобразования в устройствах на базе микроконтроллеров ADuC841. Рассмотрены особенности работы аналого-цифрового преобразователя микроконтроллера ADuC 841. Приведен вариант программы аналого-цифрового преобразования сигнала. Предложены структуры устройств, которые целесообразно реализовать на аналоговых микроконтроллерах ADuC 84.

Общая постановка проблемы. В последние годы увеличился интерес к построению различных бытовых устройств на микроконтроллерах, работающих совместно с персональным компьютером (ПК). Персональный компьютер значительно повышает вычислительные возможности устройств и обеспечивает хорошее отображение информации на экране дисплея. К таким устройствам можно отнести цифровые осциллографы, домашние метеостанции, различные медицинские приборы, в частности, электрокардиографы и т.п. При создании таких приборов часто требуется измерять аналоговые сигналы. Для этих целей особенно удобны микроконтроллеры со встроенными аналого-цифровыми преобразователями (АЦП). Так широко используемые недорогие

микроконтроллеры с RISC архитектурой Atmega фирмы ATMEL и PIC фирмы Microchip содержат в своем составе 10 – разрядные АЦП и позволяют измерять напряжения на одном из восьми аналоговых входах. Однако ограниченный объем внутренней памяти данных, и сложность построения внешней памяти ограничивают использование этих контроллеров в устройствах, требующих ввода и хранения больших массивов данных. Да и реальная разрешающая способность АЦП этих контроллеров не всегда обеспечивают необходимую точность измерений.

Пути решения проблемы. Увеличения качества аналого-цифрового преобразования возможно за счет использования отдельной микросхемы АЦП. Но это приводит к усложнению и удорожанию системы.

Фирма Analog Devices разработала линейку специализированных микроконтроллеров ADuC 8xx для сбора и обработки аналоговых сигналов на базе ядра с архитектурой MCS 8052 [1]. Отличительной особенностью этих устройств является наличие аналого-цифровых и цифро-аналоговых (ЦАП) преобразователей с высокими характеристиками. Кроме этого устройства имеют расширенную память программ и данных на кристалле, встроенные последовательные интерфейсы UART, SPI, I²C. Напряжение питания микросхем может быть, как 5В, так и 3В.

Микроконтроллеры ADuC 8xx отличаются быстродействием, объемом внутренней памяти программ и данных, характеристиками АЦП и ЦАП, набором периферийных устройств. Среди микроконтроллеров линейки ADuC 84x микроконтроллер ADuC 841 обладает самым высоким быстродействием 20 MIPS, памятью программ объемом в 62 Кбайт, возможностью расширения внешней памяти данных до 16 Мбайт, внутренней памятью данных до 2304 байт, двенадцати разрядным АЦП со скоростью преобразования до 400 KSPS.

Одним из недостатков микропроцессорного ядра с архитектурой MCS 52 является низкое быстродействие, обусловленное тем, что машинный цикл осуществляется за 12 машинных тактов. В новой серии аналоговых микроконтроллеров ADuC84x этот недостаток в значительной мере преодолен. За счет использования конвейерного принципа 80% команд микроконтроллера выполняется за один машинный такт, т.е. быстродействие увеличилось почти в 12 раз по сравнению с рассмотренными выше сериями.

Для разработки и отладки программного обеспечения ADuC8xx применяются различные программные средства.

Интегрированная программная среда Keil μ Vision 5 обеспечивает возможность разработки программного обеспечения на языках C и Ассемблер, а также отладку программ с использованием программного симулятора или внутрисхемного эмулятора. Программная среда может использоваться для программирования практически всех микроконтроллеров с архитектурой MCS51 и MCS52, включая все аналоговые микроконтроллеры ADuC8xx. Симулятор

пакета Keil позволяет выполнять программу в пошаговом режиме как на уровне команд языка C, так и на уровне машинного кода.

Все эти качества микроконтроллера ADuC 841 обеспечивают решение многих проблем, которые возникают при проектировании устройств с повышенными требованиями к точности преобразования, производительности, к объему измеряемых и хранимых данных.

Аналого-цифровой преобразователь ADuC 841. Микроконтроллер ADuC841 содержит высокопроизводительный 8-ми каналный АЦП, питающийся однополярным напряжением [2]. В состав блока АЦП входят: многоканальный мультиплексор, усилитель выборки-хранения, встроенный источник опорного напряжения (ИОН), средства калибровки и собственно АЦП. Все элементы блока конфигурируются и управляются с помощью трех регистров специальных функций.

АЦП построен по схеме последовательного приближения с емкостным ЦАП. АЦП обеспечивает преобразование от 0 В до величины опорного напряжения (V_{ref}). Внутренний источник опорного напряжения – это высокоточный откалиброванный на заводе источник с малым дрейфом и величиной 2,5В. Вместо внутреннего источника опорного напряжения может быть подключен внешний источник, величина которого может быть в пределах от 1 В до напряжения питания AV_{DD} .

Однократный или непрерывный режимы преобразования могут инициироваться программно или с помощью внешнего сигнала запуска. Периодический запуск преобразователя может быть осуществлен также с помощью таймера 2. АЦП может быть сконфигурирован для работы в режиме прямого доступа в память (ПДП). В режиме ПДП АЦП непрерывно преобразует и записывает данные во внешнюю память (XRAM) без участия центрального процессора. Для этого может быть использована вся внешняя память объемом 16 Мбайт.

Пример программы аналого-цифрового преобразования.

Пример тестовой программы работы АЦП ADuC 841 приведен на рис. 1. Программа написана на языке C в среде программирования Keil μ Vision. Отладка программы была проведена с помощью оценочной платы MicroConverter SAR Eval Board (Rev. A3).

Программа измеряет сигнал, который поступает на заданный канал и записывает его значение в массив внутренней памяти XRAM. Номер измеряемого канала задается с помощью переменной CHAN. Периодический запуск АЦП осуществляется таймером 2.

В памяти микроконтроллера зарезервирован массив типа unsigned char из 100 элементов, каждый из которых имеет размерность один байт. Для работы с внутренней памятью микроконтроллера необходимо установить нулевой бит регистра CFG841. Это осуществляется посредством побитного «ИЛИ» с соответствующей маской. Затем осуществляется инициализация АЦП и выбор

работы по переполнению таймера 2. В регистр ADCCON1 записывается соответствующее значение. В регистр ADCCON2 записывается номер канала, с которого осуществляется считывание сигнала. Переменной CHAN мы можем устанавливать номер канала, а также подавать на вход АЦП землю или напряжение V_{ref} . Таймер 2 в данном случае работает в режиме автозагрузки, поэтому в регистры RCAP2H и RCAP2L записывается число, которое будет загружаться в счетчик таймера при его очередном переполнении. С помощью битов EA и EADC, разрешаются общие прерывания и прерывание от АЦП. Запуск таймера 2 осуществляется путем установки TR2 в единицу. После этого следует цикл, который повторяется до тех пор, пока включен таймер 2. При переполнении таймера происходит прерывание и выполняется запись в массив данных значений из регистров ADCDATAH и ADCDATAL, которые являются результатом преобразования АЦП. В самом прерывании при каждом вхождении выполняется инкремент счетчика считываний и в конце выполняется проверка. Если все сто значений считаны и записаны в память, осуществляется остановка таймера, и запрещаются прерывания, путем обнуления TR2, EADC, EA. Программа выходит из цикла while и попадает в бесконечный цикл с меткой, имитирующий ожидание.

Фрагмент листинга тестовой программы:

```
int k=0,cikl=0;
unsigned char xdata array[100] _at_ 0x000000;
unsigned char mask;
void adc_int() interrupt 6{
    array[k] = ADCDATAH;
    k++;
    array[k] = ADCDATAL ;
    k++;
    cikl++;
    if (cikl>50) {
/*Отключение прерываний и таймера*/
        EA = 0;
        EADC = 0;
        TR2=0;
    }
    return;
}

void main()
{
int CHAN=0x01B;        //Установка канала
mask = 1;
CFG841 = CFG841 | mask;    //Разрешение работы с внутренней
памятью
```

```

ADCCON1 = 0x09E;           // Включение АЦП и режим работы по
таймеру
ADCCON2 = CHAN ;          // Выбор канала
RCAP2L = 0x011;           //Число для перезагрузки счетчика
RCAP2H = 0x0FF;
TL2    = 0x0D2;
TH2    = 0x0FF;
/*Активация таймера 2*/
EA     = 1;                // Разрешить прерывания
EADC   = 1;                // Разрешить прерывания АЦП
TR2    = 1;                // Запустить таймер 2

while (TR2==1){}
m1: goto m1;
    
```

Электрокардиограф на базе ADuC 841. В работе [3] предлагается простой электрокардиограф на базе аналогового контроллера ADuC841. Структурная схема кардиографа показана на рис.2. Кардиограф подключается к персональному компьютеру, в котором осуществляется основная обработка сигнала, формирование и отображение кардиограммы. Существует возможность передачи сигнала в систему моделирования MATLAB. Использование системы моделирования MATLAB позволяет разрабатывать и отлаживать алгоритмы цифровой обработки сигнала кардиограммы. Рассматриваемый кардиограф можно использовать для разработки новых алгоритмов цифровой обработки биопотенциалов.

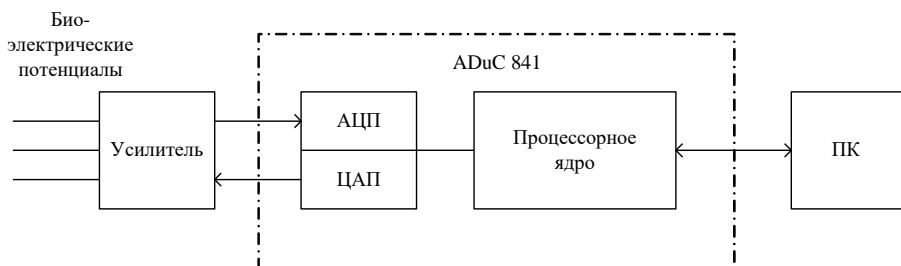


Рис.2. Структурная схема кардиографа

Цифровой осциллограф на микроконтроллере ADuC 841. В работе [4] рассматривается цифровой осциллограф на базе аналогового контроллера ADuC 841. Отличительной особенностью данного осциллографа является возможность сопряжения его с системой моделирования MATLAB. Система MATLAB дает

мощные средства для цифровой обработки полученных сигналов. В сочетании с возможностями MATLAB визуализации результатов обработки сигналов можно получить удобную среду для экспериментального исследования различных объектов.

Структурная схема осциллографа представлена на рис. 3.

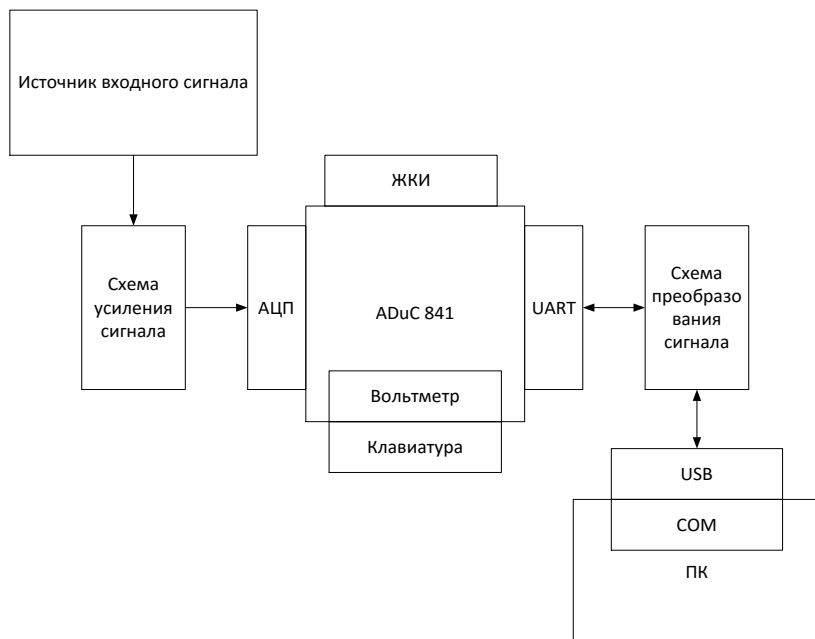


Рис. 3. Структурная схема цифрового осциллографа

Устройство может работать в двух режимах:

- с персональным компьютером;
- автономном.

При работе с персональным компьютером выполняются основные стандартные режимы осциллографа. Режимы работы осциллографа, номера каналов, частота дискретизации сигналов и т.п. задаются с помощью управляющей программы. Предусмотрен режим однократного ввода сигнала с последующей обработкой в системе моделирования MATLAB. Использование системы моделирования MATLAB позволяет разрабатывать алгоритмы цифровой обработки сигналов с использованием реальных сигналов, что повышает скорость и качество разрабатываемых алгоритмов.

Связь с компьютером осуществляется с помощью универсального последовательного порта (UART) через интерфейс RS-232. Так как многие современные компьютеры не обладают последовательным портом (интерфейсом RS-232), то предусмотрена возможность подключения к интерфейсу USB через схему преобразования интерфейсов. Причем персональный компьютер с устройством может работать как с виртуальным COM - портом, так и с устройством USB.

В автономном режиме работы прибор используется для измерения постоянных напряжений или действующих напряжений синусоидального тока. Этот режим также используется для тестирования прибора.

Вывод результатов измерений при этом может выполняться посредством использования ЖКИ базового аналогового контроллера ADuC 841.

Выводы. Рассмотрены способы организации аналого-цифрового преобразования в микроконтроллерах ADuC 841.

Предложены структуры устройств с повышенными требованиями к качеству аналого-цифрового преобразования и объему измеряемых и хранимых данных, которые могут быть реализованы на микроконтроллерах ADuC 841. Рассмотрены структурные схемы портативного кардиографа и цифрового осциллографа на базе микроконтроллера ADuC 841 с возможностью передачи полученных данных в персональный компьютер.

Список литературы

1. Зайцев О., Троицкий Ю. Аналоговые микроконтроллеры ADuC83x и ADuC84x //Электронные компоненты, 2005. - №1. – С. 45 – 48.
2. AduC841/ AduC842/ AduC843 - [Электронный ресурс].- Режим доступа: http://www.analog.com/media/en/technical-documentation/data-sheets/ADUC841_842_843.pdf
3. Плахин С.В. Портативный кардиограф на базе микроконтроллера ADuC 841/ Плахин С.В., Краснокутский В.А.- Компьютерная и программная инженерия. Сборник материалов международной научно- технической конференции студентов, аспирантов и молодых учёных 15-16 декабря 2015 года . - Донецк, ДонНТУ - 2015. – С. 13 – 14.
4. Яременко П.А. Разработка структуры цифрового осциллографа / Яременко П.А., Краснокутский В.А. - Компьютерная и программная инженерия. Сборник материалов международной научно- технической конференции студентов, аспирантов и молодых учёных 15-16 декабря 2015 года. - Донецк, ДонНТУ - 2015. – С. 18 – 19.

Секция № 9
Системный анализ
и управление



УДК. 517.978

Основные функции и задачи системы учета реестра продаж мототехники в мотосалоне

Громоклицкая М.Н., магистрант.

Донецкий национальный технический университет

Кафедра ИИСА

E-mail: malush123321@yandex.ru

Аннотация

Громоклицкая М.Н. Основные функции и задачи системы учета реестра продаж мототехники в мотосалоне. Рассматриваются существующие системы автоматизации продаж, проводится сравнительный анализ их основных характеристик. Проводится системный анализ мотосалона «МОТОДОМ Харьков». Выделяются основные требования, задачи и характеристики информационной системы реестра продаж автотранспорта в мотосалоне «МОТОДОМ Харьков».

Общая постановка проблемы. В настоящее время процесс автоматизации затронул не только производственную, техническую и технологическую сферы деятельности человечества, но и информационное пространство. Системы автоматизации помогают сократить избыточность хранимых данных, также влияют на уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте, помогают ускорить оперативность получения необходимой информации, что значительно улучшает качество обслуживания клиентов.

Целью данной работы является усовершенствование информационной системы мотосалона «МОТОДОМ Харьков» при помощи рассмотрения уже существующих систем, учета их недостатков и достоинств, а также проведения системного анализа мотосалона «МОТОДОМ Харьков», для последующего выделения основных функций и бизнес-процессов.

Практическое значение полученных результатов заключается в увеличении объема продаж, получении оперативной информации, интересующей покупателя, тем самым, повышая качество обслуживания, что повлечет за собой привлечение новых клиентов в мотосалон.

Необходимо рассмотреть существующие системы автоматизации и провести их сравнительный анализ, для выделения их основных преимуществ и недостатков, что поможет улучшить информационную систему мотосалона «МОТОДОМ Харьков».

1. Программный продукт «Альфа-Мото: Мотосалон + Мотосервис + Мотозапчасти» предназначен для комплексной автоматизации учета на предприятиях мотобизнеса (магазинов оптовой и розничной продажи запчастей, мотосервисов и станций технического обслуживания мототехники.).

2. Программа МотоСалон предназначена для автоматизации деятельности мотосалонов, торгующих мототехникой, в части, касающейся ведения учета, продаж мототехники и формирования договорных документов. Также предусмотрено формирование и печать документов, специфичных для данного вида деятельности (справка–счет, ПТС, транзитный номер).

3. Система «1С: Предприятие 8.2» имеет в своей основе ряд механизмов, определяющих концепцию создания прикладных решений.

В качестве ключевых моментов можно выделить изоляцию разработчика от технологических подробностей, алгоритмическое программирование только бизнес–логики приложения, использование собственной модели базы данных и масштабируемость прикладных решений без их доработки.

Исследования. Состав прикладных механизмов «1С: Предприятия» ориентирован на решение задач автоматизации учета и управления предприятием. Использование проблемно-ориентированных объектов позволяет разработчику решать самый широкий круг задач складского, бухгалтерского, управленческого учета, расчета зарплаты, анализа данных и управления на уровне бизнес-процессов.

Потребность в удобных и эффективных средствах разработки программного обеспечения, а также быстрое развитие вычислительной техники, привели к появлению систем программирования, ориентированных на так называемую «быструю разработку», среди которых можно выделить Borland Delphi и Microsoft Visual Basic.

4. Delphi 7 представляет собой современную и мощную систему программирования и имеет определенные особенности.

У существующих аналогов были выявлены следующие недостатки:

- нет гибких механизмов разграничения доступа к данным;
- недостаточная функциональность;
- нет многопользовательского доступа;
- высокая цена.

Выше указанные программы «1С: Предприятие 8.2», а так же «Borland Delphi» могут быть использованы для улучшения работы мотосалона «МОТОДОМ Харьков», но было принято решение разработать автоматизированную систему учета реестра продаж мототехники в салоне, с учетом конкретных проблем и нюансов работы «МОТОДОМ Харьков».

Для того, чтобы обозначить бизнес-процессы, которые необходимы при разработке информационной модели, необходимо разобраться в организации,

составе и функциях рассматриваемого мотосалона. Для этого проведем системный анализ «МОТОДОМ Харьков».

Для описания модели «черный ящик» необходимо составить модели входов, выходов. Входами в системе «Мотосалон «МОТОДОМ Харьков»» являются:

- продукция с завода;
- клиент на обслуживание;
- каталоги продукции завода;
- финансовые средства.

В системе «Мотосалон «МОТОДОМ Харьков»» можно выделить такие выходы:

- каталоги продукции мотосалона;
- накладные;
- обслуженный клиент;
- реализованная продукция.

Модель «черный ящик» приведена на рисунке 1.

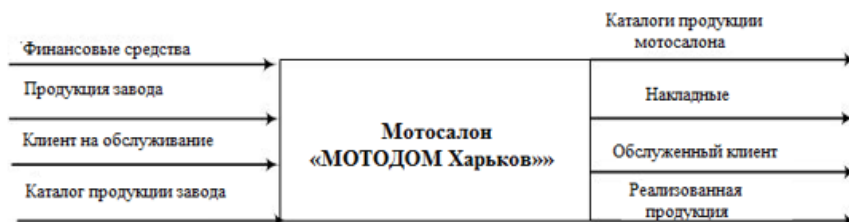


Рис. 1 – Модель «Черный ящик»

Для разработки модели состава проведем субстратный анализ, определив из каких основных элементов состоит мотосалон «МОТОДОМ Харьков».

Мотосалон «МОТОДОМ Харьков» можно разделить на: склад, руководство мотосалона, отдел продаж и бухгалтерию. Склад состоит из отдела тестирования, отдела приема продукции и отдела выдачи продукции. Руководство мотосалона «МОТОДОМ Харьков» это – генеральный директор мотосалона и директор. Отдел продаж состоит из главного менеджера, ведущего менеджера и менеджера. А бухгалтерия состоит из главного бухгалтера и бухгалтера.

На рисунке 2 приведена модель состава системы.

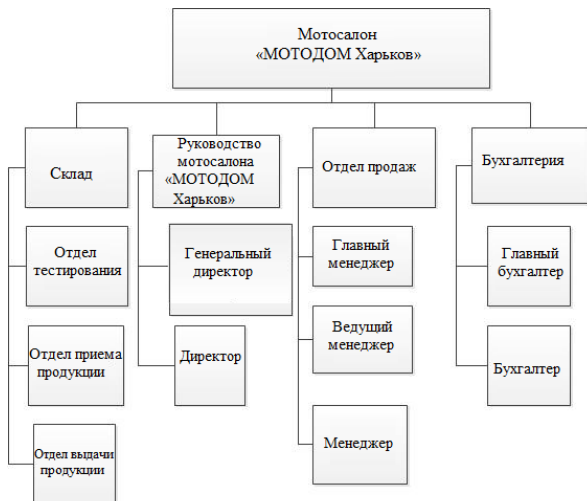


Рис. 2 – Модель состава системы

Осуществим функциональное описание системы, которое включает в себя раскрытие ее внешнего и внутреннего функционирования. Для разработки функциональной модели системы нам необходимо составить контекстную диаграмму, которая представлена на рисунке 3.

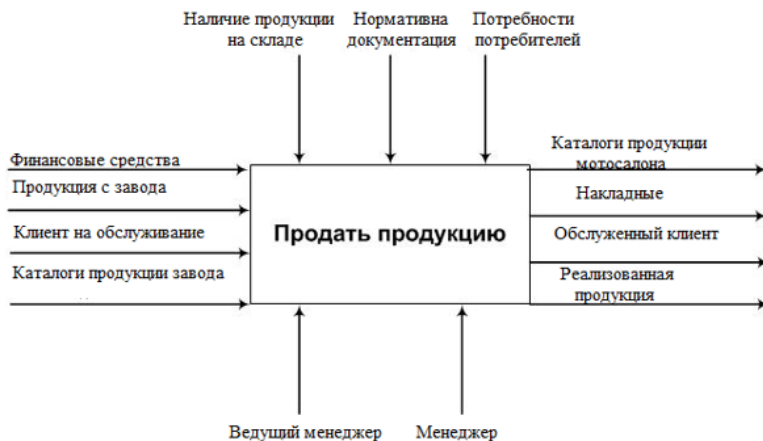


Рис. 3 – Контекстная диаграмма А-0

Декомпозированная контекстная диаграмма, которая необходима для более полного рассмотрения способности выполнения системой своей внешней функции, представлена на рисунке 4.

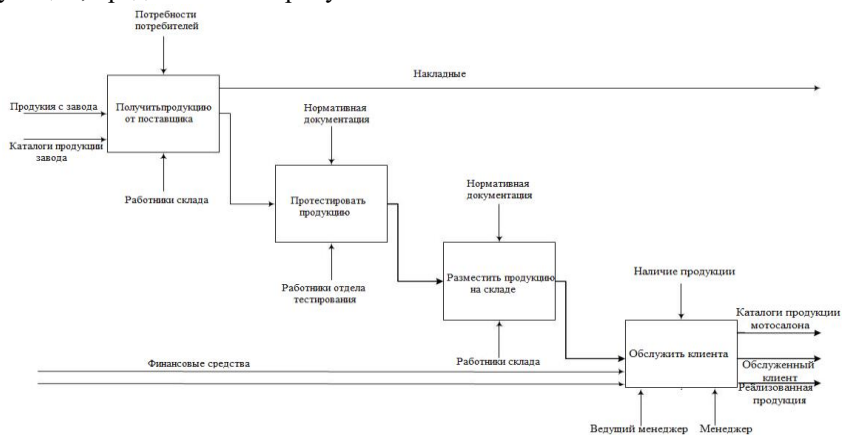


Рис. 4 - Декомпозированная диаграмма А0

На основе проведенных исследований необходимо выделить основные характеристики и задачи, а также бизнес-процессы, необходимые для создания информационной системы учета и реестра продаж мототранспорта «МОТОДОМ Харьков».

Рассмотрим бизнес – процессы, необходимые для учета реализации мототехники в мотосалоне:

1) заказ мототехники – после выбора оформляется заказ на выбранную модель, подготавливается и отправляется запрос на завод - изготовитель, принимается предоплата и выдается квитанция о предоплате.

2) прием мототехники – принятие на внутренний учет, проведение предпродажной подготовки и диагностики, оповещение покупателя.

3) Реализация мототехники – осмотр покупателем, оформление договора купли-продажи.

4) регистрация оплаты

5) формирование отчетных документов:

- формирование отчета «Прайс-лист»;
- формирование отчета «Анализ продаж»;
- формирование отчета «Заказы мототехники»;
- формирование отчета «Состояние заказов».

АИС «Учет мототехники в мотосалоне» должна обеспечивать выполнение следующих функций:

1) ввод справочных данных;

- 2) ввод сведений о поставщиках;
- 3) ввод сведений о покупателях;
- 4) ввод сведений об мототехники;
- 5) ввод сведений о заказах;
- 6) внесение изменений сведений о мототехники;
- 7) внесение изменений сведений о поставщиках;
- 8) формирование различных отчетов: прайс-лист мототехники, анализ продаж, реестр справок счетов, заказы мототехники, состояние заказов мототехники;
- 9) внесение сведений о проданной техники;
- 10) выписка счета на оплату.

Выводы. В работе были рассмотрены существующие системы автоматизации, проведен их сравнительный анализ, были выделены основные достоинства и недостатки автоматизированных систем. Также был проведен системный анализ мотосалона «МОТОДОМ Харьков», обозначены основные задачи, а также приведено описание бизнес процессов и этапов работы. Данные исследования, в дальнейшем, помогут разработать модель автоматизированной информационной системы, которая сможет повысить эффективность работы мотосалона, поможет улучшить качество обслуживания и, как следствие, увеличить прибыль «МОТОДОМ Харьков».

Список литературы

1. Спицнадель В. Н. Основы системного анализа: Учеб. пособие. — СПб.: «Изд. дом «Бизнес-пресса», 2000 г. — 326 с.
2. Лямец Владимир Ильич. Системный анализ. Вводный курс : Учебное пособие / Владимир Ильич Лямец, Андрей Дмитриевич Тевяшев . – 2-е изд., перераб. и доп. – Харків : Издательство ХНУРЭ, 2004 . – 448 с.
3. Злупко С. Принципи дослідження міждисциплінарних проблем : Згуровский М., Панкратова Н. Системный анализ: Проблемы, методология, приложения. — К.: Наук. думка, 2005. — 743 с. / С. Злупко // Вісник Національної академії наук України. - 2007. - № 2. - С. 86-88.
4. Саати Т., Кернс Л. Аналитическое планирование. Организация систем. - М.: Радио и связь, 1991. - 224 с.

УДК 65.011.56

**Разработка моделей и методов управления горным предприятием в
условиях экономических рисков**

Ванжа Т.В., студентка 5-го курса, Орлов Ю.К., канд. техн. наук, доц.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа

Email: sam_dntu@i.ua

Аннотация

Ванжа Т.В., Орлов Ю.К. Разработка моделей и методов управления горным предприятием в условиях экономических рисков. Предложено определение экономического риска предприятия, методы управления рисками, предложены этапы цикла принятия управленческих решений в сфере организации управления экономическими рисками. Для анализа использованы методы структурного анализа. Разрабатывается перечень антирисковых мероприятий и возможных альтернативных стратегий шахты.

Общая постановка проблемы. В условиях рыночных отношений проблема оценки и учета экономического риска приобретает самостоятельное теоретическое и прикладное значение как важная составная часть теории и практики управления.

Большинство управленческих решений принимается в условиях риска, что обусловлено рядом факторов – отсутствием полной информации, наличием противоборствующих тенденций, элементами случайности и многим другим.

На практике, чтобы принять экономическое решение, необходимо уметь оценивать, какие риски наиболее вероятны и к чему они могут привести в случае их реализации.

Схема процесса управления экономическим риском. Экономический риск – это возможность случайного возникновения нежелательных убытков, измеряемых в денежном выражении [1].

Объектом экономического риска выступает экономическая система, эффективность работы которой неизвестна.

Субъект риска – это физическое или юридическое лицо, имеющее разрешение на принятие решений по поводу объекта экономического риска.

К признакам экономического риска относят:

– денежное выражение потерь – возможность количественно измерить риски, и управление рисками с помощью инструментов, которые действуют на основании принципов экономической эффективности.

– нежелательность возникновения убытков – если убыток не мешает интересам субъекта, то он не будет нести в себе риск для данного субъекта.

- непредсказуемость результата – к рискованным ситуациям относятся исключительно такие, в которых невозможно предсказать появление потерь.
- вероятность возникновения убытка.

Управление экономическим риском включает в себя разработку и реализацию экономически обоснованных для предприятия рекомендаций и мероприятий, направленных на уменьшение стартового уровня риска до приемлемого финального уровня. Управление риском опирается на результаты оценки риска, технико-технологический и экономический анализ потенциала и среды функционирования предприятия, действующую и прогнозируемую нормативную базу хозяйствования, экономико-математические методы, маркетинговые и другие исследования.

Процесс управления риском представлен на рисунке 1.

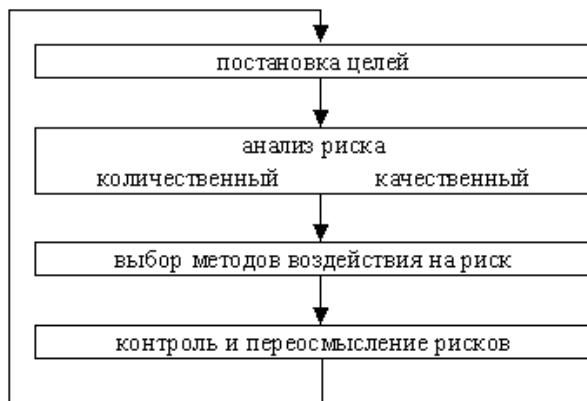


Рис. 1 – Процесс исследования риска

Объектом оценки и управления риском здесь являются стратегические решения предприятия, стратегический план, содержащий ряд утверждений относительно будущего развития внешней по отношению к предприятию среды, рекомендаций по образу действий руководства и коллектива предприятия, прогнозных высказываний о реакциях на планируемые стратегические мероприятия потребителей продукции, поставщиков сырья, конкурентов и др.

На каждом из этапов используются свои методы исследования рисков, каждый из них по отдельности дает результаты, являющиеся исходными данными и для последующих этапов, что требует объединения этих этапов в систему. Это позволит максимально эффективно добиваться целей, поставленных перед предприятием, поскольку информация, получаемая на каждом из этапов,

позволяет корректировать не только методы воздействия на риск, но и переосмысливать цели, ставящиеся перед организацией.

Эффективное управление риском требует не только внимательного наблюдения за размером риска, но также стратегию минимизации убытков. Для борьбы с рисками применяются известные методы управления, которые представлены на рисунке 2 [2].



Рис. 2 – Методы управления риском

Для управления риском на горных предприятиях создается специальное подразделение – отдел управления риском (ОУР), возглавляемое риск-менеджером, т.е. руководителем, который занимается исключительно проблемами управления риском и координирует действия всех подразделений в плане регулирования риска и обеспечения компенсации возможных потерь и убытков.

Важным этапом организации риск-менеджмента являются контроль за выполнением намеченной программы, анализ и оценка результатов выполнения выбранного варианта рискованного решения. При этом рекомендуется аккумулировать всю информацию об ошибках и недостатках разработки программы, проявившихся в ходе ее реализации. Такой подход позволит провести разработку последующих программ мероприятий по снижению рисков на более качественном уровне с использованием новых полученных знаний о риске [3].

Результаты каждого этапа становятся исходными данными для последующих этапов, образуя систему принятия решений с обратной связью. Такая система обеспечивает максимально эффективное достижение целей, поскольку знание, получаемое на каждом из этапов, позволяет корректировать не только методы воздействия на риск, но и сами цели управления рисками [5]. Этапы цикла принятия управленческих решений в сфере организации управления экономическими рисками представлены на рисунке 3.



Рис. 3 – Этапы цикла принятия управленческих решений в сфере организации управления экономическими рисками

Статистический метод определения уровня риска. Суть его заключается в том, что для расчета возможности возникновения потерь анализируются статистические данные о результативности принимаемых предприятием хозяйственных решений [4].

Условно выделяются следующие зоны риска:

- безрисковая область (гарантия получения расчетной прибыли);
- область минимального риска (потери не могут превысить размеров чистой прибыли);
- область повышенного риска (риск убытков);
- область критического риска (возможна частичная утрата активов предприятия);
- область недопустимого риска (неликвидный баланс, неплатежеспособность).

Для получения максимального уровня риска предприятия по всем или одному направлению деятельности используется статистическая информация о хозяйственных операциях, которые осуществлялись за несколько последних лет [5,6]. Информация представлена в таблице 1.

Таблица 1 – Определение уровня риска

Предшествующие периоды	Частота возникновения потерь (f, %)				
	Безрисковая область	Область минимального риска	Область повышенного риска	Область критического риска	Область недопустимого риска
	$\square_{\text{общ}}$	\square_2	\square_3	\square_4	\square_5
1	0,75	0,32 (42%)	0,33(44%)	0,05 (7%)	0 05 (7%)
2	0,80	0,35(45%)	0,20 (25%)	0,20(25%)	0,05 (5%)
3	0,85	0,005(5%)	0,17(20%)	0,25(30%)	0,38 (45%)

Частота возникновения потерь определяется по формуле (1) [7]:

$$f = \frac{n}{n_{\text{общ}}}, \quad (1)$$

где n – число случаев наступления конкретного уровня потерь;

$n_{\text{общ}}$ – общее число случаев в статистической выборке, включая успешные пред-принимательские риски.

Уровень риска $U_{p,\text{max}}$ определяется по частоте возникновения потерь для чего строится график кумулятивных итогов представленный на рисунке 4.

За последний год кумулятивные итоги означают, что частота потерь в области II - 5%, III - 25%, IV - 55% , V - 100%.

Кривая AbC является кривой Лоренца, характеризующей уровень риска. Чем больше отклонение кривой Лоренца от прямой AC (безрисковой прямой), тем выше уровень риска [8]. Количественно этот показатель может быть определен формулой (2).

$$y_p = \left(1 - \frac{ab}{AC}\right). \quad (2)$$

где ab – отклонение кривой Лоренца;
 AC – безрисковая прямая.

Линия Лоренца построена по нескольким периодам, что позволяет провести некоторые обобщения по поводу допустимости уровня риска для конкретных условий хозяйствования.

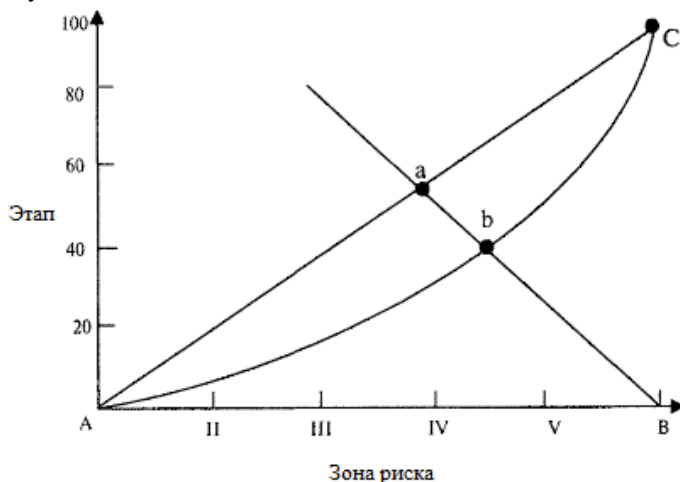


Рис.4 – Определение уровня риска с помощью графика Лоренца

Выводы. Стратегические и управленческие решения определяются характером решаемых задач, внутренними и внешними условиями производства. Отсутствие научно обоснованных подходов к формированию экономической стратегии и стратегических управленческих решений приводит к недостаточно эффективной направленности годовых и перспективных планов предприятий, что не позволяет стабилизировать работу шахт и создать условия для их развития.

Список литературы

1. Гранатуров В.М. Экономический риск: сущность, методы измерения, пути снижения: Учебное пособие. – М.: Изд-во «Дело и Сервис», 1999. – 112 с.
2. Методы анализа рисков. [Интернет-ресурс]. – Режим доступа: <http://www.risk24.ru/metodianaaliza.htm>.
3. Ванжа Т.В. Разработка Моделей И Методов Управления Горным Предприятием В Условиях Экономических Рисков. Мир компьютерных технологий: Сборник статей студенческой научно- технической конференции, г.Севастополь, 04 – 08 апреля 2016 г/ М-во образования и науки РФ,

Севастопольский государственный университет; науч. ред. Е.Н. Машенко – г. Севастополь: СевГУ, 2016. – 173-177 с.

4. Вяткин В.Н., Гамза В.А., Екатеринославский Ю.Ю., Хэмптон Дж., Дж. Управление риском в рыночной экономике. – М.: Экономика, 2002. – 195 с.

5. Балабанов И.Т. Риск-менеджмент. – М.: Финансы и статистика, 1996. – 135 с.

6. Мамаева, Л.Н. Управление рисками: Учебное пособие / Л.Н. Мамаева. – М.: Дашков и К, 2013. – 256 с.

7. Соложенцев Е. Д. Сценарное логико-вероятностное управление риском в бизнесе и технике: изд. 2-е. — СПб.: Издательский дом «Бизнес-пресса», 2006. — 530 с.

8. Леонова О.А. Принципы и подходы к оценке рисков горнодобывающего предприятия // Известия вузов Северо-Кавказского региона, Технические науки – Ростов-на-Дону, 2003. – Приложение 2. – С. 107 – 110.

УДК 004.91

Разработка моделей и методов синтеза информационно-справочной системы диагностирования и лечения болезней для медицинских звений Бровкина А.С., студентка 5-го курса, Орлов Ю.К., канд. техн. наук, доц.

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
E-mail: annbrunya@gmail.com

Аннотация

Бровкина А.С., Орлов Ю.К. Разработка моделей и методов синтеза информационно-справочной системы диагностирования и лечения болезней для медицинских заведений. Рассмотрена классификация медицинских информационных систем. Определены требования к созданию системы хранения и использования медицинской информации. Определены основные направления развития и автоматизации современных информационно-справочных систем для использования медицинских заведений.

Общая постановка проблемы. В настоящее время информатизация охватывает всё более широкие сферы человеческой деятельности. При этом темпы роста цифровой информации значительно опережают темпы роста информации на бумажных носителях.

Данная работа будет направлена на разработку, анализ и усовершенствование методов проектирования комплексной медицинской

информационной экспертной системы с использованием математического моделирования процессов её функционирования.

Исследования. Существующие информационно-справочные системы не в полной мере удовлетворяют всем потребностям медицинских учреждений. За прошедшие 30 лет разработано большое количество сложных медицинских систем: лабораторные, радиологические и др. Предыдущие попытки решить проблемы здравоохранения в части своевременного информационного обеспечения или введения статистических программ вовлекали различные формы автоматизации. Некоторые из этих попыток были в виде телефонной библиотеки ответов на медицинские вопросы. Другие попытки были направлены на обеспечение врачей автоматизированными средствами помощи для использования при обследовании пациента. В последнее же время основные усилия направлены на то, чтобы «научить» системы обмениваться между собой информацией и обеспечить комплексный подход к лечению пациента.

Современные медицинские организации производят и накапливают огромные объемы данных. От того, насколько эффективно эта информация используется врачами, руководителями, управляющими органами, зависит качество медицинской помощи, общий уровень жизни населения, уровень развития страны в целом и каждого ее территориального субъекта в частности. Поэтому необходимость использования больших, и при этом еще постоянно растущих, объемов информации при решении диагностических, терапевтических, статистических, управленческих и других задач, обуславливает сегодня создание информационных систем в медицинских учреждениях. Системы, основной функцией которых является накопление данных и информации (необходимость накопления больших объемов профессионально ценной информации – одна из проблем, с которой имеют дело врачи). Компьютер выступает как средство надежного хранения профессиональных знаний, обеспечивающее доступный и быстрый поиск необходимых сведений. Помимо этого, в память компьютера может быть заложена информация, классифицированная по объектовому признаку (материально-техническая база, лекарственные средства), по видам информации (экономическая, научная, нормативно-правовая и др.), или же по ее характеру (первичная, вторичная, оперативная, обзорно-аналитическая, экспертная, прогностическая и). Хранение, обработка и использование информации является основой создания информационной среды. В медицинских информационных системах (МИС) имеются диагностические и консультирующие системы (проблема принятия решений в сложных и спорных ситуациях). Машина-эксперт позволяет рассмотреть все возникшие варианты и поставить более точный диагноз. Системы, обеспечивающие медицинское обслуживание (используя современные вычислительные средства, определяются косвенными методами важные физиологические функции, такие

как сердечный индекс, плотность печени, объем и положение кисты и др.). К таким приборам относятся эхокардиографы, гамма-камеры. Возможные погрешности измерений (вычислений) этих функций – приблизительно 15 % данных, получаемых с помощью любой из мониторинго-компьютерных систем.

Медицинские информационно-справочные системы предназначены для комплексной или частичной автоматизации работы амбулаторно-поликлинического и стационарного лечебно-профилактического учреждений и направлены на решение актуальных задач, таких как:

- уменьшение времени обслуживания пациентов в регистратуре,
- автоматизированная запись на прием к специалистам и в лабораторное отделение,
- равномерное распределение пациентов по приемным дням специалиста,
- формирование экономической и статистической отчетности в системах ОМС и ДМС, а также внутренней отчетности лечебного учреждения,
- повышение скорости формирования медицинской документации,
- персонифицированный учет расходования медикаментов,
- автоматизированный ввод схем прививок и учет выполненных работ, связанных с иммунопрофилактикой населения,
- проведение внутренней экспертизы качества оказания медицинской помощи.

С внедрением системы значительно возрастёт уровень сервиса, в том числе доступный объем справочной информации для пациентов. Поиск информации или информационный поиск (одна из реализуемых задач в информационно - справочной системе) представляет один из основных информационных процессов. Человечество издревле занималось им. Цели, возможности и характер поиска всегда зависели от наличия, информации, её важности и доступности, а также средств организации поиска. Интеграция программного обеспечения с информационными киосками и телевизионными панелями в помещениях медучреждений, управляющих организаций и социальных служб, позволяет мгновенно обновлять и доносить до граждан интересующие их открытые данные и информационные сообщения. Такие, как расписание приема врачей, график работы диагностических кабинетов и более специальную информацию, которая может быть представлена по конкретному запросу без отвлечения работников регистратуры или специалистов.

Современные системы помогают отслеживать себестоимость медицинских услуг, анализируя такие ключевые параметры, как время специалистов, применение лекарственных средств и материалов, загрузку диагностического и лабораторного оборудования. Через информационную систему администрация обретает контроль над своими финансами практически в

режиме реального времени. На основании подробных данных управленческого учета и внутренней статистики применяются мотивирующие схемы оплаты врачей в зависимости от персональных показателей каждого специалиста. Принятие управленческих решений требует гораздо меньше времени, так как необходимая для этого информация доступна руководителю не время от времени, а постоянно.

Учитывая это, многие отечественные лечебно-профилактические учреждения (ЛПУ) в своей деятельности активно прибегают к услугам комплексных медицинских информационных систем (МИС). Последние представляют собой целостный (универсальный) программный продукт, позволяющий на качественно новом уровне осуществлять руководство деятельностью медучреждения и оказывать медицинские услуги.

Комплексная МИС, например, существенно упрощает взаиморасчеты с клиентами. При посещении коммерческих клиник пациентам удобно иметь выбор между разными формами оплаты за полученные услуги. Актуальна задача взаиморасчетов со страховыми компаниями, как в рамках обязательного, так и добровольного медицинского страхования. Данная функция особенно важна в связи с предполагаемым переходом к преимущественно страховой модели финансирования здравоохранения на общегосударственном уровне.

Непосредственно процедура приема пациентов также претерпевает ряд положительных изменений. Врачам уже не нужно так много писать от руки. Система создает им простые и эффективные средства для регистрации результатов осмотра и формирования направлений. Врач может прямо на приеме, если имеется такая необходимость, минуя регистратуру, записать пациента к другому врачу.

С появлением единой базы данных гарантирована сохранность медицинской карты пациента и истории болезни. В отличие от традиционной бумажной карты электронную медицинскую карту (ЭМК) пациент не может унести с собой, потерять или уничтожить. А государственные стандарты в сфере информатизации, соблюдение которых является нормой для разработчиков промышленных медицинских систем, позволяют существенно повысить безопасность персональных данных. При помощи электронной очереди и равномерного распределения пациентов по приемным дням специалистов приведет к полному исчезновению очередей.

Вопреки распространенным заблуждениям и слухам медицинский персонал также выигрывает от использования информационной системы.

Разработка систем, основанных на знаниях, является составной частью исследований по ИИ, и создает компьютерные методы решения проблем, обычно требующих привлечения экспертов-специалистов.

Вопрос о том, как помочь пользователю понять структуру и функции некоторого сложного компонента программы, связан со сравнительно новой

областью взаимодействия человека и машины, которая появилась на пересечении таких областей, как искусственный интеллект, промышленная технология, физиология и эргономика. На сегодня вклад в эту область исследователей, занимающихся экспертными системами, состоит в разработке методов представления информации о поведении программы в процессе формирования цепочки логических заключений при поиске решения.

Автоматизированная система моделирования (АСМ) – компьютерная система, предназначенная для оказания помощи пользователю по представлению нужной ему задачи в виде определённой математической схемы, принятой в данной системе, решить задачу (провести моделирование по полученной схеме) и проанализировать результаты. АСМ состоит из трёх основных компонентов: функциональное наполнение, язык заданий и системное наполнение.

Функциональное наполнение является совокупностью конструктивных элементов (модулей), из которых составляется схема (модель).

Системное наполнение – это набор программ, отражающих специфику реализации АСМ и обеспечивающих собственно функционирование системы: трансляцию и исполнение заданий, поддержку базы знаний о предметной области и т.д. Например, поисковая система (ПС). Язык заданий (ЯЗ) служит для описания задач, вводимых в систему.

ПС с большим набором функций и возможностей обычно входят в состав СУБД и именуются информационно-поисковыми системами. Они также создаются и используются для эффективного нахождения пользователями необходимых им данных, в том числе в Интернете.

Информационно-поисковая система (ИПС) трактуется и как система, обеспечивающая поиск и отбор необходимых данных на основе информационно-поискового языка и соответствующих правил поиска, а база данных – как совокупность средств и методов описания, хранения и манипулирования данными, облегчающих сбор, накопление и обработку больших информационных массивов. Организация различных БД отличается видом объектов данных и отношений между ними.

Функционирование современных информационно-поисковых систем основано на том, что:

- 1) документы, необходимые пользователю, объединены наличием некоторого признака или комбинации признаков;
- 2) пользователь способен указать этот признак.

Современная концепция информационных систем предполагает объединение электронных записей о больных (electronic patient records) с архивами медицинских изображений и финансовой информацией, данными мониторинга с медицинских приборов, результатами работы автоматизированных лабораторий и следящих систем, наличие современных

средств обмена информацией (электронной внутрибольничной почты, Internet, видеоконференций и т.д.).

Выводы. Таким образом, особенностью МИС является автоматизированный поиск и справочное (семантическое) сопровождение по вводимому термину при регистрации, дифференциальном сопоставлении и сопровождении клинических или других данных. Система обеспечивает информационно-интеллектуальное сопровождение лечебно-диагностического процесса.

На сегодняшний день интерфейс работы информационной системы настолько прост, что для пользования системой не требуются специальные знания и навыки. Каждый сотрудник офиса или предприятия в течении небольшого времени может полностью изучить работу нужной информационно-справочной системы.

Список литературы

1. Материалы по информационным технологиям [Электронный ресурс]. – Режим доступа: <http://inftis.narod.ru/ais/ais-n9.htm>
2. Материалы по информационным технологиям [Электронный ресурс]. – Режим доступа: <http://daxnow.narod.ru/index/0-17>
3. Citforum [Электронный ресурс]. – Режим доступа: <http://citforum.ru/consulting/articles/med/>
4. Медицинская энциклопедия [Электронный ресурс]. – Режим доступа: <http://www.medical-enc.ru/>
5. Материалы по информационным технологиям [Электронный ресурс]. – Режим доступа: <http://daxnow.narod.ru/index/0-10>

УДК 005+65.011.56

Формализация задачи принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости

Милая А.С., Савкова Е.О.

Донецкий национальный технический университет

E-mail: milaya.don@gmail.com

Аннотация

Милая А.С., Савкова Е.О. Формализация задачи принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости. В статье рассматривается формализация задачи принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости на примере организации «Корпорация ПАРУС», предоставляющей консультационные услуги в сфере информации организациям Донецкой и Луганской областей. Для решения формализованной задачи к ней применяются различные критерии выбора оптимальной альтернативы и исследуются результаты, полученные с помощью этих критериев.

Общая постановка проблемы. В настоящее время руководителю необходимо владеть современными технологиями принятия управленческих решений, поэтому возникла необходимость формализации реальных задач принятия решения. Современная наука в этой области поднялась на качественно новый уровень, на ее основе разработаны эффективные управленческие технологии, компьютерные системы поддержки принятия решений, экспертные системы, автоматизированные системы экспертного оценивания, предназначенные осуществлять в рамках подготовки к принятию решений значительные объемы экономических, математических, логических и других видов расчетов, и позволяющие решать сложные управленческие задачи, характерные для современных организаций.

Целью данной статьи является формализация реально существующей задачи принятия решения в условиях неопределённости. То есть, приведение её к такому виду, чтобы для поиска оптимального решения можно было применить уже существующие научные методы или комбинации этих методов.

Задачи:

- Дать определение основным понятиям;
- Описать процесс формализации задачи;
- Формализовать задачу принятия решения в условиях неопределённости;
- Описать методы, которые можно применить для решения данной задачи.

Актуальность статьи обусловлена ростом потребности руководителей как можно более эффективно распределить объём работ на определённый период времени существования организации.

Постановка задачи. Для формализации каждой конкретной оптимизационной задачи при анализе системы управления в условиях неопределённости необходимо реализовать следующие процедуры. [1]

Определить множество $\{\theta_1, \theta_2, \dots, \theta_n\}$ всех возможных внешних ситуаций (не зависящих от ЛПР), которые влияют на экономический результат соответствующих решений в рамках анализируемого проекта (задачи). Указанный набор ситуаций $\{\theta_j, j=\overline{1, n}\}$ должен представлять собой полную группу событий. Что означает обязательное выполнение следующих двух условий:

$$1.1. \quad \forall (k, l) \quad \theta_k \cap \theta_l = \emptyset$$

(т. е. одновременное наступление двух событий такой полной группы невозможно)

$$1.2. \quad \bigcup_{j=1}^n \theta_j = \Omega$$

(т. е. одно из событий полной группы наступит обязательно)

Здесь Ω обозначает пространство всех элементарных исходов. Вероятности $q_j = P\{\theta_j\}$ для случайных событий соответствующей полной группе неизвестны. Вопрос о том, рассматривать или нет в формате соответствующей модели конкретное случайное событие, решает непосредственно ЛПР.

2. Составить перечень $\{X_1, X_2, \dots, X_m\}$ всех альтернативных решений, которые требуется анализировать, и для которых экономический результат будет зависеть от реализованной «внешней» ситуации, т. е. будет зависеть от того, какое из событий полной группы $\{\theta_j, j=\overline{1, n}\}$ (выделенной на предыдущем шаге) наступит.

3. Определить ожидаемые доходы a_{ij} для случаев, когда будет принято решение X_i (из множества указанных выше анализируемых альтернатив), а внешняя, не зависящая от ЛПР ситуация сложится такая, которая соответствует событию θ_j (из множества событий полной группы, влияющей на экономический результат). Они оформляются в виде матрицы $A = (a_{ij})$, которую в теории называют матрицей полезностей. Структура матрицы полезностей следующая:

$$A = \begin{matrix} & \theta_1 & \theta_2 & \theta_3 & \dots & \theta_n \\ \begin{matrix} X_1 \\ X_2 \\ \dots \\ X_m \end{matrix} & \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix} \end{matrix}$$

(как видно, элемент a_{ij} стоит на пересечении i -той строки, которая соотносится с решением X_i , и j -того столбца, который соотносится с внешней ситуацией θ_j).

4. Далее для представленной таким образом задачи требуется из рассматриваемого множества альтернативных решений $\{X_i, i = \overline{1, m}\}$ выбрать одну альтернативу (наилучшую для ЛПР). Далее в статье будут рассмотрены критерии, которые можно использовать для реализации наилучшего выбора при оптимизации систем управления.

Формализация задачи. Итак, задача принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости состоит в следующем: ООО «Корпорация ПАРУС» предоставляет консультационные услуги в сфере информации организациям Донецкой и Луганской областей. Руководителю необходимо принять решение о распределении работ по обслуживанию шести организаций:

1. управление физкультуры и спорта (государственная организация);
2. министерство здравоохранения (государственная организация);
3. министерство образования (государственная организация);
4. Енакиевский коксохимический завод;
5. Ясиновский коксохимический завод;
6. Донецкий металлургический завод (ДМЗ).

Желаемое время обслуживания государственных организаций – 30 часов, коксохимических заводов – 20 часов, а ДМЗ – 70 часов в месяц. Корпорация может выделить на обслуживание этих предприятий не более 140 часов в месяц. При этом обязательное время, которое корпорация должна выделить для обслуживания ДМЗ – 30 часов, а всех остальных организаций – 10 часов, т. е. за месяц каждую организацию необходимо обслужить не менее заданного времени.

Оплата за час обслуживания составляет: для государственных организаций – 400 рублей, для Енакиевского коксохимического завода – 480 рублей, для Ясиновского – 320 рублей и для ДМЗ – 350 рублей. Вначале месяца каждой организацией вносится предоплата в размере 25% от стоимости работы, которую планируется выполнить. Работа, за которую внесена предоплата, будет выполнена в любом случае, вне зависимости от неопределенных ситуаций.

Руководителю необходимо распределить время обслуживания таким образом, чтобы получить максимальный доход.

Факторами, влияющими на конечный результат, являются:

1. факт выполнения работы (выполнена/не выполнена);
2. причины того, что работа не выполнена: либо по вине исполнителя, либо по вине заказчика, либо из-за форс-мажорных обстоятельств (пожар, наводнение, землетрясение);
3. факт оплаты выполненной работы (оплачена/не оплачена).

Лицом, принимающим решение, в данной задаче является руководитель. Вышеуказанные факторы считаются неопределенными ситуациями, так как ЛПР

не может на них повлиять или с определенной точностью предсказать, какая из ситуаций произойдет.

Выполним формализацию задачи согласно ранее описанным этапам формализации:

1. Определяем множество θ всех возможных внешних ситуаций, не зависящих от ЛПР.

Так как причины невыполнения работ могут возникнуть только в случае, если работа не будет выполнена, а выполненная работа может быть либо оплачена, либо не оплачена, имеем пять неопределенных ситуаций:

θ_1 – работа не выполнена по вине исполнителя (в этом случае объем услуг увеличивается на 25%, а заказчик оплачивает 22% всей работы);

θ_2 – работа не выполнена по вине заказчика, клиента (в этом случае заказчик платит штраф в размере 22% от стоимости невыполненной работы);

θ_3 – работа не выполнена из-за форс-мажорных обстоятельств (в этом случае работа не оплачивается заказчиком);

θ_4 – работа выполнена и оплачена;

θ_5 – работа выполнена и не оплачена (заказчик платит штраф в размере 20% от стоимости всей работы, а оплата всей работы переносится на следующий месяц).

Таким образом, имеем набор ситуаций, образующих полную группу событий, $\{\theta_j, j = \overline{1,5}\}$.

2. Составляем перечень всех альтернативных решений, которые требуется анализировать, при этом суммарное время обслуживания всех организаций по каждой альтернативе не должно превышать 140 часов.

X_1 – обслуживать все организации минимально требуемое время: государственные организации и коксохимические заводы – по 10 часов, а ДМЗ – 30 часов

$$(10+10+10+10+10+30=80 \leq 140);$$

X_2 – обслуживать все организации половину желаемого времени: государственные организации – по 15 часов, коксохимические заводы – по 10 часов, а ДМЗ – 45 часов

$$(15+15+15+10+10+45=110 \leq 140);$$

X_3 – обслуживать государственные организации желаемое время, а остальные – минимально требуемое время: государственные организации – по 30 часов, коксохимические заводы – по 10 часов, а ДМЗ – 30 часов

$$(30+30+30+10+10+30=140 \leq 140);$$

X_4 – обслуживать государственные организации минимально требуемое время, а остальные – желаемое время: государственные организации – по 10 часов, коксохимические заводы – по 20 часов, а ДМЗ – 70 часов

$$(10+10+10+20+20+70=140 \leq 140);$$

Таким образом имеем множество альтернативных решений $\{X_i, i = \overline{1, 4}\}$, для которых экономический результат будет зависеть от реализованной «внешней» ситуации, т. е. от того, какое из событий определенной на предыдущем этапе полной группы наступит.

3. Определим ожидаемые доходы $\{a_{ij}, i = \overline{1, 4}, j = \overline{1, 5}\}$, учитывая все вышеопределенные факторы.

Для начала построим матрицу полезностей для текущей задачи:

$$A = \begin{matrix} & \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 \\ \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{matrix} & \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{pmatrix} \end{matrix}$$

Далее рассчитаем каждый элемент матрицы a_{ij} , учитывая все ранее определенные факторы.

Заполняем первую строку матрицы полезностей. Для этого посчитаем обязательную предоплату в размере 25% за запланированную работу для первой альтернативы:

$$((10*400)*3+10*480+10*320+30*350)*0.25=5\ 875 \text{ рублей.}$$

Далее к этой сумме будем прибавлять или отнимать от нее результаты, связанные с неопределенными ситуациями.

Рассчитываем элементы строки.

- a_{11} – пересечение альтернативы X_1 (все организации обслуживаются минимально требуемое время) и ситуации θ_1 , то есть, при этом работа не выполнена и её объём увеличивается на 25%. То есть, рассчитываемая сумма – это 22% от стоимости 125% работы плюс предоплата. Таким образом имеем:

$$a_{11}=5\ 875+((10+10*0.25)*400*3+(10+10*0.25)*480+(10+10*0.25)*320+(30+30*0.25)*350)*0.22=5\ 875+8\ 387.5=14\ 262.5 \text{ (рублей).}$$

- a_{12} – пересечение альтернативы X_1 и ситуации θ_2 , то есть, при этом работа не выполнена и заказчик платит штраф в размере 22% от стоимости невыполненной работы. То есть, так как работа, за которую внесена предоплата, будет выполнена в любом случае, рассчитываемая сумма – это 22% от стоимости 75% работы плюс предоплата. Таким образом имеем:

$$a_{12}=5\ 875+((10-10*0.25)*400*3+(10-10*0.25)*480+(10-10*0.25)*320+(30-30*0.25)*350)*0.22=5\ 875+5\ 032.5=10\ 907.5 \text{ (рублей).}$$

- a_{13} – пересечение альтернативы X_1 и ситуации θ_3 , то есть, при этом работа не выполнена, а окончательная сумма равна предоплате:

$a_{13}=5\ 875$ (рублей).

▪ a_{14} – пересечение альтернативы X_1 и ситуации θ_4 , то есть, при этом работа выполнена и оплачена, и рассчитываемая сумма – это стоимость 100% работы. Таким образом имеем:

$a_{14}=10*400*3+10*480+10*320+30*350=23\ 500$ рублей.

▪ a_{15} – пересечение альтернативы X_1 и ситуации θ_5 , то есть, работа выполнена, но не оплачена, и заказчик платит штраф в размере 20% от стоимости всей работы. То есть, рассчитываемая сумма – это 20% от стоимости 100% работы, плюс предоплата. Таким образом имеем:

$a_{15}=5\ 875+(10*400*3+10*480+10*320+30*350)*0.2=5\ 875+6\ 100=11\ 975$ (рублей).

Аналогично рассчитываем элементы остальных строк матрицы. В итоге получаем окончательную матрицу полезностей:

	θ_1	θ_2	θ_3	θ_4	θ_5
X_1	14 262.5	10 907.5	5 875	23 500	11 975
X_2	21 918.75	17 326.25	10 437.5	41 750	18 787.5
X_3	28 612.5	22 617.5	13 625	54 500	24 525
X_4	27 562.5	21 787.5	13 125	52 500	23 625

Далее рассмотрим критерии, с помощью которых можно решить задачу. То есть, выбрать наилучшую альтернативу.

Решение задачи с помощью классических критериев

Классическими критериями принятия решений в условиях неопределённости считаются следующие [2]:

- максиминный критерий;
- оптимистический критерий;
- нейтральный критерий;
- критерий Сэвиджа.

Для каждого критерия приведем алгоритм решения и применим его к вышеописанной задаче.

Максиминный критерий (ММ-критерий или критерий Вальда). Этот критерий характеризуется крайней осторожной или пессимистической позицией отношения ЛППР к неопределённости экономического результата.

Целевая функция критерия:

$$Z_{MM} = \max_i \{ \min_j a_{ij} \}$$

Оптимистический критерий (Н-критерий). Этот критерий характеризуется крайней оптимистической позицией отношения ЛППР к неопределённости экономического результата, то есть, позицией «азартного игрока», уверенного в том, что ему должно повезти, и поэтому склонного к самым рискованным выборам.

Целевая функция критерия:

$$Z_H = \max_i \{ \max_j a_{ij} \}$$

Нейтральный критерий (N-критерий). Этот критерий характеризуется нейтральной или средневзвешенной позицией отношения ЛПР к возможным значениям конечного экономического результата при случайных ситуациях, описываемых полной группой событий.

Целевая функция критерия:

$$Z_N = \max_i \left\{ \frac{1}{n} * \sum_{j=1}^n a_{ij} \right\}$$

Применим вышеописанные критерии к задаче. Для N-критерия $n=5$, так как в задаче всего 5 случайных ситуаций, описываемых полной группой событий.

Таким образом, получаем:

Таблица 2 – Классические критерии

	θ_1	θ_2	θ_3	θ_4	θ_5	Z_{MM}	Z_H	Z_N
X ₁	14 262.5	10 907.5	5 875	23 500	11 975	5 875	23 500	13 304
X ₂	21 918.75	17 326.25	10 437.5	41 750	18 787.5	10 437.5	41 750	22 044
X ₃	28 612.5	22 617.5	13 625	54 500	24 525	13 625	54 500	28 776
X ₄	27 562.5	21 787.5	13 125	52 500	23 625	13 125	52 500	27 720

Согласно этим критериям оптимальной является альтернатива X₃, то есть, руководителю (ЛПР) стоит принять решение обслуживать государственные организации желаемое время, а остальные – минимально требуемое время: государственные организации – по 30 часов, коксохимические заводы – по 10 часов, а ДМЗ – 30 часов.

Критерий Сэвиджа (S-критерий). Этот критерий характеризуется крайней осторожной (пессимистической) позицией отношения ЛПР к возможным потерям из-за отсутствия достоверных сведений о том, какая из ситуаций, влияющих на экономический результат, будет иметь место в конкретном случае.

Целевая функция критерия:

$$Z_S = \min_i \{ \max_j l_{ij} \},$$

где $l_{ij} = \max_i \{ a_{ij} \} - a_{ij}$ (элементы новой матрицы потерь L)

Применим критерий к задаче.

Построим матрицу потерь. Для этого в каждой строке выберем максимальный элемент:

Таблица 3 – Критерий Сэвиджа. Максимальные элементы строк

	θ_1	θ_2	θ_3	θ_4	θ_5
X ₁	14 262.5	10 907.5	5 875	23 500	11 975
X ₂	21 918.75	17 326.25	10 437.5	41 750	18 787.5
X ₃	28 612.5	22 617.5	13 625	54 500	24 525
X ₄	27 562.5	21 787.5	13 125	52 500	23 625

Далее отнимем каждый элемент строки от её максимального элемента. Получаем матрицу потерь, к которой применяем вышеописанный алгоритм действий для определения лучшей альтернативы:

Таблица 4 – Критерий Сэвиджа. Матрица потерь

	θ_1	θ_2	θ_3	θ_4	θ_5	Z_S
X ₁	14 350	11 710	7 750	31 000	12 550	31 000
X ₂	6 693.75	5 291.25	3 187.5	12 750	5 737.5	12 750
X ₃	0	0	0	0	0	0
X ₄	1 050	830	500	2 000	900	2 000

Согласно этому критерию оптимальной является альтернатива X₃.

Модификация максиминного критерия (MM_{mod}-критерий). Этот критерий, как и представленные выше MM- и S-критерии, характеризуется также весьма осторожной, пессимистической позицией отношения ЛППР.

Целевая функция критерия:

$$Z_{MM_{mod}} = \max_i \{\min_j (\hat{a}_{ij})\},$$

где (\hat{a}_{ij}) – модифицированная следующим образом матрица полезностей: к каждому элементу любого отдельного столбца матрицы полезностей добавляется одно и то же число, зависящее от столбца $\Delta_j = \max_i \{\max_j a_{ij}\} - \max_i (a_{ij})$.

Применим критерий к задаче.

Построим модифицированную матрицу. Для этого выберем максимальные элементы в столбцах:

Таблица 5 – MM_{mod}-критерий. Максимальные элементы по столбцам

	θ_1	θ_2	θ_3	θ_4	θ_5
X ₁	14 262.5	10 907.5	5 875	23 500	11 975
X ₂	21 918.75	17 326.25	10 437.5	41 750	18 787.5
X ₃	28 612.5	22 617.5	13 625	<u>54 500</u>	24 525
X ₄	27 562.5	21 787.5	13 125	52 500	23 625

Далее рассчитаем «добавки» для каждого столбца (отнимем каждый максимальный элемент столбца от максимального элемента матрицы):

$$\Delta_1 = 25\,887.5; \Delta_2 = 31\,882.5; \Delta_3 = 40\,875; \Delta_4 = 0; \Delta_5 = 29\,975.$$

Получим модифицированную матрицу, сложив каждый элемент столбца с соответствующей «добавкой» и применим к полученной матрице вышеописанный алгоритм действий для определения лучшей альтернативы:

Таблица 6 – MM_{mod} -критерий. Модифицированная матрица

	θ_1	θ_2	θ_3	θ_4	θ_5	$Z_{MM_{mod}}$
X ₁	40 150	42 790	46 753	23 500	41 950	23 500
X ₂	47 806.25	49 208.75	51 312.5	41 750	48 762.5	41 750
X ₃	54 500	54 500	54 500	54 500	54 500	54 500
X ₄	53 450	53 670	54 000	52 500	53 600	52 500
Δ	25 887.5	31 882.5	40 875	0	29 975	

Согласно этому критерию оптимальной является альтернатива X₃.

Решение задачи с помощью производных критериев

К производным относят критерии, которые модифицируют или обобщают классические критерии:

- критерий Гурвица;
- критерий произведения;
- критерий Гермейера и его модификация;
- критерий наиболее вероятного исхода.

Для каждого критерия приведем алгоритм решения и применим его к вышеописанной задаче. [5]

Критерий Гурвица (НВ-критерий). Этот критерий характеризуется взвешенной позицией “пессимизма-оптимизма”, отражающей отношение ЛПР к неопределённости экономического результата.

Целевая функция критерия:

$$Z_{HW} = \max_i \{c * \min_j \{a_{ij}\} + (1 - c) * \max_j \{a_{ij}\}\}.$$

где c ($0 \leq c \leq 1$) – «вес», с которым учитывается оценка классического ММ-критерия, а $(1 - c)$ – «вес», с которым учитывается оценка классического H -критерия.

Критерий произведения (P-критерий). Этот критерий характеризуется менее пессимистической позицией отношения ЛПП к неопределённости экономического результата, чем, например, при ММ-критерии, но более пессимистической, чем при N -критерии.

Целевая функция критерия:

$$Z_P = \max_i \left\{ \sqrt[n]{\prod_{j=1}^n a_{ij}} \right\}, \text{ причём } (a_{ij} > 0).$$

В случае, если хотя бы один элемент матрицы меньше или равен нулю, используют преобразование всех элементов матрицы полезностей к виду $a_{ij} + a$, ($a > 0$).

Применим критерий к задаче. В нашем случае $n=5$. тогда матрица с дополнительным столбцом будет иметь вид:

Критерий наиболее вероятного исхода. Особенность использования этого критерия для выбора наилучшего решения при оптимизации соответствующего звена/звеньев цепи поставок обуславливается следующим. В конкретной ситуации ЛПП может оказаться уверенным в том, что среди всех случайных событий полной группы $\{\theta_j, j = \overline{1,5}\}$ имеется именно одно такое событие j^* , которое является настолько вероятным, что ЛПП хочет и может, практически не сомневаясь, ориентировать свой выбор применительно к соответствующей ситуации θ_{j^*} .

Целевая функция критерия:

$$Z = \max_i \{a_{ij^*}\}$$

Применим вышеописанные критерии к задаче.

Для критерия наиболее вероятного исхода, какое бы событие не было принято ЛПП за наиболее вероятное, результат будет одинаков – альтернатива X_3 , так как в каждом столбце (при каждом событии полной группы) максимальный элемент соответствует именно этой альтернативе. Для критерия Гурвица примем $c=0,4$:

Таблица 7 – Производные критерии

	θ_1	θ_2	θ_3	θ_4	θ_5	Z_{MM}	Z_H	Z_{HW}	Z_P
X_1	14 262.5	10 907.5	5 875	23 500	11 975	5 875	23 500	$0.4*5\ 875+$ $+0.6*23\ 500=16\ 450$	12 079.65
X_2	21 918.75	17 326.25	10 437.5	41 750	18 787.5	10 437.5	41 750	$0.4*10\ 437.5+$ $+0.6*41\ 750=29\ 225$	19 885.13
X_3	28 612.5	22 617.5	13 625	54 500	24 525	13 625	54 500	$0.4*13\ 625+$ $+0.6*54\ 500=38\ 150$	25 957.84
X_4	27 562.5	21 787.5	13 125	52 500	23 625	13 125	52 500	$0.4*13\ 125+$ $+0.6*52\ 500=36\ 750$	25 005.26

Согласно этому критерию оптимальной является альтернатива X_3 .

Критерий Гермейера (G-критерий) и модифицированный $G(mod)$ -критерий Гермейера предназначены для решения задач в условиях риска, так как для них необходимо знать вероятности наступления событий полной группы, а для этого ЛПР нужно проводить дополнительные расчёты и искать дополнительную информацию.

Выводы. В ходе работы была формализована задача принятия решения об оптимизации работы по обслуживанию клиентов в условиях неопределённости. К формализованной задаче были применены различные критерии поиска оптимального решения в условиях неопределённости, которые дали одинаковые результаты. Однако, однокритериальные задачи встречаются на практике редко, чаще ЛПР имеет дело с многокритериальными задачами. Так для текущей задачи дополнительным критерием, помимо максимизации прибыли, можно считать минимизацию разницы планового и реально выделенного времени обслуживания.

Из этого следует, что каких-либо особых рекомендаций для выбора критерия определения наилучшей альтернативы нет. Тем не менее, ЛПР стоит выбирать тот критерий, который наиболее близко характеризует его позицию: пессимистическую, нейтральную или оптимистическую. Однако, наилучшей рекомендацией будет использование нескольких критериев, чтобы оценить задачу, если не со всех, то хотя бы с нескольких позиций отношения ЛПР к результатам выбора.

И в заключение хотелось бы отметить, что руководитель организации, принимающий окончательное управленческое решение, основанное на научном подходе больше ориентирован на успех, чем руководитель, который предполагает лишь «шансы на выигрыш».

Список литературы

1. Бродецкий Г.Л. Системный анализ в логистике. Выбор в условиях неопределённости. – М.: Academia. 2010. – 336 с.
2. Кочетов Ю. Курс лекций по теории принятия решений. – Режим доступа: <http://www.math.nsc.ru/LBRT/k5/>.
3. Смирнов Э.А. – Управленческие решения. – М.: ИНФРА -М.2001. – 264 с.
4. Литвак Б.Г. – Разработка управленческого решения. – М.: Издательство «Дело». 2002. – 392 с.
5. Евланов Л.Г. Теория и практика принятия решений. – М.: Экономика. 1984. – 176 с.

УДК 519.71

О сравнении поведения ОДк-эталона и автоматов, порождаемых его локальными преобразованиями Вергелес А.А., Копытова О.М.

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 сравнении поведения ОДк-эталона и автоматов, порождаемых его локальными преобразованиями. Рассматривается инициальный ОД-к автомат-эталон и класс автоматов, которые получаются из эталона переброской нескольких дуг. Для указанного класса вводится бэровская метрика, позволяющая определить расстояние между любой парой автоматов. Найдена достижимая верхняя оценка для этого расстояния. Описаны подклассы автоматов, находящиеся на заданном расстоянии от эталона.

Ключевые слова: автомат, ОД-к автомат, поведение, переброска дуг, бэровская метрика.

Введение. Работа посвящена одной из классических и актуальных проблем теории конечных автоматов - задаче сравнения поведения автоматов с помощью экспериментов. Структура классов автоматов, близких по поведению в том или ином смысле, исследуется достаточно давно [1]. В основном эта задача рассматривалась в рамках теории экспериментов с автоматами [2]. Во многих случаях такие классы можно описать как результат последовательности специальных преобразований над дугами автомата-эталона. Преобразованием такого типа может быть, например, переброска дуг в графе переходов автомата,

которую можно интерпретировать как «неисправность» автомата в его схемной реализации.

Влияние перебросок на поведение автомата изучалось в [3-4], где основное внимание было уделено выявлению условий, при которых исходный и преобразованный автоматы остаются изоморфными. В настоящей работе исследуется влияние переброски более, чем одной дуги на поведение инициального определенно-диагностируемого порядка k эталона с точки зрения определения степени близости его поведения до и после переброски. Целью работы является:

- 1) поиск меры близости поведения эталона, являющегося инициальным ОД- k -автоматом, и автомата, полученного из эталона переброской одной дуги,
- 2) оценка этой меры близости;
- 3) описание класса автоматов с заданной степенью близости.

Постановка задачи. Неопределяемые понятия теории автоматов можно найти в [5].

Под автоматом будем понимать инициальный приведенный определенно-диагностируемый порядка k (ОД- k) автомат Мили $A = (S, X, Y, \delta, \lambda, s_0)$, где S, X, Y – конечные множества состояний, входных и выходных символов соответственно, δ, λ – функции переходов и выходов, а s_0 – начальное состояние. При необходимости, чтобы не возникало путаницы, к обозначениям этих объектов будем добавлять индекс A или значок штриха.

Функции δ и λ обычным образом распространяются на множество X^* всех входных слов конечной длины. 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 $\lambda(s, p) \neq \lambda'(t, p)$.

Будем говорить, что вход-выходное слово $w = (p, q)$ порождается состоянием s автомата A , если $\lambda(s, p) = q$. Длину слова w обозначим $|w|$. 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 $\lambda(a, p) = \lambda'(b, p)$, где λ' – функция выходов автомата A или B . Автомат называется приведенным, если все его состояния попарно неэквивалентны. Каждому состоянию s поставим в соответствие множество λ_s всех вход-выходных слов, порождаемых этим состоянием. Под поведением автомата A будем понимать множество λ_{s_0} вход-выходных слов, порождаемых его начальным состоянием.

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 – ее началом, а t – концом. Если $e = (t, x, y, u)$ – дуга в графе переходов автомата A , то

переброской этой дуги из состояния u в состояние v , отличное от u , называем замену её дугой (t, x, y, v) .

Пусть автомат A' получен из автомата A произвольной переброской дуг. Будем для определённости считать, что в результате переброски произошла замена дуги (t, x, y, u) на дугу (t, x, y, v) , где $t, u, v \in S$, $u \neq v$. Состояния автомата A' будем помечать штрихом так, что переброшенная дуга в нём имеет вид $e' = (t', x, y, v')$. Полученный автомат A' назовём автоматом, локально порождённым из эталона A . Обозначим через $K(A)$ класс всех автоматов, локально порождённых из автомата A , включая сам эталон. Интерес представляют следующие вопросы: 1) как определить степень близости поведения автоматов A и A' ; 2) насколько изменилось поведение автомата после переброски дуг; 3) какова структура класса $K(A)$ с точки зрения разбиения его на подклассы автоматов, эквивалентных по степени близости их поведения поведению эталону.

Формализуем поставленную задачу. Введем на классе автоматов бэрзовскую метрику [6] следующим образом. Пусть i – длина некоторого кратчайшего входного слова p , различающего начальные состояния s_0 и s'_0 автоматов A и A' соответственно, т.е. слова, для которого $\lambda(s_0, p) \neq \lambda'(s'_0, p)$. Тогда положим

$$\rho(A, A') = \begin{cases} 0, & \text{если } A = A' \\ 1/i, & \text{если } A \neq A' \end{cases}$$

Ясно, что, чем длиннее различающее слово p , тем меньше расстояние между A и A' . Пусть $|X|=m$ и $|S|=n$. Каждая дуга заканчивается в одном из n состояний, и, следовательно, её можно перебросить в одно из оставшихся $(n-1)$ состояний. Так как число дуг равно mn , то количество возможных перебросок равно $mn(n-1)$, и, значит, класс $K(A)$ состоит из $(mn(n-1)+1)$ автоматов, включая эталон. Данный класс разбивается на подклассы $K_i(A)$ автоматов, находящихся на одинаковом расстоянии от эталона. По определению,

$\rho(A, A) = 0$, и тогда $K_0(A) = \{A\}$. Известно [6], что переброска одной дуги всегда приводит к неизоморфному автомату. Поэтому начальные состояния (s_0, s'_0) автоматов A и A' всегда различимы некоторым словом, и, значит,

$\rho(A, A') > 0$. Заметим, что при переброске дуги её метка не изменяется, т.е. состояния s_0 и s'_0 выдают одну и ту же реакцию на каждый входной символ. Следовательно, длина различающего их слова больше или равна 2, откуда $\rho(A, A') \leq 1/2$ и $K_1(A) = \{A\}$. С другой стороны, пара состояний (s_0, s'_0) всегда различима некоторым входным словом длины, не превосходящей $(2n-1)$, поэтому $\rho(A, A') \geq 1/2n-1$.

Задача состоит в следующем:

- 1) для любой переброски дуги оценить $\rho(A, A')$;

2) описать классы $K_i(A)$ для $i \geq 2$.

Основные результаты. Теорема 1. Множество входно-выходных деревьев высотой k , параметризуемых $X(A)$ равно множеству входно-выходных деревьев высотой k , параметризуемых A .

Доказательство. Усеченные пути диагностического дерева, построенного для A и $X(A)$, не описывают минимальных диагностических последовательностей, однако, множество диагностических путей, представляемых диагностическим деревом, должно содержать пути, которые описывают все минимальные диагностические последовательности для A и $X(A)$. Так как все диагностические пути, представляемые деревом, имеют одинаковую длину, то все они должны быть минимальными. Если диагностическое дерево не представляет диагностических путей, то все эти пути оканчиваются, и, следовательно, для A и $X(A)$ не существует диагностической последовательности.

Пусть A' есть результат переброски дуги (t, x, y, u) в состояние v и d – длина кратчайшего входного слова w , для которого $\delta(s_0, w) = t$.

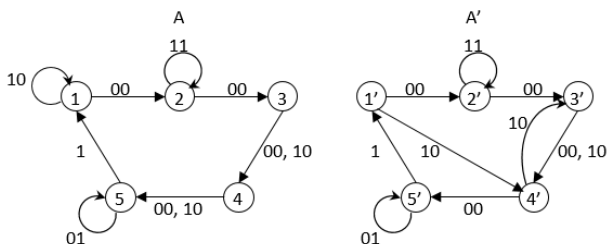


Рис. 1 – Пример переброски дуг в автомате ОД2

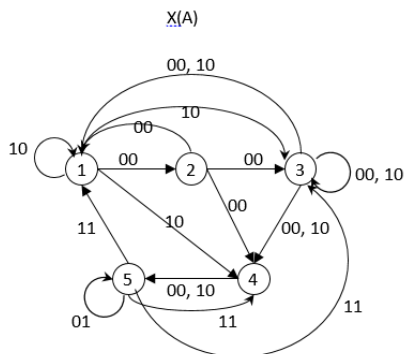


Рис. 2 – Пример $X(A)$ для автомата A

Исходя из теоремы 1 справедлива следующая

Теорема 2. Любая детерминизация автомата A дает A' (с перебросками дуг) с тем же множеством деревьев.

Список литературы

1. Кудрявцев В. Б., Грунский И. С., Козловский В. А.. Анализ поведения автоматов // Дискрет. матем., т.21, вып.1, 2009. – С. 3–35.
2. Bhattacharyya A. Checking experiments on sequential machines. – New York: J. Wiley and Sons, 1989. – 155 с.
3. Копытова О. М. Устойчивость автоматов к неисправностям их функции переходов / О. М. Копытова // Тр. ин-та / ИПММ АН Украины. – 2010. – № 21. – С. 57–66.
4. Копытова О.М. О структуре автоматов, сохраняющих поведение при перебросках дуг. Труды VIII Международной конференции "Дискретные модели в теории управляющих систем". Москва: Макс-Пресс, 2009, с. 155 – 159.
5. Капитонова Ю.В., Кривий С.Л., Летичевский А.А., Луцкий Г.М., Печурин Н.К. Основы дискретной математики. – К.: Наукова думка, 2002. – 568 с.
6. В. Б. Кудрявцев, И. С. Грунский, В. А. Козловский. Анализ и синтез абстрактных автоматов //Фундамент. и прикл. матем., т.15, вып.4. – 2009. – С. 101–175
7. Грунский И.С., Копытова О.М. О структуре контрольного эксперимента для определенно-диагностируемого автомата // Теория управляющих систем.- К.: Наук.думка, 1987. – С.40-54.
8. Гилл А. Введение в теорию конечных автоматов. М.:Наука, 1966.- 272с.

УДК 004.942

Разработка моделей оптимизации уровня запасов производства

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 моделей оптимизации уровня запасов производства. Рассмотрены задачи управления запасами, выделены основные преимущества автоматизации процесса расчета запасов. Определена наиболее удобная модель эффективного управления запасами как с точки зрения логистической системы, так и для сокращения запасов и затрат предприятия.

Общая постановка проблемы. Для того чтобы понять, что такое управление запасами, необходимо обратиться к определению логистики.

Логистика – это наука о планировании, организации, управлении, контроле и регулировании движения материальных и информационных потоков в пространстве и во времени от их первичного источника до конечного потребителя.

Логистика включает в себя ряд наук, одной из которых является логистика запасов. Логистика запасов занимает ключевое место в логистической системе, как отдельной организации, так и экономики в целом.

Исследования. Главная задача логистической науке в сфере производства управления запасами заключается в том, что она направлена на принятие оптимальных решений в своей системе с управлением на основе оценки эффективности ее функционирования. Задачи управления запасами возникают, когда экономический объект не может работать без производственных или товарных запасов, поскольку их отсутствие приводит к простоям, штрафам, потери клиентов и т.д. Исследование логистической системы управления запасами отвечает на многие вопросы, возникающие при функционировании предприятия.

Политика управления запасами не может быть использована без стратегии самого предприятия, а в качестве критерия эффективности принятой стратегии управления запасами выступает функция затрат. Она представляет собою суммарные затраты на поставку и хранение запасаемого ресурса, в том числе включая потери от его порчи при хранении и моральном старении, потери прибыли из-за иммобилизации затрат в оборотные средства, а также потери от штрафов и т.д. Важным фактором при выборе стратегии в управлении запасами является нахождение такой стратегии пополнения и расхода запасов, при которой функция затрат принимает минимальное значение.

Таким образом, запасы являются важным экономическим элементом, который играет огромную роль в осуществлении деятельности производства, а также в обращении продукции. Управление запасами же является неотъемлемым условием успешного функционирования любой фирмы, и представляют из себя процесс прогнозирования, нормирования, планирования, организации, контроля, стимулирования и регулирования сроков и объемов выполнения заказов на восполнение нормы запасов в логистической системе «закупки – производство – распределение».

Задачи управления запасами являются одними из самых сложных в математическом смысле, которые существуют в области управления торговлей. Появившись во время второй мировой войны как задачи управления поставками боеприпасов, они затем нашли свое место в системах управления поставками готовой продукции у крупных промышленных корпораций. С развитием вычислительной техники, созданием огромных систем разветвленной розничной торговли на западе появились модели управления планированием поставок.

Запас материалов необходим практически для каждого вида деятельности, связанного с привлечением поставок. Запас состоит из годных к употреблению, но неиспользуемых ресурсов: материалов, готовой продукции, машин, инструмента. Под стратегией функционирования складской системы понимают правило, которое определяет, когда и сколько следует заказывать.

Причины появления запасов:

- необходимость накопления партии потребления при поставке малых партий от партнеров;
- необходимость накопления партии транспортировки при изготовлении малыми партиями;
- рассогласование режимов работы поставщика и потребителя;
- компенсация сбоев в системе поставок материалов.

Управление запасами заключается в решении двух основных задач:

- 1) определение размера необходимого запаса, т.е. нормы запаса, и частоты его пополнения;
- 2) создание системы контроля за фактическим размером запаса и своевременным его пополнением в соответствии с установленной нормой.

Норма запаса – расчетный минимальный уровень запасов, который должен быть на складе предприятия для обеспечения бесперебойного снабжения производства продукции или реализации товаров. Для определения норм запасов используют три группы методов: эвристические методы, методы технико-экономических расчетов и экономико-математические методы.

Эвристические методы расчета уровня запаса предполагают использование опыта специалистов, которые изучают отчетность за предыдущий период, анализируют рынок и принимают решения о минимально необходимых запасах, основанные, в значительной степени, на субъективном понимании тенденций развития спроса. В качестве специалиста может выступать работник предприятия, постоянно решающий задачу нормирования запасов. В этом случае метод решения задачи называется опытно-статистическим. Если используется опыт сразу нескольких специалистов, то их субъективные оценки ситуации анализируются по специальному алгоритму, проверяются на непротиворечивость и трансформируются в окончательное решение, близкое к оптимальному. Такой метод называется методом экспертных оценок.

Метод технико-экономических расчетов заключается в разделении совокупного запаса в зависимости от целевого назначения на отдельные группы, например, номенклатурные или ассортиментные позиции. Затем для этих групп отдельно рассчитываются страховой, текущий и сезонный запасы. Каждый из этих уровней также может быть разделен на более мелкие составляющие, например, страховой запас на случай нарушения поставок, или страховой запас на случай увеличения спроса и т.д. Метод технико-экономических расчетов

позволяет довольно точно определять необходимый размер запасов, но отличается большой трудоемкостью. Такой подход используется также в системе MRP.

Экономико-математические методы позволяют определять норму запаса на основе построенных математических моделей управления запасами (УЗ), либо с помощью методов экстраполяции прогнозировать будущий запас на основе темпов изменения и тенденций в образовании и запасов в предыдущем периоде.

Эффективность работы системы управления запасами во многом зависит от того, насколько точно будет предсказан спрос на ресурс и, следовательно, насколько правильно будет проведено нормирование. Это является довольно сложной задачей.

В целом, автоматизации поддаются процессы, которые протекают по некоторому четкому алгоритму. Например, чтобы рассчитать уровень запасов в месяцах, нужно взять текущие остатки и разделить на среднемесячные продажи (если их использовать в качестве прогноза). Эта последовательность действий легко прописывается в виде формул, следовательно, легко автоматизируется. Перечислим другие процессы в управлении закупками, которые поддаются автоматизации.

- Учет движения товаров и финансов: приходы, расходы, остатки, задолженность и т.д.

- Формирование отчетов из имеющихся данных, в том числе из разных источников: сводные таблицы, графики, расчеты, выгрузка в Excel и т.д.

Такие операции составляют огромный объем вычислений и, если их не автоматизировать, возникнут большие проблемы со скоростью обработки данных. Но это только одна часть от всего процесса управления закупками.

Существует достаточно много моделей, которые позволяют определить оптимальный уровень инвестиций в запасы и потому многие модели получили достаточное распространение на практике.

Для решения проблем, связанных с запасами предназначены модели управления запасами. Модели должны отвечать на два основных вопроса: сколько заказывать продукции и когда. Есть множество разнообразных моделей, каждая из которых подходит к определенному случаю.

Все модели управления запасами можно разделить на группы:

- Детерминированы модели управления запасами.
- Модели оперативного управления запасами при случайном спросе
- Модели управления запасами в системе с периодическими проверками при случайном спросе.
- Модели управления запасами в течение одного периода.
- Динамические модели управления запасами.

Большинство моделей в базисе своем содержат нижеприведенную (или аналогичную) формулу:

Прибыль = Выручка – Производственные запасы – Издержки хранения – Стоимость разочарования клиента

Стоимость разочарования клиента является, как правило, вмененной величиной, которая может быть оценена и условно включена в затраты, чтобы более полно учесть экономическую прибыль. При наличии зависимого спроса может быть использован подход MRP – планирования потребности в материалах (material requirements planning).

Суть этого подхода заключается в расчете потребностей во всех видах материалов, сырья, комплектующих, деталей, необходимых для производства каждого продукта из плана производства в требуемом объеме, и подаче соответствующих заказов на поставку. В расчетах используются ведомости спецификации – упорядоченный список всех составляющих, необходимых для производства конкретного продукта.

Еще одним способом планирования потребности в материалах является подход "точно в срок" (just in time) или JIT. Цель JIT – обеспечение доставки материалов непосредственно ко времени выполнения конкретных операций, благодаря чему запас фактически уничтожается.

Правильное взаимодействие между закупщиком и компьютером – ключ к эффективному управлению закупками, т.к. оно позволяет объединить и направить в одно русло 2 силы: автоматические расчеты и человеческую логику. Сделать это легче, быстрее и дешевле с помощью грамотных отчетов MS Excel. Специалисту не нужно изучать каждую цифру в бесконечных таблицах с остатками и продажами. По максимуму задействованы инструменты визуализации, с помощью которых на оценку состояния склада уходит всего несколько секунд. Такой подход позволяет мгновенно сконцентрироваться на проблемных местах, оставив в стороне все остальное. Далее работа начинается уже с отдельными позициями.

Возможности Excel для практических целей почти безграничны. Правильно написанные формулы, форматирование ячеек и другие инструменты позволяют решить любую аналитическую задачу. Единственное, пожалуй, для чего Excel не приспособлен, это накапливание и хранение больших массивов данных. Вот для этих целей прежде всего и нужны автоматизированные системы учета (1С и т.п.).

Одним из основных факторов конкурентоспособности промышленного предприятия является эффективное управление его запасами. Сокращение запасов происходит за счет согласованности действий участников логистических процессов, повышения надежности поставок, рациональности распределения запасов.

Процессы управления производством товаров и их распределением чрезвычайно усложняются по всей логистической цепи, следовательно, усложняются процедуры управления и контроля прохождения материальных

потоков по звеньям логистической цепочки. Это требует повышения эффективности распределения ресурсов всех видов, в том числе пребывающих в форме товарно-материальных запасов.

Ни одно современное производство не может обойтись без функций хранения сырья и материалов, полуфабрикатов и комплектующих изделий, инструментов и инвентаря.

Многие результирующие показатели хозяйственной деятельности предприятия в значительной степени зависят от того, как решаются проблемы, связанные с хранением товарно-материальных запасов и поддержанием определенного их уровня, оптимизация которого может быть осуществлена за счет использования логистики.

Возникновение запасов обусловлено естественными природными и экономико-технологическими условиями. Необходимость образования запасов связана также с разделением труда, условиями производства и потребления товаров, которые разобцены как во времени, так и в пространстве.

Современное предприятие, чтобы выпускать определенную продукцию, вынуждено получать различные материальные ресурсы от многих других предприятий и организаций, в том числе территориально отдаленных от него.

Выводы. Интеграция управления закупками и управления производством – это составная часть логистической стратегии предприятия, которая в конечном итоге позволяет получить значительный экономический эффект. Как показали исследования, на предприятиях, где расходование запасов соответствует плановым потребностям производства в сырье и материалах, производственные затраты снижаются, а степень использования инвестированного капитала повышается.

Список литературы

1. Хедли Дж., Уайтин Т. Анализ систем управления запасами, перев. с англ., М.:«Наука», 1969. - 511 с.
2. Кремер Н.Ш. Исследование операций в экономике, М.:Юнити, 2002.- 407 с.
3. Исследование операций: (Учебное пособие для университетов и техн. вузов) - 2-е изд., Киев: Вища школа, 1979. - 552 с.
4. Кудрявцев Б.М. и др. Модели управления запасами. -М.: Ин-т управления им. С. Орджоникидзе,1987. -52с.
5. Алесинская Т.В.Основы логистики. Функциональные области логистического управления, Таганрог: Изд-во ТТИ ЮФУ, 2009. 79 с.

**ИНФОРМАТИКА,
УПРАВЛЯЮЩИЕ СИСТЕМЫ,
МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ
МОДЕЛИРОВАНИЕ
(ИУСМКМ – 2016)**

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

26 мая 2016г.

**Редактор А.Ю. Харитонов
Дизайнер В.В. Харитонова**