



Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»
Отдел студенческой науки и магистратуры

52-я НАУЧНАЯ КОНФЕРЕНЦИЯ
АСПИРАНТОВ, МАГИСТРАНТОВ И СТУДЕНТОВ

**КОМПЬЮТЕРНЫЕ СИСТЕМЫ
И СЕТИ**

25-30 апреля 2016 года

Программа и пригласительный билет

Минск 2016

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

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

**МАТЕРИАЛЫ 52-Й НАУЧНОЙ КОНФЕРЕНЦИИ АСПИРАНТОВ,
МАГИСТРАНТОВ И СТУДЕНТОВ**

(Минск, 25-30 апреля 2016 года)

УДК XXX.X (XXX)
ББК XX.XXX.XXX (4Бей)
К63

Редакционная коллегия:

В. А. Прытков (главный редактор),
М. М. Лукашевич, Е. А. Сасин, Н. В. Лапицкая, С. Е. Карпович,
Г. Ф. Смирнова, Г. И. Малыхина, Н. А. Волорова, Д. И. Самаль,
А. А. Петровский, В. В. Цегельник,

К63 Компьютерные системы и сети: материалы 52-й науч-
ной конференции аспирантов, магистрантов и студентов
(Минск, 25-30 апреля 2016 г.). – Минск: БГУИР, 2016. – 174 с.
ISBN XXX-XXX-XXX-XXX-X.

В сборник включены лучшие доклады, которые были представлены на 52-й научной конференции аспирантов, магистрантов и студентов БГУИР, отобранные по следующим направлениям: электронные вычислительные машины, программное обеспечение информационных технологий, физика, высшая математика, математическое моделирование технических систем и информационные технологии, информатика, философия и жизнь, философия и методология науки, встраиваемые вычислительные системы реального времени.

Для научных и инженерно-технических работников, преподавателей, аспирантов, магистрантов и студентов вузов.

УДК XXX.X (XXX)
ББК XX.XXX.XXX (4Бей)

ISBN XXX-XXX-XXX-XXX-X

© УО «Белорусский государственный
университет информатики
и радиоэлектроники», 2016

СОДЕРЖАНИЕ

СЕКЦИЯ «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

1.	КЛАСТЕРИЗАЦИЯ ДАННЫХ МЕТОДОМ УСТОЙЧИВЫХ ГОМОЛОГИЙ.....	11
2.	ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ: ПРИНЦИПЫ И ПРИЛОЖЕНИЯ К РЕШЕНИЮ АКТУАЛЬНЫХ ЗАДАЧ.....	13
3.	СИСТЕМА ОЦЕНКИ КАЧЕСТВА АЛГОРИТМОВ КЛАССИФИКАЦИИ.....	14
4.	АЛГОРИТМ ИДЕНТИФИКАЦИИ ОБЪЕКТОВ ТОПОЛОГИИ МИКРОСХЕМ НА МИКРОСКОПНЫХ ИЗОБРАЖЕНИЯХ СЛОЕВ ТОПОЛОГИИ СБИС.....	15
5.	МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ЭКСПРЕСС-АНАЛИЗА СОСТОЯНИЯ РАСТИТЕЛЬНОСТИ ПО ИЗОБРАЖЕНИЯМ.....	17
6.	ИСПОЛЬЗОВАНИЕ ЯЗЫКА JAVA ДЛЯ ПОСТРОЕНИЯ МУЛЬТИПЛАТФОРМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ ЦОД	20
7.	РАСПРЕДЕЛЕННЫЙ АНАЛИЗ КРИПТОВАЛЮТНОГО BLOKCHAIN.....	21
8.	СИСТЕМА ПОЗИЦИОНИРОВАНИЯ И УПРАВЛЕНИЯ ПОЛОЖЕНИЕМ КВАДРОКОПТЕРА НА ПЛОСКОСТИ	23
9.	СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ УРОВНЯ ГИБРИДНОСТИ СЕМЯН ПО ЭЛЕКТРОФОРЕГРАММЕ.....	24
10.	ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ ВЗАИМОДЕЙСТВИЯ В СИСТЕМАХ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА.....	25
11.	РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА ОДНОСЛОЙНОГО ПЕРСЕПТРОНА ДЛЯ ПЛАТФОРМЫ OPENCL	26
12.	МЕТОДЫ СИНТЕЗА ЗНАНИЙ, ОПЫТА И РЕСУРСОВ ПОЛЬЗОВАТЕЛЕЙ В КОНТЕКСТЕ «ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СОВМЕСТНОЙ РАБОТЫ»	28
13.	ПРОГРАММНЫЙ МОДУЛЬ РАСПОЗНАВАНИЯ КРЕДИТНЫХ КАРТ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ	30
14.	СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ОБРАБОТКИ ДАННЫХ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА	31
15.	АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ SAAS ДЛЯ ИССЛЕДОВАНИЙ В ОБЛАСТИ ЯДЕРНОЙ ФИЗИКИ	33
16.	ДЕКЛАРАТИВНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PROLOG.....	35

17.	ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ИМИТАЦИИ ПРОЦЕССА ФУНКЦИОНИРОВАНИЯ РЛС ЗРК В УСЛОВИЯХ ВОЗДЕЙСТВИЯ АКТИВНЫХ ПОМЕХ.....	36
-----	--	----

СЕКЦИЯ «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

18.	ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ.....	39
19.	ИССЛЕДОВАНИЕ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ С ИСПОЛЬЗОВАНИЕМ СОФТ-ПРОЦЕССОРА MICROBLAZE	40
20.	АЛГОРИТМЫ И МОДЕЛИ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ ВЕБ- СЕРВИСОВ.....	41
21.	СИСТЕМА ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ В ПОМЕЩЕНИИ.....	42
22.	РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЙ НА REACT.JS, С ПРИМЕНЕНИЕМ FLUX АРХИТЕКТУРЫ.....	43
23.	АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ КВАЛИФИКАЦИЕЙ ПЕРСОНАЛА	44
24.	ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АНАЛИЗА ВИБРАЦИОННЫХ СИГНАЛОВ НА ПЛАТФОРМЕ ANDROID.....	46
25.	VIPER АРХИТЕКТУРА В МОБИЛЬНОЙ РАЗРАБОТКЕ.....	47
26.	БЫСТРОЕ ОПРЕДЕЛЕНИЕ ПРИНАДЛЕЖНОСТИ ПОЛЬЗОВАТЕЛЯ К СЕКМЕНТУ ПО IP-АДРЕСУ	48
27.	МОДЕЛЬ СИСТЕМЫ РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ ДАННЫХ	49
28.	МЕТРИКИ ОЦЕНКИ ФУНКЦИОНИРОВАНИЯ ПРИЛОЖЕНИЙ ПОД УПРАВЛЕНИЕМ JAVA EE СЕРВЕРОВ	50
29.	АВТОМАТИЗИРОВАННЫЕ МЕТОДЫ ОБРАБОТКИ ПСИХОЛОГИЧЕСКИХ ТЕСТОВ С ВОПРОСАМИ ОТКРЫТОГО ТИПА.....	51
30.	ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ФРОНТЕНДА ВЫСОКОНАГРУЖЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ.....	52
31.	МОДЕЛЬ ПРОЦЕССА УПРАВЛЕНИЯ.....	53
32.	МОДЕЛИ И АЛГОРИТМЫ ОЦЕНКИ ЭФФЕКТИВНОСТИ БАЗ ДАННЫХ.....	54
33.	ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ СВЯЗИ МЕЖДУ КОМПЬЮТЕРАМИ.....	55
34.	АЛГОРИТМ ПРОГНОЗА РЕЗУЛЬТАТОВ СПОРТИВНЫХ МАТЧЕЙ.....	56

35.	РАСПОЗНАВАНИЕ ВОЗДУШНЫХ ЦЕЛЕЙ ПО РАДИОЛОКАЦИОННЫМ ПОРТРЕТАМ С ПОМОЩЬЮ ЭВРИСТИЧЕСКОЙ ВОЗМОЖНОСТНОЙ КЛАСТЕРИЗАЦИИ.....	57
36.	ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ПРОСТРАНСТВЕННЫХ ДАННЫХ	58
37.	ПЕРЕХОД К ИНТЕРАКТИВНОМУ ОБУЧЕНИЮ	59
38.	ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ.....	60
39.	АВТОМАТИЗАЦИЯ ОБРАБОТКИ ИНФОРМАЦИИ В СИСТЕМЕ МЕДИАПЛАНИРОВАНИЯ	61
40.	РАСШИРЯЕМОЕ ПРОГРАММИРОВАНИЕ ПРИЛОЖЕНИЙ MICROSOFT OFFICE	62
41.	ПЕРСПЕКТИВЫ АВТОМАТИЗИРОВАНИЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ ПОЖАРНОЙ БЕЗОПАСНОСТИ	64
42.	ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ И ИХ ПРИМЕНЕНИЕ В КРИПТОГРАФИИ	65
43.	ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ ТРЕНИНГА ПО КУРСУ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»	67
44.	ЯЗЫК ПРОГРАММИРОВАНИЯ ДЛЯ ОПИСАНИЯ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ ПРИНЦИПОВ DOMAIN-DRIVEN DESIGN.....	68
45.	ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА СИГНАЛА СПЕКТРА, ПРИМЕНЯЕМАЯ В СИСТЕМЕ ПОДДЕРЖКИ ЛАЗЕРНОЙ ЭКСПРЕССНОЙ ЭКСПЕРТИЗЫ.....	69
46.	ОБ ОБЩИХ ПОДХОДАХ К РАЗРАБОТКЕ ГРАФИЧЕСКОГО ФРЕЙМВОРКА ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ.....	70
47.	ПОСТРОЕНИЕ ИНФРАСТРУКТУРЫ СИСТЕМЫ АВТОМАТИЗАЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	72
48.	МЕТОД ОБНАРУЖЕНИЯ УЯЗВИМОСТЕЙ К SQL-ИНЪЕКЦИЯМ В WEB-ПРИЛОЖЕНИЯХ.....	73
49.	СНИЖЕНИЕ РАЗМЕРНОСТИ ИССЛЕДУЕМЫХ ДАННЫХ НА ОСНОВЕ ЭВРИСТИЧЕСКОЙ ВОЗМОЖНОСТНОЙ КЛАСТЕРИЗАЦИИ	74
50.	ПРОГРАММНОЕ СРЕДСТВО ИНТЕГРАЦИИ ERP-СИСТЕМЫ С СИСТЕМОЙ ПРОВЕДЕНИЯ ПЛАТЕЖЕЙ.....	75
51.	СТРУКТУРИРОВАННОЕ ФОРМАТИРОВАНИЕ КОДА НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ В СИСТЕМАХ РАСПРЕДЕЛЕННОГО КОНТРОЛЯ ВЕРСИЙ	76
52.	АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ ПРИ ПОМОЩИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА.....	77

53.	ЭВОЛЮЦИОННЫЙ АЛГОРИТМ ГЕНЕРАЦИИ ТЕСТОВЫХ СЦЕНАРИЕВ JSON ВЕБ-СЕРВИСОВ.....	79
54.	АНАЛИЗ ЭФФЕКТИВНОСТИ РАБОТЫ СИСТЕМЫ РАСПРЕДЕЛЕННОЙ ПЕРЕДАЧИ ДАННЫХ.....	80
55.	МОДЕЛИ КАЧЕСТВА КЛИЕНТСКОЙ И СЕРВЕРНОЙ ЧАСТЕЙ ВЕБ-ПРИЛОЖЕНИЙ.....	81
56.	ОБРАБОТКА ДАННЫХ СЕНСОРНЫХ УСТРОЙСТВ МОБИЛЬНЫХ ПЛАТФОРМ.....	82
57.	АРХИТЕКТУРА ДЕЦЕНТРАЛИЗОВАННОЙ РАСПРЕДЕЛЁННОЙ ПОДСИСТЕМЫ ХРАНЕНИЯ ДАННЫХ В СИСТЕМАХ ПРИНЯТИЯ РЕШЕНИЙ.....	83
58.	ПОСТРОЕНИЕ АНАЛИТИЧЕСКОГО СЕРВИСА ДЛЯ АНАЛИЗА СПЕКТРОВ.....	85
59.	АНАЛИЗ ЗАЩИЩЕННОСТИ WEB-ПРИЛОЖЕНИЙ.....	86
60.	СИСТЕМА КРИТЕРИЕВ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПРОГРАММНЫХ СРЕДСТВ ОРГАНИЗАЦИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ.....	87
61.	СЕМАНТИЧЕСКАЯ НАВИГАЦИЯ В БОЛЬШИХ ИНФОРМАЦИОННЫХ МАССИВАХ.....	89
62.	ХАРАКТЕРИСТИКИ КАЧЕСТВА СПЕЦИФИКАЦИЙ ТРЕБОВАНИЙ.....	91

СЕКЦИЯ «ФИЗИКА»

63.	РАСПРОСТРАНЕНИЕ ЭЛЕКТРОМАГНИТНОГО ИЗЛУЧЕНИЯ.....	94
64.	МЕТОД РЕЗЕРФОРДОВСКОГО ОБРАТНОГО РАССЕЯНИЯ КАК СПОСОБ ИЗУЧЕНИЯ СТРУКТУРЫ ВЕЩЕСТВА: ОСНОВЫ И ТЕХНОЛОГИИ.....	95
65.	МЕТОДЫ СИНТЕЗА ЗНАНИЙ, ОПЫТА И РЕСУРСОВ ПОЛЬЗОВАТЕЛЕЙ В КОНТЕКСТЕ «ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СОВМЕСТНОЙ РАБОТЫ».....	96
66.	ОПТИЧЕСКИЕ ИЛЛЮЗИИ.....	98
67.	ФИЗИКО-МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА ТАЯНИЯ ЛЕДНИКОВ И ЕЁ ПРОЕКЦИЯ НА ГЕОФИЗИЧЕСКУЮ ПРОБЛЕМУ ПОВЫШЕНИЯ УРОВНЯ МИРОВОГО ОКЕАНА.....	99
68.	КОЛЕБАНИЯ УПРУГОЙ СТРУНЫ.....	100
69.	СПІНТРОНІКА: БУДУЧЫНЯ ЭЛЕКТРОННЫХ ПРЫЛАД.....	102
70.	УСТРОЙСТВО РОТОРНОГО ВЕТРОГЕНЕРАТОРА. ЗАВИСИМОСТЬ ПРОИЗВОДИТЕЛЬНОСТИ ОТ ФОРМЫ И ПЛОЩАДИ ЛОПАСТЕЙ.....	103

СЕКЦИЯ «ВЫСШАЯ МАТЕМАТИКА»

71. МЕТОД МОДЕЛИРУЕМОГО ОТЖИГА И ЕГО ПРИМЕНЕНИЕ ДЛЯ РЕШЕНИЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЯ УЧАСТИЯ АКТЕРОВ В КИНОСЪЕМКЕ..... 105

СЕКЦИЯ «МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ СИСТЕМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

72. УДАЛЕННОЕ УПРАВЛЕНИЕ МНОГОКООРДИНАТНОЙ СИСТЕМОЙ ПЕРЕМЕЩЕНИЙ ЧЕРЕЗ ИНТЕРНЕТ..... 108
73. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РЕШЕНИЯ ПРЯМОЙ ЗАДАЧИ КИНЕМАТИКИ МАНИПУЛЯТОРА НА ГИБРИДНОМ ЛИНЕЙНОМ ПРИВОДЕ 110

СЕКЦИЯ «ИНФОРМАТИКА»

74. APPLICATION OF IBM WATSON FOR THE BEST SPECIALITY OPTION PREDICTION FOR PROSPECTIVE STUDENTS..... 113
75. ПРОБЛЕМА ДОСТИЖЕНИЯ КОНСЕНСУСА В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ..... 115
76. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ КОГНИТИВНЫХ СЕРВИСОВ IBM WATSON, ДОСТУПНЫХ В РАМКАХ ОБЛАЧНОЙ ПЛАТФОРМЫ IBM BLUEMIX 116
77. АНАЛИЗ ИСПОЛЬЗОВАНИЯ МЕТОДОВ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ ДЛЯ ЗАЩИТЫ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ 117
78. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ВЫСОКОНАГРУЖЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ JAVA 118
79. МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЙ..... 119
80. ОБРАБОТКА И АНАЛИЗ ИЗОБРАЖЕНИЯ ОТПЕЧАТКА ПАЛЬЦА ВЫДЕЛЕНИЕ ОСОБЫХ ТОЧЕК ОТПЕЧАТКА 120
81. ИСПОЛЬЗОВАНИЕ ОБУЧАЮЩИХСЯ АЛГОРИТМОВ ДЛЯ АНАЛИЗА СПРОСА НА ПРОДУКЦИЮ И ОРГАНИЗАЦИИ ЦЕПОЧЕК ПОСТАВЩИКОВ 122
82. ВЕРОЯТНОСТНАЯ МОДЕЛЬ РЕЙТИНГОВ ИГРОКОВ В КОМАНДНЫХ ВИДАХ СПОРТА 124
83. ПРОБЛЕМЫ ДИНАМИЧЕСКОГО СОЗДАНИЯ И МОНИТОРИНГА ВИРТУАЛЬНЫХ МАШИН 126
84. ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ ВЕБ-СЕРВИСОВ НА ОСНОВЕ МОДЕЛИ АКТЕРОВ 127

85.	ИССЛЕДОВАНИЕ СОВРЕМЕННЫХ МЕТОДОВ ПОСТРОЕНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ	129
86.	ВЗЛОМ ПОЛИНОМИАЛЬНЫХ ХЕШЕЙ ПО РАВНЫМ СТЕПЕНИ ДВОЙКИ МОДУЛЯМ С ПОМОЩЬЮ ПОСЛЕДОВАТЕЛЬНОСТИ МОРСА-ТУЭ	130
87.	СТЕГАНОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ	131
88.	ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ВОЗМОЖНОСТЕЙ ПРОГРАММНЫХ ПЛАТФОРМ NODE.JS, PYTHON И GO	133
89.	СОБЫТИЙНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ОБРАБОТКИ НАБОРОВ ДАННЫХ	134

СЕКЦИЯ «ФИЛОСОФИЯ И ЖИЗНЬ»

90.	ПРОБЛЕМЫ ФОРМИРОВАНИЯ ЛИЧНОСТИ В СОВРЕМЕННОМ ОБЩЕСТВЕ	137
91.	ФИЛОСОФИЯ СМЕРТИ ИЛИ ЕСТЬ ЛИ ЖИЗНЬ ПОСЛЕ СМЕРТИ?	139
92.	ЧЕЛОВЕК КАК СИНЕРГЕТИЧЕСКАЯ СИСТЕМА	141
93.	АГРЕССИВНОСТЬ ПОДРОСТКОВ КАК СОЦИОКУЛЬТУРНОЕ ЯВЛЕНИЕ... ..	142
94.	В ЧЁМ СЧАСТЬЕ ЧЕЛОВЕКА?	143
95.	НРАВСТВЕННЫЕ ЦЕННОСТИ В ФИЛОСОФИИ АЛЬБЕРА КАМЮ	144
96.	ЛОГИКА ХОАРА.....	145
97.	ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ КАК ОДНА ИЗ ПАРАДИГМ ПРОГРАММИРОВАНИЯ	146
98.	АНТОЛОГИЧЕСКИЙ ЭТАП АНТИЧНОЙ ФИЛОСОФИИ: СОФИСТЫ И СОКРАТ	148

СЕКЦИЯ «ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»

99.	СИСТЕМА АНАЛИЗА ИНТЕРНЕТ-ТРАФФИКА И ЭФФЕКТИВНОСТИ КАНАЛОВ ЕГО ПРИВЛЕЧЕНИЯ.....	151
100.	РАЗРАБОТКА USBPOS-КЛАВИАТУРЫ НА БАЗЕ 32-ХРАЗРЯДНОГО МИКРОКОНТРОЛЛЕРА ARMCORTEX-M0.....	152
101.	АЛГОРИТМ УЧЁТА КОЛЛИЗИЙ ПЛАНАРНЫХ ПОЗИЦИОНЕРОВ НА ОДНОМ СТАТОРЕ	153

102.	ОБНАРУЖИТЕЛЬНАЯ СПОСОБНОСТЬ ДЛЯ АВТОМАТИЧЕСКОГО КОНТРОЛЯ ТОПОЛОГИИ.....	155
103.	СПЕЦИАЛИЗИРОВАННЫЙ ПРОЦЕССОР ШИФРОВАНИЯ СТАНДАРТА СТБ 34.11.31-2007	156
104.	СПОСОБ ФОРМИРОВАНИЯ ВЫСОКОЧАСТОТНОЙ СОСТАВЛЯЮЩЕЙ СПЕКТРА СИГНАЛА В НИЗКОСКОРОСТНОМ АУДИОКОДЕРЕ.....	158
105.	СИСТЕМА ИДЕНТИФИКАЦИИ ДИКТОРА В АКУСТИЧЕСКИХ ШУМАХ НА ОСНОВЕ АНТРОМОРФИЧЕСКОЙ ОБРАБОТКИ РЕЧЕВОГО СИГНАЛА	160
106.	СИСТЕМА УЛУЧШЕНИЯ ВОСПРИЯТИЯ И РАЗБОРЧИВОСТИ АУДИО СИГНАЛА В АГРЕССИВНОЙ ШУМОВОЙ СРЕДЕ.....	162
107.	ПРИМЕНЕНИЕ ЧИСЛЕННЫХ МЕТОДОВ ДЛЯ СИНТЕЗА КОЭФФИЦИЕНТОВ ПАРАУНИТАРНОГО БАНКА ФИЛЬТРОВ НА ОСНОВЕ АЛГЕБРЫ КВАТЕРНИОНОВ.....	164
108.	КВАНТОВАНИЕ И УПАКОВКА КОЭФФИЦИЕНТОВ СУБПОЛОС В КОДЕРЕ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ПАРАУНИТАРНОГО БАНКА ФИЛЬТРОВ.....	166
109.	ОПИСАНИЕ ПРОЦЕССОРНОГО ЯДРА MOTOROLA MC6800 НА ЯЗЫКЕ ОПИСАНИЯ АППАРАТУРЫ VHDL	167
110.	РАЗРАБОТКА ИГРОВОГО ПРОЕКТА НА UNREALENGINE 4	168
111.	СИСТЕМА УПРАВЛЕНИЯ КОДОВЫМ ЗАМКОВ НА БАЗЕ ПЛАТФОРМЫ ARDUINOMEGA 2560	169
112.	СИСТЕМА ДЕТЕКТИРОВАНИЯ КЛЮЧЕВЫХ ФРАЗ В РЕЧЕВОМ СИГНАЛЕ	171
113.	СИСТЕМА ИДЕНТИФИКАЦИИ ДИКТОРА ПО ОБРАЗЦУ ГОЛОСА НА БАЗЕ РЕЧЕВЫХ ЗАПИСЕЙ	173

СЕКЦИЯ «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

КЛАСТЕРИЗАЦИЯ ДАННЫХ МЕТОДОМ УСТОЙЧИВЫХ ГОМОЛОГИЙ

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

Бубнов Я. В.

Иванов Н. Н. – канд. физ-мат. наук, доцент

Задача кластеризации в анализе данных может быть представлена как разбиение большого конечного множества векторов на подмножества на основании выбора подходящей метрики векторного пространства. В общей постановке топологический анализ данных предоставляет методику анализа данных со слабой чувствительностью к конкретным метрикам, размерности пространства и высокой устойчивостью к зашумленности обрабатываемых данных.

Теория гомологии – это раздел алгебраической топологии, изучающий геометрическую структуру данных. В приложениях теории особое внимание уделяется вычислительной сложности алгоритмов построения и исследования структуры данных, а именно, алгоритмам с линейной и полиномиальной сложностью.

Классический подход в применении гомологии для кластеризации данных состоит в представлении образов (совокупностей векторов информационных признаков) в виде множества симплексов $\sigma^i \subseteq K(P)$ n -мерного полиэдра $P \subset R^N$ (замкнутого N -мерного многогранника) [1].

Основанием для разбиения множества образов на кластеры является n -мерная гомологическая группа $H_n(P)$ полиэдра P , которая представляет собой факторгруппу $Z_n(P) / B_n(P)$ n -мерных циклов по n -мерным границам и определяет совокупность топологических инвариантов (такие как дыры и пустоты), которые и дают основание для разбиения пространства образов.

Пространство образов получается путем объединения симплексов в комплексы на основании их близости, т.е. используется некоторая метрика d . При этом при построении группы, операция взятия границы симплекса вычисляется путем понижения размерности симплекса, то есть поочередным «отбрасыванием» вершин симплекса v_j [2]:

$$\partial \sigma^i = \sum_{j=0}^i (-1)^j [\sigma^i - v_j]$$

Например, отбрасывание вершины треугольника (двумерной фигуры) оставляет в результате две вершины, то есть отрезок (одномерная фигура). Однако, рассмотрение статического топологического пространства, построенного по метрике d не позволяет судить в целом о структуре пространства [3]. Для этого используется фильтрация комплекса K [4]:

$$K_0 \subset K_1 \subset \dots \subset K_k$$

$$K_i \subset \sigma_i \subset K_{i+1} \subset \sigma_{i+1} \subset K_{i+2}$$

где K_0, K_1, \dots, K_k – симплициальные комплексы, причем объединение (склеивание) двух n -мерных симплексов σ_i, σ_j некоторой гранью e_{ij}^n осуществляется только в том случае, если расстояние между ними не превышает заданной величины ϵ – это основа кластеризации образов.

Далее вводится последовательность отображений гомологических групп фильтрации путем применения гомологического функтора к каждому элементу фильтрации:

$$H_0(K_0) \rightarrow H_0(K_1) \rightarrow \dots \rightarrow H_0(K_k)$$

причем отображение $f_{i,j}: H_n(K_i) \rightarrow H_n(K_j)$ является инъективным на основании вложенности множеств $K_i \subset K_j$. Тогда совокупность интервалов всех размерностей будут образовывать так называемые диаграммы устойчивости [4].

Алгоритм вычисления интервалов устойчивости гомологий базируется на концепции порождающих и поглощающих симплексов фильтрации $K^i = \{\square^i \mid 0 \leq i \leq k\}$. Порождающим называется n -симплекс \square^i , полученный в результате i -ого этапа фильтрации, который будет принадлежать n -циклу c^n , и поглощающим в противном случае. Порождающие симплексы определяют создание гомологического класса, а поглощающие симплексы определяют слияние класса в граничную группу и формируют границы интервала устойчивости гомологии.

Для вычисления интервалов гомологической устойчивости используется граничная матрица, где на пересечении j -го столбца и i -ой строки расположена 1, если $(n-1)$ -мерный \square^{i-1} симплекс является границей n -мерного симплекса \square^i , сформированных на n -ом этапе фильтрации топологического комплекса и 0 иначе.

Для граничной матрицы определяется операция редукции как построковое суммирование столбцов с одинаковыми значениями индекса крайнего ненулевого элемента, удовлетворяющих следующему условию:

$$j_{i-i_0} = j_i + j_{i_0}, \quad i < i_0,$$

где i_0 является индексом рассматриваемого столбца. Редуцированная описанным образом результирующая матрица будет содержать в каждом столбце информацию об образовании и слиянии новых гомологических классов.

В качестве примера использования метода устойчивых гомологий, рассмотрим решение задачи кластеризации для 16 кластеров, сформированных 1024 образами с тремя информативными признаками. Исходные данные для эксперимента взяты из проведенного автором эксперимента по выявлению угроз безопасности локальной сети. Фильтрация симплексов выполнена с помощью альфа-комплексов, базирующихся на триангуляции Делоне.

На рисунке 2(а) представлены устойчивости гомологий топологической структуры, образованной нульмерными симплексами рассматриваемого множества образов, и характеризуют компоненты связности образованного пространства.

Из результатов видно, что некоторые топологические пустоты остаются стабильными на достаточно продолжительном интервале фильтрации, что стоит рассматривать как кластеризацию пространства образов. На рисунке 2(б) изображена диаграмма устойчивости для одномерных симплексов.

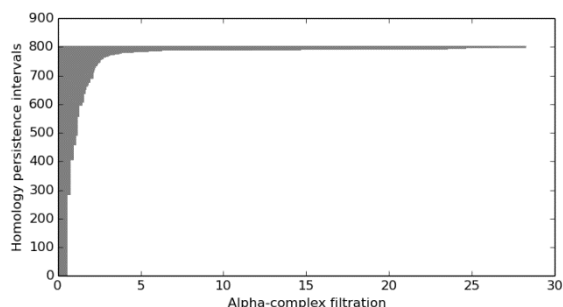


Рис. 1. Диаграммы устойчивости гомологий 0-мерных Alpha-комплексов.

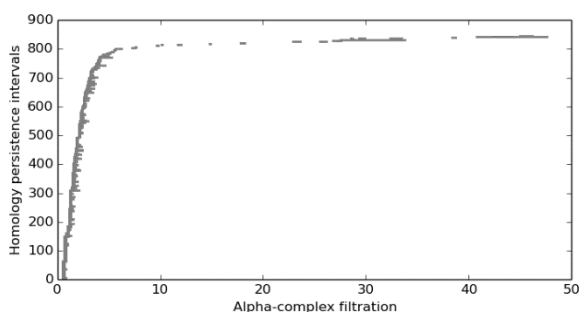


Рис. 2. Диаграммы устойчивости гомологий 1-мерных Alpha-комплексов.

При фильтрации комплексов большей размерности наблюдается низкая продолжительность существования гомологий (рис. 3) в процессе фильтрации, что говорит, в целом о неразделимости образов в пространстве рассматриваемой размерности.

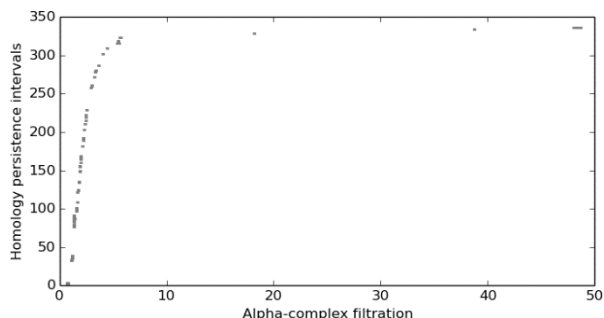


Рис. 3. Диаграмма устойчивости гомологий 2-мерных Alpha-комплексов.

Таким образом, одной из немаловажных проблем использования статистических методов кластеризации является необходимость априорного задания количества кластеров, на которое будет осуществляться разбиение исходной совокупности образов. Благодаря кластеризации методом устойчивых гомологий проблема выбора количества кластеров переносится в другую плоскость, так как результатом данного алгоритма является совокупность всех возможных разбиений, и вопрос уже ставится о необходимой степени различия между ними. Известные алгоритмы вычисления интервалов устойчивости гомологий имеют кубическую сложность [5].

Список использованных источников:

1. Johnson, J. Topological graph clustering with thin position / J. Johnson // Cornell University Library – Topological graph clustering with thin position. – Stillwater, 2012. – 2 с.
2. Хатчер, А. Алгебраическая топология / Хатчер А. // Алгебраическая Топология. Пер. с англ. В. В. Прасолова под ред. Т.Е. Панова – Москва, 2011. – 138 с.
3. Carlson, G. Topology and data / G. Carlson // American mathematical society. Topology and data. – California, 2009. – 256 с.
4. Edelsbrunner, H. Topological Persistence and Simplification / H. Edelsbrunner, D. Letscher, A. Zomorodian // Topological Persistence and Simplification. – Illinois, 2002. – 3 с.
5. Edelsbrunner, H. Computational topology: an introduction / H. Edelsbrunner, J. Harer // American mathematical society. Computational topology: an introduction. – Duke, 2010 – 182 с.

ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ: ПРИНЦИПЫ И ПРИЛОЖЕНИЯ К РЕШЕНИЮ АКТУАЛЬНЫХ ЗАДАЧ

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

Хвалько Д. В.

Искра Н. А. – старший преподаватель

Логическое программирование радикально отличается от других парадигм программирования. Сегодня в связи с техническим развитием в области параллельных вычислений и BigData стало возможным эффективное практическое применение логического программирования.

Отличительной чертой логической парадигмы программирования является наличие кванторов существования - утверждений о том, что существуют некоторые объекты с заданными свойствами. При работе с этой парадигмой используется конструктивный метод доказательства целевых утверждений: если требуемое было доказано, то в процессе доказательства находятся элементы, соответствующие ранее не определённым объектам, которые, однако, входят в целевое утверждение. Это соответствие и создаёт результат вычислений. Буквально это можно воспринимать как два выражения: программа – множество аксиом, вычисление – конструктивный вывод целевого утверждения из программы (множества аксиом).

Парадигма логического программирования имеет следующие особенности:

1. В логических языках отсутствуют операторы присваивания и побочные эффекты.
2. Применяется естественная математическая модель вычислений.
3. Есть возможность возвратов и перебора.
4. В язык встроены возможности по представлению списков, деревьев и др. коллекций.
5. Высоко развиты возможности мета-программирования.

Аппарат логического программирования, как и первые языки, основанные на этой парадигме, были разработаны в шестидесятые годы двадцатого века. С тех пор, за неимением необходимых технологий и достаточных ресурсов, про эту парадигму почти забыли, но в наши дни, в след за появлением параллельных вычислений, облачных вычислений и больших данных, благодаря увеличению вычислительных мощностей компьютеров, парадигма логического программирования вышла на новый виток развития. Логическое программирование в наши дни позволяет частично решать проблемы неоднозначности человеческого языка и проблему обобщения знаний в задачах искусственного интеллекта и построения экспертных систем [1].

Языки логического программирования представляют собой формализованные языки, где программа есть описание требуемого решения в терминах математической логики, а решение задачи строится в процессе логического вывода по заданному описанию. Существует много подходов к разработке программ на логических языках программирования: например, с использованием индуктивной логики или с использованием ограничений.

Исторически первым языком логического программирования был язык Planner. Этот язык был основан на автоматическом выводе результатов из данных и заданных правил перебора вариантов (которые и назывались планом, отсюда и название языка). Planner использовался, например, для снижения требований к вычислительным ресурсам с помощью поиска с возвратом.

Немного позднее был разработан язык Prolog, который не требовал плана перебора вариантов и был, в некотором роде, упрощением языка Planner.

От языка Planner в свою очередь произошли такие языки, как Conniver, Popler, QA-4, и QLISP. Языки программирования Datalog, Mercury, Visual Prolog были выделены как подмножества языка Prolog. Существуют также альтернативные языки логического программирования, которые не используют поиск с возвратом, например, Ether.

В последние годы, Datalog, который является синтаксическим подмножеством языка Prolog для представления и обработки данных, вновь возник в центре внимания создателей широкого спектра новых приложений, в том числе в таких сферах как:

- интеграция данных;

- мониторинг компьютерных сетей;
- извлечение информации (Data Mining);
- защита информации;
- облачные вычисления [2].

Общим для всех этих сфер является использование языка Datalog как более высокого уровня абстракции данных в выполнении запросов и построении дедуктивных баз данных [3].

Список использованных источников:

1. Цуканова, Н.И. Теория и практика логического программирования на языке Visual Prolog 7/ Н.И. Цуканова, Т.А. Дмитриева // Международный журнал экспериментального образования, №2, 2012. – стр. 77.
2. Datalog and Emerging Applications: An Interactive Tutorial / Shan Shan Huang, Todd J. Green, Boon Thau Loo // SIGMOD'11. – Athens, 2011.
3. Катериненко, Р. Дедуктивные базы данных: построение продукционной модели с помощью современных СУБД / Lambert Academic Publishing, 2012.

СИСТЕМА ОЦЕНКИ КАЧЕСТВА АЛГОРИТМОВ КЛАССИФИКАЦИИ

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

Михалькевич Е. Ф.

Лукашевич М. М. – канд. техн. наук

Система предназначена для оценки качества различных алгоритмов классификации. Оценка может производиться с использованием различных методик и метрик оценки, а также с использованием любых входных данных, что позволяет подобрать алгоритм классификации под конкретную задачу.

Классификатор – это абстрактный автомат (математическая модель), принимающий решение об отнесении вектора информативных признаков к одному из классов. Предполагается, что информативные признаки объединяются по некоторому правилу в интегрированный параметр, по которому и принимается решение. Самым тривиальным решением является дуальное решение - ДА/НЕТ. Для правильной работы классификатора его необходимо обучать и тестировать. Чтобы настроить классификатор под определенную задачу, необходима репрезентативная обучающая последовательность, по которой можно достаточно точно восстановить параметры классификатора. Обычно, для тестирования и обучения классификаторов, базы берутся в различных открытых репозиториях. Самым известным из них является UCI [1].

При решении любой проблемы связанной с машинным обучением в первую очередь производится анализ исходных данных, специфики требований к интеллектуальной системе, а также может строиться модель системы. Это все требуется для выбора методики классификации или кластеризации.

На сегодняшний день существует большое множество различных классификаторов. Каждый из них имеет свои сильные и слабые стороны. Поэтому основной проблемой является выбор классификатора под конкретную задачу или под конкретные данные.

Как известно, классификация не сводится к методам разделения наборов подмножеств в признаковом пространстве. Для заказчика важно не только получить алгоритм, реализующий (возможно с некоторыми ошибками) требуемое разделение классов, но и иметь оценку надежности решения поставленной задачи, т.е. знать, как часто данный алгоритм будет ошибаться при классификации вновь предъявляемых объектов. Ясно, что указанная оценка напрямую определяет качество решения поставленной задачи. На практике же дать такую обоснованную оценку часто оказывается затруднительным. [2]

Процесс обучения классификатора важен не менее чем построение модели. Хотя не существует общих алгоритмов обучения для всех моделей, используются схожие методы для отдельных методов классификации [4].

Предложенная система позволяет решить проблему подбора классификатора. Однако она может использоваться и для других приложений. Например, исследование влияния каждого из признаков на качество классификаторов и уменьшение размерности пространства признаков.

Система оценки качества классификаторов обладает гибкостью конфигурации тестирования алгоритмов. На сегодняшний день существует огромное количество классификаторов. Многие исследователи адаптируют и модифицируют алгоритмы для своего приложения [3]. Из всех классификаторов можно выделить несколько основных и наиболее используемых.

При работе с системой можно выбрать:

- Количество признаков образов;
- Несколько алгоритмов классификации для сравнения;
- Алгоритмы обучения для каждого алгоритма классификации;
- Критерии остановки обучения;

- Метрики для оценки качества.

Одним из ключевых моментов является возможность добавления собственных алгоритмов классификации и обучения. Это достигается с помощью обобщения алгоритмов в группы с одинаковыми интерфейсами и возможностями языка Python.

После получения результатов тестирования их можно сравнить между собой, сохранить для последующего использования, а также экспортировать для печати.

Таким образом, была разработана система оценки качества алгоритмов классификации. Она имеет возможность выбирать различные алгоритмы классификации и разные метрики для оценки. Задачей дальнейших исследований является реализация функционала для алгоритмов кластеризации.

Список использованных источников:

1. <http://archive.ics.uci.edu/ml/> - Электронный ресурс.
2. Гуров, С.И. Как оценить надежность алгоритма классификации // Таврический вестник информатики и математики №1. – Симферополь, 2002.
3. Ramdass, D. Document Classification for Newspaper Articles / D. Ramdass, S. Seshasai // 6.863 Final Project – MIT, Spring 2009.
4. Michie, D. Machine Learning, Neural and Statistical Classification / D. Michie, D.J. Spiegelhalter, C.C. Taylor // Overseas Press, 2009 edition (August 28, 2009) – 290 p.

АЛГОРИТМ ИДЕНТИФИКАЦИИ ОБЪЕКТОВ ТОПОЛОГИИ МИКРОСХЕМ НА МИКРОСКОПНЫХ ИЗОБРАЖЕНИЯХ СЛОЕВ ТОПОЛОГИИ СБИС

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

Питкин А.Б.

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

В процессе производства сверхбольших интегральных микросхем (СБИС), возможно появление дефектов на топологических слоях. Такие дефекты возникают например в процессе переноса изображения или в процессе фотолитографии. Автоматический контроль качества позволит в кратчайшие сроки выявлять различные дефекты. Идея заключается в поиске заданных шаблонов известных дефектов, либо шаблонов с изображением эталонов элементов на значительно большем изображении представляющем собой снимок слоя готовой микросхемы.

Простейшим способом детектирования на изображении является поиск по шаблону (template matching) [2]. Предварительно изображение обрабатывается с использованием фильтра дистантного преобразования [3], что хорошо выделяет контуры объектов на изображении. Далее происходит постепенный параллельный перенос изображения шаблона в окне над исходным изображением слоя и вычисление корреляции между ними по одной из формул [4]. На основании полученного значения корреляции изображений делается вывод о совпадении шаблона с фрагментом под ним.

Однако элементы на слое интегральной микросхемы могут располагаться с вращением кратным 90° в плоскости изображения, что увеличивает число проходов по изображению в четыре раза. Также изображение элемента может незначительно отличаться от эталонного изображения в связи с особенностями технологического процесса производства микросхемы.

Целью алгоритма является задача разработать метод поиска изображения устойчивый к поворотам изображения и масштабированию и позволяющий с высокой степенью вероятности найти все повторения заданного шаблона на изображении.

Универсальными методами поиска изображений являются методы, основанные на выделении ключевых точек на изображении. Поиск на изображении определенных характерных примитивов позволит перейти от описания изображения в виде массива цветных пикселей к массиву параметров описывающих эти примитивы. Таким образом, сокращается количество параметров, по которым необходимо производить дальнейшее сравнение изображения. Поиск на изображении при использовании методов основанных на ключевых точках состоит из трех этапов:

1. Этап детектирования ключевых точек. На данном этапе происходит сканирование изображения и нахождение на нем характерных точек интереса.
2. Этап создания дескрипторов для ключевых точек, найденных на предыдущем шаге. Таким образом, на данном этапе формируется математическое описание найденных ключевых точек в виде вектора значений.
3. Сопоставление наборов значений дескрипторов полученных в процессе обработки двух изображений и нахождение близких друг к другу значений.

Наиболее эффективными типами дескрипторов в данный момент являются дескрипторы SIFT (SIFT - Scale Invariant Feature Transform) [5] и SURF (Speeded Up Robust Features) являющийся улучшением

предыдущего [6]. SURF производит поиск особых точек с помощью матрицы Гессе. Детерминант матрицы Гессе (так называемый гессин) достигает экстремума в точках максимального изменения градиента яркости. Он хорошо детектирует пятна, углы и края линий. Технически детектирование основано на вычислении перепадов яркости изображения.

Точки, найденные с помощью дескриптора SIFT устойчивы к повороту изображения, растяжению. В результате сопоставления дескрипторов на изображении было обнаружено большое число ложных срабатываний дескриптора. Данная проблема связана с малым размером шаблона, по которому выполняется поиск на изображении и как следствие малым количеством информации, которую несет в себе этот шаблон. Это приводит к тому, что дескрипторы шаблону случайным образом совпадают с некоторыми точками на изображении.

Перед применением алгоритма поиска ключевых точек необходимо произвести предварительную обработку изображения слоя топологии интегральной микросхемы. Одной из задач тут является удаление шумов на изображении, а также сглаживание изображения.

Рассмотрим два детектора ключевых точек при поиске заданных шаблонов на изображении. Важным параметром при работе детекторов на последнем этапе является максимально допустимое расстояние между дескрипторами на изображении и на шаблоне. Слишком малое значение отсечет часть верных результатов, слишком высокое значение будет давать слишком большое число ложных срабатываний.

В качестве тестовых изображений использовалось изображение слоя топологии, а также пять шаблонов изображений. Также были созданы копии изображений шаблонов, повернутые на 90 градусов. В процессе тестирования использовались алгоритмы, включенные в библиотеку OpenCV.

Результаты сведены в таблицу 1.

Таблица 1 – Результаты работы детекторов

Шаблон/ детектор	SURF			SIFT		
	Ключевых точек на шаблоне	Совпадений ключевых точек	Итог	Ключевых точек на шаблоне	Совпадений ключевых точек	Итог
Шаблон 1	17	7	+	17	2	+
Шаблон 1 повернутый	17	7	+	19	0	-
Шаблон 2	14	4	+	19	4	+
Шаблон 2 перевернутый	14	4	+	17	0	-
Шаблон 3	2968	1803	+	3066	2772	+
Шаблон 3 перевернутый	2967	1803	+	3054	702	+
Шаблон 4	117	49	+	59	12	+
Шаблон 4 перевернутый	113	56	+	54	1	+
Шаблон 5	6	1	+	15	1	+
Шаблон 5 перевернутый	6	1	+	19	0	-

Время поиска ключевых точек и вычисления дескрипторов в тестовой программе составило 33 секунды для метода SIFT и 23 секунды для метода SURF, при этом использовалось изображение 7000 на 3000 пикселей. В дальнейшем этап поиска ключевых точек и вычисления дескрипторов можно опустить, сохранив найденные значения в кэш. Время сопоставления ключевых точек составляет менее 100 миллисекунд.

Таким образом, в процессе тестирования было определено, что детектор SURF более качественно находит ключевые точки даже на перевернутых изображениях и справился со всеми десятью тестами. Дескриптор SIFT не всегда находил ключевые точки на повернутых изображениях и число совпадений дескрипторов меньше. Оба дескриптора показали хорошие результаты уникальности ключевых точек и отсутствие случайных совпадений отдельных точек с фрагментами на изображении. По сравнению с методами, основанными на поиске по шаблону, данные методы дают значительно меньшее число ложных срабатываний. Это объясняется тем, что происходит сравнение не яркостных характеристик шаблона и изображения, а сравниваются их морфологические особенности.

Были исследованы методы поиска на изображении с использованием ключевых точек. Методы обеспечивают нахождение совпадений в случае поворота изображения и незначительных изменений их размеров.

Список использованных источников:

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Москва, 2005. – 1072с
2. Template matching [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Template_matching
3. Distance Transform [Электронный ресурс]. – Режим доступа: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm>
4. Which are the matching methods available in OpenCV [Электронный ресурс]. – Режим доступа: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html#which-are-the-matching-methods-available-in-opencv
5. Scale-invariant feature transform [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ЭКСПРЕСС-АНАЛИЗА СОСТОЯНИЯ РАСТИТЕЛЬНОСТИ ПО ИЗОБРАЖЕНИЯМ

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

Романовский К. В.

Дудкин А. А. – д-р. техн. наук, профессор

Описана и решена проблема быстрой обработки большого количества данных в системе распознавания сельскохозяйственной растительности путем ее разделения на основную обработку (сервер) и экспресс-анализ (мобильная платформа).

При принятии решений в процессе мониторинга сельскохозяйственных полей основным является распознавание пораженных заболеваниями участков. Для решения данной задачи предложено использование программно-аппаратного комплекса, мощности которого представлены мобильной системой и сервером. Серверная часть предполагает использование мощного компьютера. Она представлена подсистемами параллельной обработки изображений и принятия решений. Первая из них включает в себя наиболее часто используемые методы обработки фотоаэроснимков, вторая – отвечает за выбор способа построения информационных признаков на основе данных дистанционного зондирования Земли (ДЗЗ). В результате эксперт получает возможность детального изучения обработанных фотоаэроснимков различными методами, сэкономив при этом время на обработку за счет распараллеливания на современных вычислительных системах.

Мобильное приложение (мобильная подсистема) – это программные средства, предоставляющие возможность отображать информацию о состоянии сельскохозяйственной растительности на карте, которая является набором упорядоченных по географическим координатам цифровых фотографий. Данная подсистема оснащена возможностью экспресс-оценки состояния растительного покрова и определения зон (сегментов) с нарушением фотосинтеза, позволяет получать наглядную информацию о состоянии растительности в рамках целого поля или о его отдельных участках. Скорость обработки данных обуславливается нахождением мобильной платформы непосредственно у изучаемого участка – цифровые фотографии попадают в систему сразу после окончания процесса съемки, без их длительной транспортировки на сервер. В процессе обработки эксперт формирует выборки необходимых для дальнейшей обработки фотографий, уменьшая количество поступающих на вход сервера данных.

Источником данных был выбран беспилотный летательный аппарат (БПЛА), но не исключено использование современной сельскохозяйственной техники и средств ДЗЗ. Главным требованием к источнику данных является возможность использовать информацию со спутников глобального позиционирования и добавлять ее к изображениям.

В докладе предложена структура мобильного приложения и алгоритм экспресс-анализа состояния сельскохозяйственной растительности. Использование мобильной платформы позволило ускорить принятие решений при мониторинге состояния растительности данных за счет более ранней обработки и экспресс-анализа, проводимых уже на этапе получения информации, а также за счет сокращения объема данных, посылаемых для обработки на сервер.

Разработанное мобильное приложение включает пять основных модулей (рис. 1): приемки и передачи информации, обработки изображений, построения карты, хранения данных, управления БПЛА.

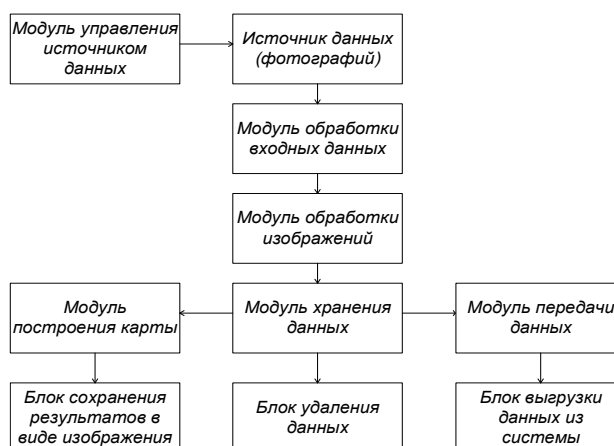


Рис. 1. Структурная схема мобильного приложения

Основой концепции точного земледелия является тот факт, что состояние растительных покровов является неоднородным в пределах одного поля. Для оценки и детектирования этих неоднородностей используются новейшие технологии, такие как системы глобального позиционирования (GPS, ГЛОНАСС), специальные датчики, аэрофотоснимки и снимки со спутников, а также специальные программные системы на базе ГИС [2].

Важной особенностью получаемых снимков является их большой размер, что, вместе с ограниченными вычислительными ресурсами компьютеров, накладывает определенные требования к алгоритмам обработки данных изображений: возможности их оперативной обработки и распараллеливания, ограничения по типу (точечный, локальный или глобальный) и сложности алгоритмов [3, 4].

С точки зрения вычислительной сложности и необходимости лишь экспресс-анализа на мобильном компьютере предпочтительнее будет использовать алгоритм точечного типа с использованием цветовой модели RGB (Red, Green, Blue – красный, зеленый, синий). Использование алгоритмов глобального типа также может ограничиваться большим размером исходного снимка при ограниченных ресурсах компьютера.

Алгоритм обработки состоит в следующем.

1. БПЛА согласно заданным параметрам совершает облет поля и предоставляет фотографии на вход системы. Происходит формирование выборок, если это необходимо.
2. На основе выбранной пользователем фотографии задаются основные параметры – выбирается пять эталонных цветов, их диапазон и пороговое процентное содержание на фотографии.
3. Производится обработка всех фотографий. Подсчитывается количество пикселей для каждого эталона, выполняется проверка, не превышает ли это количество заданный диапазон. Всякое превышение регистрируется и показывается пользователю на информативной карте.
4. Производится обработка GPS-координат фотографии. Данные берутся из информации, закрепленной за каждой цифровой фотографией согласно стандарту EXIF. Упорядочивая снимки по долготе и широте, формируется информативная карта (на ней указаны проблемные участки сельскохозяйственного поля).

При необходимости, можно приступить к новому анализу с другими параметрами, сделать более глубокий анализ проблемных фотографий или сделать запрос на повторное фотографирование участка.

Главное окно программы содержит семь опций, расположенных слева и справа окна. В центре находится область для отображения исходных данных и результатов поэтапной работы алгоритма. На рис. 3 показан вид окна программы с картой (неотсортированные фотографии). Чтобы показать фотографии в соответствии с их координатами, необходимо нажать кнопку «Показать упорядоченные данные».

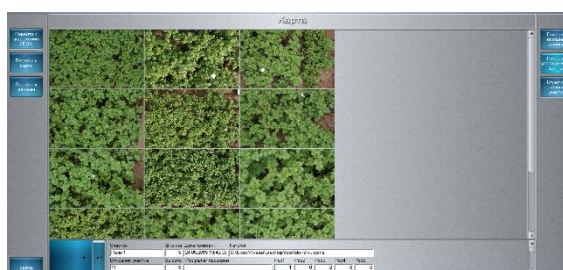


Рис. 2. Вид карты с неотсортированными фотографиями

Разработаны структура и алгоритм работы мобильного приложения системы распознавания сельскохозяйственной растительности, что позволило решить задачу экспресс-анализа сельскохозяйственной растительности без задействования мощностей стационарного сервера и производить экспресс-анализ растительности непосредственно возле исследуемого участка, существенно ускоряя работу эксперта.

Цифровые фотографии попадают в систему сразу после окончания процесса съемки, без их длительной транспортировки на сервер. В процессе обработки эксперт формирует выборки необходимых для дальнейшей обработки фотографий, что, в свою очередь, уменьшает количество данных, поступающих на сервер.

Список использованных источников:

1. Ganchenko, V. Development of the hardware and software complex for fertilizer application on agricultural fields / V. Ganchenko, A. Doudkin, A. Petrovsky, T. Pawlowski // Journal of Research and Applications in Agricultural Engineering. – 2014. – Vol. 59(1). – P. 34–39.
2. Ганченко, В.В. Информационные технологии в задаче мониторинга состояния сельскохозяйственной растительности / В. В. Ганченко, А. А. Дудкин, А. И. Петровский // Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2013): доклады XII Междунар. конфер., Минск, 20 нояб. 2013 г. – Минск: ОИПИ НАН Беларуси, 2013. – С. 315–320.
3. Foley, J. Computer Graphics: Principles and Practice / J.D. Foley. – Addison-Wesley Publishing Company, 1996. – 1175 p.
4. Richards, J. Remote Sensing Digital Image Analysis: An Introduction / J.A. Richards. – Springer-Verlag Berlin Heidelberg, 2006. – 431 p.

ИСПОЛЬЗОВАНИЕ ЯЗЫКА JAVA ДЛЯ ПОСТРОЕНИЯ МУЛЬТИПЛАТФОРМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ ЦОД

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

Казак Н. П.

Искра Н. А. - магистр техн. наук, старший преподаватель

В современных ЦОД одной из главных задач является построение мультиплатформенной системы управления ресурсами. Требование мультиплатформенности для систем управления возникает потому, что серверный пул, часть ресурсов которого используется для функционирования систем управления, может быть реализован с использованием различных серверных платформ, функционирующих под управлением различных операционных систем. При решении данного класса задач одним из наиболее приоритетных инструментов является язык программирования Java.

Системы управления ресурсами подразделяются на два класса: внешние системы управления и интегрированные системы управления.

Можно выделить базовые требования к типовой интегрированной системе управления ресурсами:

1. Код должен выполняться на всех основных аппаратных платформах и работать в среде всех основных операционных систем.
2. Программное обеспечение системы управления должно запускаться на любом сервере либо компьютере без процедуры установки.
3. Программное обеспечение должно быть максимально изолировано, работа данного программного обеспечения не должна влиять на отказоустойчивость серверной платформы.
4. Программное обеспечение должно требовать для своей работы минимальное количество дополнительных программных продуктов.
5. Программное обеспечение системы управления должно поддерживать многопоточность для оптимального использования ресурсов мультипроцессорных многоядерных серверных систем.
6. Система управления должна взаимодействовать со всеми основными web-браузерами.

Рассмотрим преимущества языка программирования Java при реализации данного класса систем:

1. Язык программирования Java является мультиплатформенным, поэтому программное обеспечение написанное на нем выполняется на всех основных аппаратных платформах и может работать в среде всех основных операционных систем.
2. Программное обеспечение системы управления может выполняться на любом сервере. Единственным требованием является установка на данном сервере исполняющей системы Java, так называемой Java Runtime Environment (JRE).
3. Программное обеспечение реализованное на языке Java выполняется на обособленной виртуальной машине JVM и не взаимодействует напрямую с аппаратными ресурсами сервера и программными ресурсами операционной системы.
4. Для работы программного обеспечения написанного на языке Java требуется только установка Java Runtime Environment.
5. Одной из сильных сторон виртуальной машины Java является её способность работы с несколькими потоками. JVM оптимизирована для больших многоядерных машин, и она без проблем может управлять большим количеством потоков. Наиболее очевидная область применения многопоточности – это программирование интерфейсов систем управления.
6. Все основные web-браузеры имеют поддержку для работы с приложениями написанными на языке Java.

Перечислим основные задачи, решаемые с использованием языка программирования Java в данных системах:

1. Консоли управления серверами.
2. Модули доступа к внешним носителям информации.
3. Диалоговые модули управления подсистемами сервера.
4. Диалоговые модули управления инфраструктурой.

К данному классу систем относятся: HP Integrated Lights-Out, IBM Hardware Management Console (HMC), integrated Dell Remote Access Controller, Brocade SAN Fabric Manager, Cisco SAN switch manager и другие.

При реализации внешних систем управления есть определённые задачи, для решения которых использование языка программирования Java является наиболее оптимальным:

1. Построение универсального диалогового интерфейса, который функционирует под управлением различных операционных систем.
2. Реализация WEB-интерфейса приложений.
3. Реализация универсального интерфейса программирования приложений (API).
4. Реализация мультиплатформенных модулей сбора и анализа статистической информации.

Рассмотрим, как используется язык программирования Java при реализации таких систем на примере HP System Insight Manager, IBM Tivoli, Hitachi Tuning Manager.

Внешние системы управления можно разделить на несколько основных подсистем:

1. Диалоговая система управления;

2. Система контроля функционирования;
 3. Система сбора и анализа статистических данных.
 4. Универсальный интерфейс программирования приложений (API).
- Примеры использования языка программирования Java при реализации внешних систем управления:
1. При реализации диалогового интерфейса язык программирования Java используется во всех вышеперечисленных системах управления. Таки образом реализуется мультиплатформенный интерфейс и минимизируются затраты на адаптацию к различным операционным системам.
 2. При разработке подсистем контроля функционирования и сбора и анализа статистических данных компании HP и Hitachi использовали JDBC для организации связи Java-приложений с базой данных. Именно эта библиотека входит в комплект JavaBeans и поставляется в комплекте средств разработки JDK. Эта библиотека обеспечивает Java-программам универсальное средство для работы с СУБД.
 3. Для работы с своим API компания Hitachi использует язык JavaScript, а компания HP в качестве базового языка для своего API выбрала язык программирования Java.

Таким образом мы видим, что язык программирования Java нашел очень широкое применение при реализации систем управления центрами обработки данных и полноценного конкурента в этом сегменте при создании мультиплатформенных приложений у него пока нет.

Список использованных источников:

1. HP Integrated Lights-Out. User Guide. - Part Number 238882-004 – 2003 – 249 с.
2. IBM System i and System p. Managing the Hardware Management Console (HMC) – 1998 – 132с.
3. Integrated Dell Remote Access Controller 7 (iDRAC7) Version 1.50.50 User's Guide – 2014 – 258 с.
4. BROCADE Fabric OS. Administrator's Guide - 53-1002920-02 – 2013 – 694 с.
5. Cisco Fabric Manager Fundamentals. Configuration Guide - OL-21502-02 – 2010 – 338 с.
6. HP Systems Insight Manager 7.5. User Guide - Part Number: 601823-402a – 2016 – 257 с.
7. IBM Tivoli Monitoring Administrator's Guide - SC22-5446-01
8. Hitachi Command Suite. Tuning Manager. Server Administration Guide - MK-92HC021-38 – 2015 – 280 с.

РАСПРЕДЕЛЕННЫЙ АНАЛИЗ КРИПТОВАЛЮТНОГО BLOKCHAIN

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

Ващук Д.В.

Самаль Д. И. – к.т.н, доцент

В последнее время криптовалюты приобретают все большую ценность, и все больше финансовых организаций, продавцов и потребителей обращают внимание на них. В первую очередь это связано с преимуществами, которые дают криптовалюты перед фиатной валютой: отсутствующие или мизерные комиссии, анонимность, возможность быстро и беспрепятственно отправлять «криптоденьги» в любой уголок мира. Разработчики и финансисты нуждаются в системах, которые бы позволили быстро находить определенную информацию в Blockchain. Но разработать такую систему не так просто, так как приходится постоянно работать с большими объемами данных в Blockchain. Задача ускорения анализа криптовалютного Blockchain на примере криптовалюты Bitcoin была решена в рамках данной работы.

Bitcoin — полностью распределенная платежная система, которая использует одноименную расчетную единицу — биткойн. Система предоставляет возможность осуществления полностью необратимых денежных транзакций, когда денежный перевод происходит без доверенной стороны-посредника, и ни одна из сторон не могла бы заблокировать, отменить или принудительно совершить транзакцию. Для обеспечения функционирования и защиты системы используются криптографические алгоритмы, но вся информация о транзакциях всегда доступна в открытом виде, так как не шифруется [1]. Запущенные на множестве узлов в сети Internet программы-клиенты соединяются между собой в одноранговую сеть. Заранее известны объем и время эмиссии новых биткойнов. Но распределяются они случайно среди тех, кто использует свое оборудование для расчета хеш-кодов блоков, которые являются специфичным механизмом регулирования и подтверждения легитимности операций в системе Bitcoin — метод Proof-of-work [2].

Распределенную базу блоков транзакций, впервые реализованную в криптовалюте Bitcoin, называют Blockchain. Блок транзакций — специальная структура для записи группы транзакций в системе Bitcoin и аналогичных ей. Чтобы транзакция считалась подтвержденной, ее формат и криптографическую подпись должны проверить и затем группу транзакций записать в блок. Блоки одновременно формируются множеством агентов системы — так называемых, *майнеров*. Удовлетворяющие критериям блоки отправляются в сеть, включаясь в распределенную базу блоков. Копия Blockchain одновременно хранится на множестве компьютеров и синхронизируется согласно правилам построения цепочки блоков [2, 3].

		Debit	Credit
Transaction e14768c1d648b98a52cb796af30af186140c5209a2fb53f1c8097db579f01cc0			
INPUTS			
Previous Output	Signature		
6120ceaab25cfee257... : 0	3046022100aaf227f9...	0.0145	
6eb36c1d347f8fdb6e... : 1	3046022100810c9d7a...	0.0923	
OUTPUTS			
Address	Spent		
1NqUaJrFeStshjad1bhrEFFzWSQw6JHbqv	<input checked="" type="checkbox"/>		0.0122
1FrRypBwstUQ4X9KQdQByx6fWXLGGuPNT	<input checked="" type="checkbox"/>		0.0945
Transaction b6f4ec453a021ac561b01039f78e7168a653af176353c86d607343cc77e779b9			
INPUTS			
Previous Output	Signature		
e14768c1d648b98a52... : 0	30450221008a396b69...	0.0122	
OUTPUTS			
Address	Spent		
1HiMoMgBaAikFHgAt3M4YJtstp4HrnsiXu	<input type="checkbox"/>		0.001
1Q3Jw1wRxZyJ767mVrTEpBVTH49HNBA83V	<input checked="" type="checkbox"/>		0.0111

Рис. 1. Структура списка транзакций

На данный момент свободно распространяется не так уж много инструментов, с помощью которых можно проанализировать Blockchain и найти необходимую информацию. К задачам анализа Blockchain можно отнести: поиск списка переводов средств с определенного кошелька; поиск адресов кошельков, которые осуществляли транзакции на исследуемый адрес кошелька; сбор различного рода статистики по осуществлению денежных переводов в реальном времени. Как правило, ускорение анализа в найденных инструментах происходит благодаря предварительному индексированию Blockchain. Еще большего ускорения анализа можно достичь при распределении вычислений на несколько вычислительных ядер, а еще лучше – на несколько вычислительных машин. В контексте данной работы использовался функциональный язык программирования Elixir [4] и несколько одноплатных компьютеров с процессором ARMv7 Cortex-A и операционной системой Debian GNU/Linux. Общая структура кластера выглядит следующим образом:

- Несколько вычислительных узлов с большим объемом дискового пространства, предпочтительно SSD, используются для запуска *bitcoind* сервиса, который взаимодействует с остальными узлами Bitcoin сети, загружает и верифицирует блоки и транзакции, т.е. поддерживает Blockchain в актуальном состоянии. Помимо этого, данный сервис предоставляет RPC-JSON API для других приложений [1].
- Остальные узлы не нуждаются в больших объемах дискового пространства и используются для производства задач анализа блоков и транзакций, которые были прочитаны из Blockchain [3].
- Необходим еще один узел, который будет выступать в качестве супервизора. В задачи супервизора входит: формирование задач для каждого вычислительного узла и последующий сбор и объединение результатов.

Для полноценной работы с *bitcoind* JSON-RPC на языке Elixir необходимо было доработать Open source модуль Gold [5].

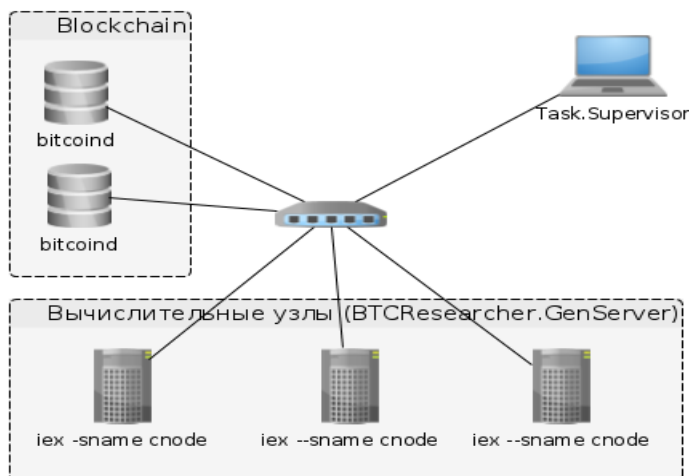


Рис. 2. Предлагаемая инфраструктура вычислительного кластера

Важной особенностью является то, что по умолчанию *bitcoind* индексирует транзакции только с *неиспользованными выводами* и транзакции, принадлежащие к текущему кошельку (если сервис собран с поддержкой функции кошелька, что не требуется в нашем случае). Неполная индексация приводит к тому, что некоторые JSON-RPC запросы *getrawtransaction* с корректным набором параметров могут завершиться с ошибкой. Индексация всех транзакций необходима для быстрого доступа к информации о любой транзакции по ее TXID. Для создания полного индекса транзакций необходимо запускать *bitcoind* с опцией *txindex* [1].

На каждом вычислительном узле запускается процесс виртуальной машины Beem VM командой *ix*. Для аутентификации узлов используется механизм *cookies*. Список узлов и *cookies* указываются в настройках супервизора.

Таким образом, описанное архитектурное решение дает существенный прирост в скорости анализа Blockchain и поиска необходимой информации в большом объеме данных. Ускорение происходит за счет распределения нагрузки на несколько вычислительных узлов и наличия в сети нескольких Blockchain, поддерживаемых в актуальном состоянии сервисом *bitcoind*.

Список использованных источников:

1. Bitcoin Developer Reference [Электронный ресурс]. – Режим доступа – <https://bitcoin.org/en/developer-reference>
2. Nakamoto, С. Bitcoin: A Peer-to-Peer Electronic Cash System [Электронный ресурс]. – Режим доступа – <https://bitcoin.org/bitcoin.pdf>
3. Так что же такое биткоин? [Электронный ресурс]. – Режим доступа – <http://bitnovosti.com/2014/07/17/tak-chto-zhe-takoe-bitcoin/>
4. Elixir: Getting started [Электронный ресурс]. – Режим доступа – <http://elixir-lang.org/getting-started>
5. Gold GIT repository [Электронный ресурс]. – Режим доступа – <https://github.com/denis4net/gold>

СИСТЕМА ПОЗИЦИОНИРОВАНИЯ И УПРАВЛЕНИЯ ПОЛОЖЕНИЕМ КВАДРОКОПТЕРА НА ПЛОСКОСТИ

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

Баранов Д. А.

Самаль Д. И. – канд. техн. наук, доцент

В современном мире наблюдается тенденция по все большей автоматизации рабочего процесса. В этом трудящимся помогают роботы, компьютеры и другие приборы, которые решают определенную задачу в автоматическом режиме почти без вмешательства людей.

Квадрокоптеры — это летательные аппараты, принцип передвижения в воздухе которых основан на вращающихся в противоположных направлениях парах винтов. Они уже нашли применение во многих сферах деятельности человека, таких как, например, охранная или транспортная. Цель данного исследования — расширить сферу использования квадрокоптеров за счет предоставления им возможности передвижения по плоскости.

Для решения поставленной задачи больше всего подходят квадрокоптеры Х- и Н-образного типа благодаря симметричному размещению винтов на корпусе. Математические модели движения таких квадрокоптеров уже хорошо изучены и описаны. Для того, чтобы такой дрон передвигался по плоскости, достаточно домножить всю систему уравнений, описывающую движение квадрокоптера в вертикальной и горизонтальной плоскости, на -1 . Тем самым, создается необходимая тяга, направленная в обратную сторону, нежели во время обычного взлета, что обеспечит прилипание дрона к поверхности, а сила трения между поверхностью и квадрокоптером не даст ему сползть вниз. Таким образом он может зафиксироваться на любой поверхности и продолжать находиться там на протяжении всего времени автономной работы (см. рисунок 1(слева)).

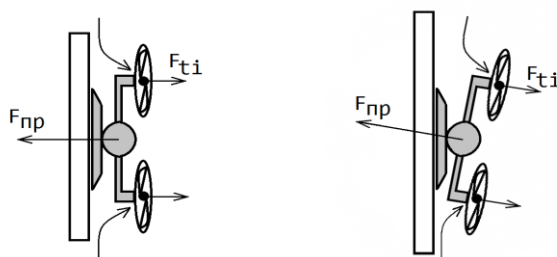


Рис.1 Направление сил, действующих на квадрокоптер в неподвижном режиме(слева) и при движении вверх(справа)

Для перемещения квадрокоптера по заданной поверхности необходимо слегка модифицировать посадочную конструкцию, и сделать ее не жесткой, а несколько наклоняемой по всем направлениям перемещения. Это требование связано с особенностью поворота квадрокоптеров в пространстве. Так, например, при повороте направо, дрон слегка наклоняется в правую сторону и т. д. (см. рисунок 1 (справа)).

Форма X- или H-образного квадрокоптера — симметричный квадрат со строго определенными правой, верхней, левой и нижней границами, поэтому для автоматизации перемещения описанного квадрокоптера будет использован, например, алгоритм Брезенхэма. Для этого плоскость, по которой предполагается перемещение, предварительно фиксируется камерой на управляющем устройстве, разбивается на мнимые «пиксели», размеры которых соответствуют размерам квадрокоптера (требуется для того, чтобы лопасти винтов дрона не могли повредиться при контакте с границами области перемещения), а затем решается задача построения кратчайшего пути по заданным «пикселям». После этого управляющее устройство передает команды на перемещение квадрокоптеру. Дальнейшее участие управляющего устройства не требуется, т. е. Процесс перемещения полностью автоматизируется после фиксирования поверхности для перемещения. Подобным способом можно решить и задачи покрытия квадрокоптером заданной плоскости.

Таким образом, с помощью небольших изменений в конструкции квадрокоптера и его системе управления можно заставить его не только летать, но и ползать по стенам, а достаточно простые алгоритмы позиционирования дрона на дискретной плоскости позволят автоматизировать его передвижение по плоскости. Данная функция может помочь при решении таких задач, как покраска стен домов, мойка окон, установка камер слежения на стенах зданий и других прикладных задач, где необходимо участие человека на высоте, что уменьшит риски рабочих в этих областях.

Список использованных источников:

1. А. Е. Гурьянов Моделирование управления квадрокоптером / Электронный научно-технический журнал «Инженерный вестник», стр. 522 -534 // Издательство: ФГБОУ ВПО «МГТУ им. Н.Э. Баумана» ISSN 2307-0595; 2012 г. – 548 с.
2. Канатников А. Н., Акопян К. Р. Управление плоским движением квадрокоптера / Сетевое научное издание «Математика и математическое моделирование», стр. 23-36 // Интернет-ресурс: DOI: 10.7463/mathm.0215.0789477

СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ УРОВНЯ ГИБРИДНОСТИ СЕМЯН ПО ЭЛЕКТРОФОРЕГРАММЕ

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

Остроухова С.А.

Татур М. М. – д-р. техн. наук, профессор

Определение типичности гибридов – важная часть анализа семян, которая предотвращает попытки фальсификации семенного материала и предупреждает подмешивание низкокачественных семян к оригинальным гибридам. Анализ электрофореграмм проводится визуально и является довольно сложным для человека. Применение компьютерного зрения позволяет повысить объективность и продуктивность анализа.

Использование семян одного вида позволяет уменьшить невыровненность посевов, избежать разных сроков созревания для одной партии семян и разной восприимчивости растений к метеорологическим факторам и болезням. Таким образом, анализ качества гибридов влияет на сокращение потери прибыли.

Типичностью (гибридностью) растений называют соответствие заявленной партии семян стандартному (типичному) семенному материалу. Эта величина определяется в процентах. Методика проведения анализа уровня гибридности семян основана на том, что каждый белок является обязательным составляющим любой клетки. Использование белковых маркеров в настоящее время является наиболее объективным способом для установления подлинности, типичности самоопыленной линии. Электрофоретический анализ белков является универсальным методом в семеноводстве для осуществления контроля качества семян [1].

В белорусских государственных инспекциях по семеноводству для определения уровня гибридности семян используют прибор фирмы Bio-Rad. Системы Gel Doc XR+ характеризуется высокой чувствительностью и высоким разрешением для обработки образцов широкого спектра. В прибор помещается окрашенная гелевая пластинка. Система состоит из темной камеры, CCD-камеры, программно-управляемых линз, источников УФ и белого света, фильтров и защиты от УФ излучения. Получаемые изображения могут использоваться для составления отчетов, поэтому система предоставляет снимки в высоком качестве. Разрешение изображений составляет 4 МП. Предусмотрена подсветка, работающая в трёх режимах: УФ и белый транс-иллюминаторы, белый эпи-иллюминатор [2].

С прибором Bio-Rad поставляется программное обеспечение для определения уровня экспрессии гибридного белка. В предоставляемом программном обеспечении можно выбрать настройки получаемого изображения (яркость, контрастность), пример которого изображения приведён на рисунке 1.

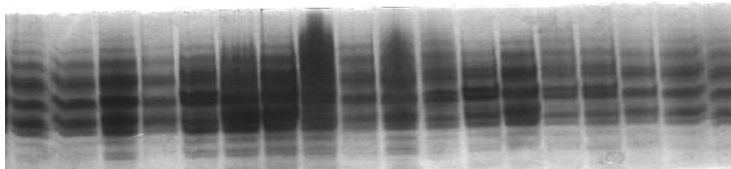


Рисунок 1 - Изображение окрашенного геля для определения гибридности семян

Результаты, получаемые с помощью этого прибора, являются воспроизводимыми, точными и упрощают последующий анализ изображений, поскольку изображения имеют одинаковый масштаб и сделаны камерой с фиксированными настройками.

Для обработки результатов специалисты используют эталонные электрофореграммы или получают электрофореграммы эталонных семян. Каждая полоска соответствует одному семечку и сравнивается с эталоном. При определении сортовой принадлежности семян по электрофореграмме используется формула, описывающая эталонный образец. На анализируемый образец накладывается сетка, и по ней определяется положение и ширина окрашенных полос. Значения сравниваются с формулой эталона.

Для определения гибридности по электрофореграммам с помощью системы технического зрения предлагается использовать такой же подход с применением размеченной сетки и формулы эталонного образца. Так как все изображения получены с помощью одного и того же прибора в одинаковых условиях, то масштаб у изображений совпадает и необходимо только определить начальную точку отсчёта для наложения сетки.

Поскольку системы технического зрения более объективны, чем обычный специалист, и имеют превосходство над человеческим зрением, то они хорошо подходят для решения поставленной задачи определения гибридности семян по электрофореграммам.

Список использованных источников:

1. Семена кукурузы. Метод определения гибридности семян первого поколения, оценка типичности и маркирование инбредных линий [Текст]: СТБ 1710 – 2006. – Введ. 2006-30-12. – М.: Госстандарт, 2006. – 10с.
2. Gel Doc XR System | Научные исследования | Bio-Rad. – (http://www.bio-rad.com/ru-ru/product/gel-doc-xr-system?pcp_loc=catprod).

ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ ВЗАИМОДЕЙСТВИЯ В СИСТЕМАХ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

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

Бычко А. А.

Татур М. М. – д. т. н., профессор

Системы электронного документооборота и смежные с ними (ERP – Enterprise Resource Planning [1] и PLM – product lifecycle management [2]) являются одним из наиболее часто разрабатываемых в настоящее время в РБ систем [3]. Однако, несмотря на широкое распространение и отлаженные технологии проектирования, данным системам присущ ряд общих недостатков, затрудняющих их использование.

Все существующие на рынке системы электронного документооборота построены по схожим принципам, диктуемым используемыми в настоящее время методологиями, инструментами разработки и типичным программным окружением, настроенным у пользователя. Как правило, система, автоматизирующая документооборот, состоит из трёх частей: уровень доступа к данным (DAL – data access layer), бизнес-логика приложения (BL – business logic) и уровень представления (PL – presentation layer) или пользовательского интерфейса. Системы строятся по трёхкомпонентной схеме: хранилище данных, серверная часть и клиентская часть.

Конечный пользователь, как правило, взаимодействует только с клиентской частью программы и не имеет доступа к инкапсулированным механизмам бизнес-логики. Наиболее распространёнными интерфейсами клиентской части являются оконные приложения («тонкие» либо «толстые» клиенты), мобильные и web-приложения. Консольные терминалы в настоящее время вышли из широкого употребления и сохраняются лишь в узкоспециализированных инструментах администрирования. Для каждого из этих вариантов интерфейсов существует множество разнообразных реализаций: WinAPI, WinForms, GTK, WPF, ASP, JSP и многие другие. Существуют также стандарты [4], регламентирующие элементы пользовательского интерфейса и их организацию и взаимодействие.

В то же время ни в существующей практике, ни в стандартах, не упоминается напрямую механизм взаимодействия пользователя с программой. Пренебрежение проектированием взаимодействия вызывает

рост когнитивного сопротивления [5] разрабатываемого программного обеспечения, что, в свою очередь, приводит к эпизодическим непродуктивным тратам времени при работе с программой, повреждению данных и росту раздражения пользователей.

Основные причины роста когнитивного сопротивления [5]:

- неадекватная оценка уровня квалификации пользователя;
- перегрузка пользовательского интерфейса элементами управления;
- неверное истолкование целей, к достижению которых стремится пользователь.

В той же работе, посвящённой преодолению когнитивного сопротивления, предложены следующие возможные решения:

- чёткое определение ролей пользователей и требуемого им функционала;
- максимальное упрощение интерфейса программ;
- проектирование взаимодействия должно осуществляться на ранних этапах разработки;
- все предположения разработчиков должны проверяться в тестовой эксплуатации.

В работе [6] был описан опыт разработки и сопровождения систем электронного документооборота в государственных учреждениях Республики Беларусь, обобщение которого полностью подтверждает выдвинутые тезисы. В настоящее время в разработке находится система автоматизации документооборота для Государственных инспекций по семеноводству, карантину и защите растений, которая проектируется и реализуется с использованием описанных принципов.

Список использованных источников:

1. O'Leary, Daniel L. Enterprise resource planning systems. — Cambridge University Press, 2000. — 232 с. — ISBN 0-521-79152-9.
2. Стандарт ИСО 9004-1-94. Управление качеством и элементы системы качества (п.5.1.1).
3. Мировой рынок систем электронного документооборота [Электронный ресурс] / Мировой рынок систем электронного документооборота. — Режим доступа: <http://citforum.ru/consulting/docflow/market/article1.8.200222.html>. — Дата доступа: 02.10.2015.
4. СанПиН 9-131 РБ 2000. "Гигиенические требования к видеодисплейным терминалам, электронно-вычислительным машинам и организации работы".
5. Alan Cooper. The inmates are running the asylum – Sams Publishing, USA, 2004.
6. Бычко, А. А. Автоматизированная система электронного документооборота с низким когнитивным сопротивлением / А. А. Бычко // Молодёжь в науке - 2015, Минск, Национальная Академия Наук Беларуси, Совет молодых ученых НАН РБ. — Минск, 2015.

РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА ОДНОСЛОЙНОГО ПЕРСЕПТРОНА ДЛЯ ПЛАТФОРМЫ OPENCL

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

Жабинский С. В.

Татур М. М. - д-р техн. наук, профессор

С увеличением объёма обрабатываемых современными информационными системами данных возникает необходимость в их эффективном интеллектуальном анализе. В докладе рассматривается реализация параллельного алгоритма однослойного персептрона для платформы OpenCL, обеспечивающей выполнение программы на графическом процессоре, обладающем высокой степенью параллелизма.

Одним из простейших алгоритмов классификации является однослойный персептрон [1]. Он относится к алгоритмам с обучением с учителем и позволяет производить линейную классификацию. Принцип работы однослойного персептрона основан на модели функционирования нервной клетки — искусственного нейрона. Однослойный персептрон является нейронной сетью с одним скрытым слоем нейронов. Схема однослойного персептрона представлена на рисунке 1.

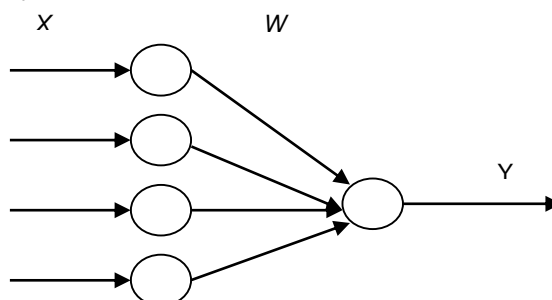


Рис. 1. Схема однослойного персептрона.

X — вектор входных значений, W — вектор весовых коэффициентов, Y — выходное значение. Выходной нейрон производит взвешенное суммирование входных значений и принятие решения о выходе персептрона. Как правило, если полученная сумма больше, либо равна 0, выход принимается равным 1, в противном случае — 0. Обучение нейрона заключается в поиске весовых коэффициентов W .

Алгоритм обучения однослойного персептрона выглядит следующим образом:

1. Вектор весовых коэффициентов W инициализируется небольшими случайными величинами.
2. Для каждого значения x_j из обучающей выборки:
 - 2.1. Определяется выход нейрона y :
 - а) Вычисляется сумма s произведения каждого признака из входного вектора признаков на соответствующий весовой коэффициент w_i .
 - б) К сумме применяется функция активации. В качестве функции активации, как правило, применяют ступенчатую функцию, принимающую значение 1, если $s \geq 0$, и значение 0, если $s < 0$.
 - 2.2. Каждый весовой коэффициент обновляется по формуле: $w_j = w_j + (l_i - y_j) * x_{ij}$, где l_i — класс, к которому относится данное наблюдение.
3. Вычисляется ошибка ϵ обучения на текущем шаге:
 - 3.1. Для каждого наблюдения из обучающей выборки вычисляется выход y .
 - 3.2. Разница между выходом персептрона y и истинным значением из обучающей выборки добавляется в сумму S .
 - 3.3. Ошибка ϵ вычисляется как S/N , где N — количество элементов в обучающей выборке.
4. Если ϵ меньше выбранного порогового значения, обучение прекращается. В противном случае повторяются шаги 2 — 4.

Заметим, что данный алгоритм может быть эффективно реализован параллельно. Для параллельной реализации будем использовать аппаратно-программную платформу OpenCL [2]. Данная платформа позволяет производить вычисления общего назначения на графическом процессоре [3].

Рассмотрим особенности реализации алгоритма однослойного персептрона для платформы OpenCL.

Платформа OpenCL относится к гетерогенным вычислительным системам. Управляющим элементом системы является центральный процессор (ЦПУ). В качестве ускорителя вычислений используется графический процессор (ГПУ). Общая схема такой системы представлена на рисунке 2. ЦПУ передаёт данные и команды в память ГПУ и получает обратно результат вычислений. Как правило, «бутылочным горлышком» таких систем является передача данных из основной памяти компьютера в память ГПУ. Поэтому следует избегать большого количества подобных операций. Также по этой причине, при малых объёмах обрабатываемых данных скорость выполнения параллельного алгоритма может быть меньше, чем скорость последовательного алгоритма, выполняющегося на ЦПУ.

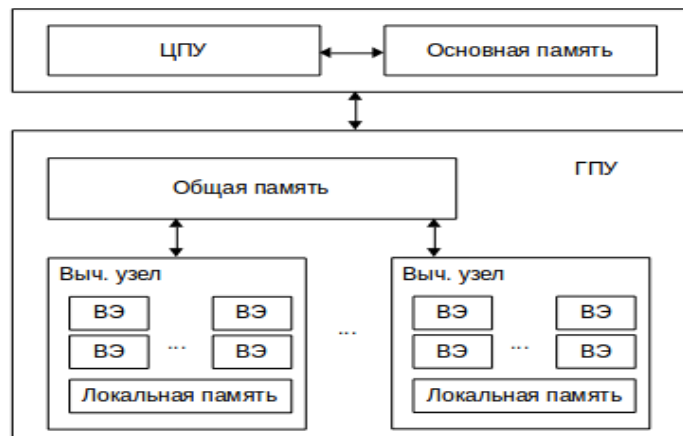


Рис. 2. Общая схема гетерогенной системы для вычислений общего назначения на ГПУ.

Для проверки эффективности параллельной реализации для платформы OpenCL были реализованы последовательный алгоритм (выполняющийся на ЦПУ), и параллельный (для OpenCL). Графики с временем обучения для данных реализаций представлены на рисунке 3.

На скорость обучения однослойного персептрона влияют размер обучающей выборки и размер пространства признаков. Тестирование производилось с различными значениями данных параметров для определения масштабируемости реализации, то есть изменение времени обучения от объёма вычислений.

Как видно из графиков, с ростом объёма вычислений, время обучения растёт практически линейно. Однако, при использовании параллельного алгоритма наклон прямой более пологий. Также следует обратить внимание на точку пересечения графиков. Данная точка показывает пороговое значение объёма данных, при котором эффективность использования реализации для OpenCL становится выше классической реализации для выполнения на ЦПУ.

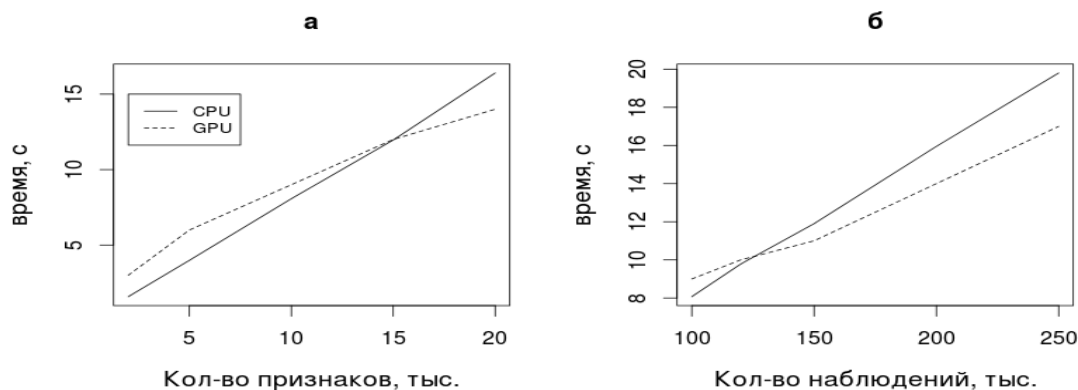


Рис. 3. Время обучения однослойного перцептрона.

Список использованных источников:

1. Беркинблит М. Б. Нейронные сети. / М. Б. Беркинблит. - Москва: МИРОС и ВЗМШ РАО, 1993. - 96 с.
2. Stone, J. E. OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems / J. E. Stone, D. Gohara, G. Shi // Computing in Science & Engineering. - 2010. - Vol. 12, iss. 3. - p. 66-73.
3. Keckler, S. GPUs and the Future of Parallel Computing. / S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, D. Glasco // IEEE 2011. - Vol. 31, № 5 - p. 7-17.

МЕТОДЫ СИНТЕЗА ЗНАНИЙ, ОПЫТА И РЕСУРСОВ ПОЛЬЗОВАТЕЛЕЙ В КОНТЕКСТЕ «ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СОВМЕСТНОЙ РАБОТЫ»

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

Краснов А.Ю.

Воронов А. А. – к-т. техн. наук, доцент

Тема доклада раскрывается в рамках магистерской диссертации: «Алгоритмы объединения и визуализации ресурсов пользователей на примере музыкальной социальной сети». Цель магистерской работы - разработка метода объединения ресурсов пользователей и интеграция данного метода в веб-приложение «социальная музыкальная сеть».

Актуальность исследования обусловлена развитием проектных форм организации работ (проектного менеджмента), которые являются ответом на такие вызовы времени как ускорение темпов научно-технического прогресса, и высокая скорость социально-экономических изменений [1]. Таким образом, возникает необходимость оперативного группового обсуждения ситуации и проблем, в условиях отсутствия экспертов извне, а также недостатка времени. Это обуславливает использование знаний, опыта и ресурсов ограниченного числа людей в рамках решения совместных задач. Исходя из этого можно сделать вывод, что разработка технологии совместной работы и объединения ресурсов пользователей является актуальной.

Наиболее близким аналогом нашего программного продукта будет класс «программного обеспечения для совместной работы» (groupware) – тип программного обеспечения (далее – ПО), созданного с целью поддержки взаимодействия между людьми, коллективно работающими над решением общих задач [2]. К данному типу программного обеспечения применим закон Меткалфа, который гласит, что полезность сети пропорциональна квадрату численности пользователей этой сети. Таким образом, чем больше людей используют что-либо, тем более ценным оно становится.

Ниже приведена классификация «программного обеспечения для совместной работы» (далее – ПОСР):

- совместная работа над документами (Google Docs, Office Online);
- вики технологии [3] (Wikipedia);
- системы управления проектами (Jira, eGroupWare, Citadel, Microsoft Project);
- аудио, видео конференции (Skype, Skype For Business, Google Hangouts,);
- облачные хранилища данных с возможностями совместной работы над файлами (Dropbox, OneDrive, Яндекс-диск);
- корпоративные социальные сети (Битрикс24, DaOffice, Jive, Yammer);
- базы знаний[4] (в особенности экспертные системы) (CLIPS, WolframAlpha, MYCIN);
- интернет-опросы и голосования (Examinare, Surveymonkey, Testograf);

- менеджеры задач (LeaderTask, WonderList, Google Tasks).

Нами был разработан собственный метод синтеза знаний, опыта и ресурсов, использующий технологии, присущие вышеприведённым ПОСР:

- создания сообществ разными пользователями;
- разрушения иерархических границ между специалистами предприятия;
- организации контента: тэги, закладки;
- коллективного обсуждения (форумы, чаты);
- голосований, опросов, рейтингов;
- совместного редактирования контента в реальном режиме времени;
- контроля и мониторинга состояния системы (возможность подписки на обновления в системе, слежение за новостями в системе);
- публикации различного типа контента (ссылки, документы, изображения, аудио и видео);
- просмотра истории всех изменений в элементах системы.

Цель метода: синтезировать ресурсы, знания и опыт пользователей.

Задачи метода:

- разделить этапы структурирования области сбора информации и этапы заполнения этой области ресурсами;
- иерархически структурировать область сбора информационных ресурсов (построить «конфигуратор» - древовидную иерархию, структурирующую данную область);
- создавать условия для быстрого, удобного и последовательного редактирования любого элемента системы заинтересованным сообществом пользователей;
- осуществить отбор информации, опираясь на мнение большинства участников группы;
- произвести слияние ресурсов пользователей, по критериям значимости для данной области знания;
- предоставить результат слияния в удобной визуальной форме.

Последовательность этапов работы сообщества по синтезу знаний:

1. Определение области, по которой будет осуществляться сбор информации, синтез знания и опыта;
2. Определение участников;
3. Построение конфигуратора:
 - 3.1. Выдвижение каждым участником структур раздела конфигуратора первого уровня. Голосование и определение победителя;
 - 3.2. Дополнение и редактирование победившего предложения путем внесения предложений и голосований по данному разделу;
 - 3.3. Обсуждение процесса редактирования каждого из разделов;
 - 3.4. Структурирование разделов путем определения подразделов и редактирования их выше описанными методами (глубина иерархии определяется участниками, нижние уровни иерархии предназначены для их непосредственного заполнения содержимым);
4. Наполнение содержим нижних ячеек конфигуратора (файлами, текстами, картинками документами, ссылками):
 - 4.1. Внесение участниками содержимого разного типа;
 - 4.2. Голосование и определение лучшего «связывающего текста» (взятие его за основу) и прилагаемых к нему «файлов-ресурсов» (под «связывающим» подразумевается текст, содержащий ссылки на «файлы-ресурсы» и таким образом объединяющий весь контент);
 - 4.3. Редактирование «связывающего текста» и прилагаемых к нему «файлов-ресурсов»;
 - 4.4. Повторение вышеприведенной процедуры для пункта «важные дополнительные материалы», т.е. конструирование «связывающего текста» и сборка прилагаемых к нему «файлов-ресурсов».

Пример работы метода синтеза знаний, опыта, ресурсов в контексте встраивания его в функционирование «музыкальной социальной сети».

Один из участников сети создает новую сессию и определяет тему, по которой будет производиться синтез ресурсов (предположим, что темой сессии будет сбор информации о музыкальной группе). После этого он добавляет необходимых пользователей сети (например, друзей) в эту сессию, наделяя их правом сделать аналогичное предложение, например, своим друзьям.

Следующим этапом является построение конфигуратора, для этого участники должны вынести на рассмотрение сообщества структуры разделов первого уровня. После внесения предложений производится голосование и определяется победитель, группа переходит к рассмотрению и редактированию победившей структуры. Во время редактирования каждый участник может предложить изменить существующий раздел, удалить его или добавить новый раздел. После предложений запускается голосование и по его итогам определяется судьба раздела. В процессе голосования по каждому разделу идет обсуждение в чате. Так как чаты разделены по разделам, общий информационный поток обсуждения не перегружается, чаты разделены и структурированы, удобны для работы. Каждый отредактированный раздел содержит историю изменений с набором всех предложений, итогами голосований за них, историей обсуждений (чаты) по каждому предложению.

Предположим, что разделами первого уровня выбраны: история группы, дискография, состав группы, концерты, статьи о группе, интересные факты. Далее по каждому из разделов запускается аналогичная

процедура которая позволяет уточнить из каких подразделов он состоит и разделить его на более мелкие структурные единицы.

Когда иерархия готова участники начинают заполнять нижние уровни (листья дерева). К этому моменту раздел дискография может состоять из подразделов с конкретными названиями альбомов группы, а раздел интересные факты может быть разделен по временным отрезкам (70-ые гг., 80-ые гг. и т. д.).

В процессе наполнения ресурсами участники вносят свои материалы в виде «связывающего текста» и «файлов ресурсов», прилагающихся к нему. Таким образом, в итоге у них получается текст по данному разделу со ссылками либо на сторонние ресурсы, либо на загружаемые ими (файлы, картинки, аудио или видео). После того как группа выбрала лучший текст из предложенных по данному тексту, строится итоговое представление, которое содержит результирующий «связывающий текст» для данного раздела. Далее участники могут работать над ним делая правки, удаления или добавления, прикрепляя новые ресурсы и так далее.

Выводы. Разработанный нами метод синтеза знаний, опыта и ресурсов позволяет заинтересованным сообществам, члены которых не могут общаться напрямую, использовать компьютерные технологии для наиболее эффективного общения, целью которого может быть обмен, анализ, слияние и синтез информационных ресурсов участников. В результате работы они получают структурированную и заполненную иерархию разделов по выбранной теме, которая может в перспективе дорабатываться и наполняться новым содержимым. Участники могут подписываться на обновления в системе и просматривать историю модификации разделов. Созданный участниками конфигуратор может использоваться в качестве основы для последующих сессий сбора информации по аналогичным темам.

Список использованных источников:

1. М. К. Румизен Управление знаниями. Полное руководство. – М: «АСТ, Астрель», 2004. – 317 с.
2. Groupware [Электронный ресурс]. – Режим доступа: <http://nexus.awakentech.com:8080/at/awaken1.nsf/UNIDs/CFB70C1957A686E98825654000699E1B?OpenDocument>
3. Иванов Д. Технология Вики / Д. Иванов, П. Смирнов [Электронный ресурс]. – Режим доступа: https://dulanov.wordpress.com/2004/06/01/tehnologiya_viki/
4. Гаврилова и др. Базы знаний интеллектуальных систем // Учебник для вузов. — СПб. Питер, 2000.

ПРОГРАММНЫЙ МОДУЛЬ РАСПОЗНАВАНИЯ КРЕДИТНЫХ КАРТ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ

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

Матюшевский Е.В.

Лукашевич М. М. – канд. техн. наук

Распознавание данных с кредитной карты одновременно является достаточно сложной алгоритмической задачей, но на данный момент реализация данной задачи весьма актуальна и востребована в связи с увеличением количества платежных транзакций через мобильные устройства. Реализация данной задачи способна избавить человека от надобности вводить большую часть данных при осуществлении интернет-платежей.

Банковская карта — это сложный документ типового размера (85,6 × 53,98 мм), содержащий набор полей, требуемых для оплаты: номер карты, дата выдачи (истечения), имя держателя. Большинство полей (за исключением CVC/CVV кода) находятся на лицевой стороне, поэтому основной фокус должен быть на распознавании лицевой стороны. Для большинства платежей требуется распознать на номер карты, имя держателя и срок действия карты. CVC/CVV код состоит из 3-4 символов, и его распознавание не является основополагающим для данного модуля, поскольку не оказывает существенного прироста ввода данных в приложение.

В качестве первого этапа необходимо найти координаты углов карты. Геометрические характеристики карты известны и неизменны (все карты выполнены строго в соответствии с стандартом ISO 781), поэтому для определения четырехугольника используется алгоритм поиска и перебора прямых. Для этого необходимо, чтобы карта была целиком в области зрения камеры и занимала большую часть кадра. Далее, с помощью высокочастотной фильтрации захваченного кадра подавляются шумы изображения. Для данной задачи имеет смысл использовать фильтр Гаусса:

$$y(n) = \sum_{i=0}^P b_i x(n-i) - \sum_{k=1}^Q a_k y(n-k)$$

На полученном изображении с помощью оператора Кэнни можно получить начальную карту границ:

$$K = \frac{1}{150}L = \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

В дальнейшем, после связи границ в совокупность точек, образующие прямые линии, требуется отсеивать прямые, не относящиеся к карте (данную задачу облегчает изначальное знание отношения сторон и их параллельности). На найденном четырехугольнике требуется применить проективное преобразование, приводящее изображение карты к ортогональному виду с фиксированным разрешением. После этого в наличии имеется область изображения, на котором находятся все интересные поля и можно приступать непосредственно к поиску и распознаванию текстовой информации.

В первую очередь, необходимо отсеять фон карты предварительной фильтрацией без подавления текстовой информации. Затем следует выделить зоны для поиска интересующей информации, чтобы увеличить быстродействие поиска и минимизировать ошибки. В данную задачу можно включить сегментацию изображения на области символов и отсеивание остальных областей, получив своеобразные «коробки символов» (прямоугольные области, интересные для распознавания текста).

Основной интерес и наибольшую сложность представляет из себя распознавание символов в «коробках символов». Для этого требуется локализовать строки символов, после чего строки-кандидаты следует сегментировать на символы в вертикальной ориентации. Этот момент является краеугольным, поскольку нахождение строки не является принципиально сложной (многочисленность символов в строке сама по себе является уточняющим фактором, уменьшающим вероятность ошибки, тогда как вертикальная сегментация работает с одним символом).

В случае успешной сегментации и выделения всех интересных символов, требуется их непосредственное распознавание. Для этого следует каждый интересный объект пропустить через нейронную сеть, предварительно обученную на выборке символов. Так как разрабатываемый модуль предполагает распознавание как embodded (выбитых), так и напечатанных символов, то требуется обучение как минимум 2 различным шрифтам. В случае невозможности распознать какой-либо символ, можно использовать дополнительный анализ нескольких последующих кадров, который сработает значительно быстрее, так как известна позиция поиска, и можно отсеять успешно распознанные символы.

После непосредственного анализа требуется применение постобработки, как использование алгоритма Луна для выявления ошибок распознавания, сравнение распознанной даты истечения на предмет актуальности текущей даты, возможно применение словарей имен и фамилий для уточнения.

Список использованных источников:

1. Canny Edge Detection [Электронный ресурс]. – Электронные данные. – Режим доступа: www.cvmt.dk/education/teaching/f09/VGIS8/AIP.
2. Shapiro, L. G. & Stockman, G. C. Computer Vision. — Prentence Hall, 2001. — С. 137, 150.

СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ОБРАБОТКИ ДАННЫХ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

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

Миранович И. А.

Иванов Н. Н. – канд. физ.-мат. наук, доцент

В сообщении описано решение задачи обнаружения и распознавания объектов бортовым оптико-электронным модулем беспилотного летательного аппарата. Задача решена алгоритмом глубинного обучения на основе сверточной нейронной сети (CNN). Задача решается в режиме реального времени.

Выбор метода для обнаружения и распознавания объектов на изображениях в режиме реального времени с учетом конкретной архитектуры беспилотного летательного аппарата (БПА) остается актуальной задачей. Специфика этой задачи в том, что большая часть известных методов работает только с «идеальным» изображением, что практически невозможно ввиду шумов при передаче видеопотока оператору, вибрация камеры в полете, влияния погодных условий, потерей кадров при передаче с удаленного БПА и т. п.

Глубинное обучение является областью более широкого машинного обучения. Глубинное обучение основано на двух постулатах: использованию большого количества уровней представления информации,

необходимых для моделирования семантических отношений в данных; и применении немаркированных данных (не включенных в процесс обучения). Глубинное обучение продемонстрировало свои преимущества в области машинного обучения, значительно превзойдя другие модели. Особенно полезным оно оказалось при решении задач распознавания речи и изображений и ряде других [1]. Недостатком глубинного обучения являются затраты ресурсов большого объема.

Целью сообщения является применение сверточной нейронной сети в режиме реального времени для глубинного обучения распознавателя прямоугольных движущихся объектов в зашумленном видеопотоке.

Отличительными чертами предлагаемого метода являются следующие аппаратные и программные особенности системы распознавания:

- 1) Все вычисления выполняются на борту БЛА, это стало возможным благодаря NVIDIA Jetson TX1.
- 2) Работа сверточной нейронной сети ускорена за счет предобработки изображения.
- 3) Для реализаций алгоритма использовалась библиотека Caffe.

В аппаратной части применяется NVIDIA Jetson TX1 – это модульный суперкомпьютер, обладающий высокой производительностью и энергоэффективностью. Он создан в 2015 году на основе архитектуры NVIDIA Maxwell, имеет 256 ядер CUDA, что обеспечивают производительность не менее, чем 1 терафлопс. Его 64-битный CPU способен кодировать/декодировать 4KB видео и интерфейс камеры со скоростью сенсора 1400 МП/с [2]. Этот суперкомпьютер идеально подходит для поставленной задачи.

Библиотека Caffe хранит данные в четырехмерных массивах, которые предоставляют интерфейс для хранения серий изображений, их параметров и данных об обновлении этих параметров. Операции над этими данными проводятся посредством взаимодействия CPU и GPU, что несколько замедляет обработку. Библиотека Caffe позволяет снизить временные затраты посредством синхронизации работы устройств «по требованию» [3].

Система детектирования объектов R-CNN впервые была представлена в 2014 году в [4] и состоит из трех модулей. Первый модуль генерирует гипотезы о местоположении объекта на изображении независимо от класса объекта. Второй модуль представляет собой большую сверточную нейронную сеть, которая извлекает признаки из каждой гипотезы, преобразуя их в вектор фиксированной длины. Третий модуль – это машина опорных векторов, осуществляющая классификацию векторов-признаков на наборе конкретных классов объектов.

Сеть содержит слои 3 видов: сверточные, подвыборочные и полносвязные. Подвыборочные слои служат лишь для масштабирования, их веса постоянны. Сверточные слои служат для нахождения признаков. Полносвязные слои представляют собой обычный перцептрон с количеством нейронов, совпадающим с количеством классов идентификации, и служит для финальной классификации.

Выбранная сфера применения алгоритма ставит одной из главных целей обнаружение и распознавание движущихся объектов. Это существенно сужает круг детектируемых объектов.

Главная идея заключается в выделении только движущиеся объекты, что существенно сокращает количество генерируемых гипотез. Метод обрабатывает данные путем фильтрации исходного изображения на основе его изменений во времени; на исключения гипотез, для которых можно сделать вывод, что они не относятся ни к одному из целевых объектов; на основании размера объекта и его формы на изображении. Пример работы метода выделения движущихся объектов приведен на рис. 1.

Таким образом, предлагаемое изменение исходного алгоритма сверточной нейронной сети состоит в следующем:

Предварительное выделение фона от новых областей на изображении. Этот этап приводит к значительному уменьшению количества обрабатываемых данных и времени, затрачиваемого на генерацию гипотез.

Игнорирование слишком больших и слишком малых объектов, которые рассматриваются как разновидности шумов на изображениях.



Рис. 1 – Выделения движущихся объектов.

Были выполнены прогоны алгоритма на рабочем стенде. В результате оказалось, что получено 12% ускорения обработки по сравнению с применяемыми методами. Это стало возможным ввиду применения сверточной нейронной сети, существенного сокращения площади поиска объектов на изображениях и использованию параллельных вычисления на GPU. Большую роль сыграла высокая производительность применяемой вычислительной техники.

Список использованных источников:

1. Пильгун В. М. Глубинное обучение нейронных сетей и достижения в их применении, Киев, 2015.
2. «Модуль JETSON TX1 Самая технически продвинутая в мире система визуальных вычислений для встраиваемых систем» [Электронный ресурс]. – URL: <http://www.nvidia.ru/object/jetson-tx1-module-ru.html> (дата обращения: 07.12.2015).
3. «Caffe is a deep learning framework» [Электронный ресурс]. – URL: <http://caffe.berkeleyvision.org/> (дата обращения: 08.12.2015).
4. Ross Girshick Rich feature hierarchies for accurate object detection and semantic segmentation / R. Girshick, J.Donahue,
5. T. Darrell, J.Malik – IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ SAAS ДЛЯ ИССЛЕДОВАНИЙ В ОБЛАСТИ ЯДЕРНОЙ ФИЗИКИ

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

Солодкий Д.М., Чистяков М.Ю., Супринович И.Ю.

Ташлыкова-Бушкевич И. И. – доцент

В статье рассматривается проблематика современных исследований в ядерной физике. Модель SaaS представлена в качестве архитектурного решения для программного комплекса обработки результатов исследований, полученных методом резерфордского обратного рассеяния. Обоснована актуальность данной модели и ее преимущества.

Фундаментальная наука редко бывает привлекательным клиентом для производителей программного обеспечения. Зачастую продуктовые IT-компаний ориентированы на производство приложений для массового рынка. Узкоспециализированные научные приложения в целом не являются коммерчески выгодными за исключением тендеров и прямых закупок. Тем не менее, именно научные сотрудники более других потребителей нуждаются в высокотехнологичном и современном ПО — это напрямую влияет на эффективность исследований и косвенно — на научный прогресс в целом.

В области физики твердого тела есть раздел исследований композиционного состава твердых тел методом ионной спектроскопии. Одним из способов изучения является метод резерфордского обратного рассеяния (РОР). Это современный неразрушающий метод изучения строения вещества, в основе которого положены законы сохранения энергии и импульса. Поток ускоренных частиц рассеивается на атомах исследуемого вещества и детектируется специальными приборами [1]. Данные, полученные детекторами

частиц, представляют собой энергетический спектр, который преобразуется в спектр обратного рассеяния. Зная начальные характеристики эксперимента, можно определить композиционный состав вещества путем обработки спектра. В результате расчетов исследователям доступна ключевая информация о строении образца: элементарный состав, виды примесей, их содержание и распределение по глубине. Однако ручная обработка спектров занимает достаточно продолжительное время. Автоматизация и использование вычислительной техники при исследованиях методом POP ощутимо ускоряет и упрощает процесс обработки данных, уменьшая вероятность ошибки. Анализ современного программного обеспечения по проблематике показал, что существующие программные комплексы имеют некоторые недостатки [2]. К ним можно отнести высокую стоимость ПО, ориентированность на конкретную ОС, морально устаревшие системные требования к оборудованию пользователя. Данные факты и интерес к исследованиям в ядерной физике побудили авторов статьи приступить к разработке собственного решения под названием **iRB.Space** [3].

Проект нацелен на автоматизацию вычислений и упрощение последующего доступа к результатам исследования методом POP. Уровень современных технологий позволяет использовать в проекте модель **SaaS** – “software as a service”, “программное обеспечение как услуга”. Яркими примерами этой модели являются сервисы электронной почты (Gmail, Яндекс.Почта) и работы с файлами (Sync.com, Box). Модель SaaS предполагает предоставление пользователю доступа к сервисам посредством разработанных клиентских приложений (преимущественно веб-версий) вместо передачи пользователю изолированного программного комплекса целиком, как в случае с классическим ПО. Единственный минус — потребность в активном либо периодическом интернет-соединении для обновления данных. С другой стороны такая модель распространения ПО дает ряд преимуществ.

Во-первых, **кроссплатформенность**. Веб-версия клиента требует лишь наличие браузера на устройстве пользователя — любой ОС и любой платформы, в то время как существующее ПО по тематике совместимо лишь с некоторыми 32-битными версиями операционной системы Windows. Поскольку модель SaaS предполагает работу типа клиент-сервер, когда основная логика работы приложения находится на сервере либо динамически предоставляется по запросу на клиент, клиентская часть может быть упрощена и быть нетребовательной к системным ресурсам. В то же время клиентская версия с легкостью доступа для расширена до уровня «классического» приложения применением технологий типа Electron и Cordova [4]. Для достижения подобного уровня гибкости в качестве архитектурного решения выбрана платформа node.js — язык программирования JavaScript в полной мере совмещает требования к быстрой и кроссплатформенной разработке вместе с производительностью и обеспечением качества кода.

Второе преимущество модели SaaS – **постоянный доступ** к данным. Хранение всей информации на едином сервере позволяет получить доступ к данным с любого устройства, поддерживающего клиентскую часть. Благодаря универсальной веб-версии, исследователи могут взаимодействовать с сервисом не только на компьютерах, но и посредством смартфонов и планшетов.

С архитектурной точки зрения подобный сервис должен иметь как структурированные данные (сведения о химических элементах, физические характеристики и зависимости, эталонные значения), так и неструктурированные (результаты исследований, в которых возможны различные вариации наборов входных значений и итоговых показателей). Следовательно для обеспечения хранения рационально использовать как реляционные (RDBMS, MariaDB), так и не реляционные (NoSQL, MongoDB) базы данных. Такой выбор технологий успешно зарекомендовал себя на реальных проектах [5].

Ещё одно преимущество – **возможность совместной работы**. Группа исследователей, работающая над одним образцом, сможет получить равный доступ к информации о результатах POP. Единый центр работы пользователей способствует обмену информацией между ними и пополнению базы данных выполненных исследований. Со стороны сервиса передача данных в реальном времени возможна с применением технологии WebSocket [6].

Наконец, **простота развёртывания и поддержки** приложения. При реализации проекта как веб-интерфейса, от пользователя не потребуются никаких дополнительных действий по скачиванию и установке приложений. Кроме того, такая система гарантирует одинаково стабильную работу для всех пользователей на устройствах с браузерами, отвечающих современным веб-стандартам.

Отметим важность стабильной работы и минимизации ошибок. Развёртывание и контроль за состоянием приложения выполняется сервисами gulp и nodemon [7], корректность кода обеспечивается модулями jshint и mocha. Поскольку актуальные алгоритмы для вычислений хранятся на сервере и едины для всех исследований, выявление и исправление ошибок происходит незаметно для пользователей. В случае с классическими приложениями разработчикам пришлось бы рассылать обновленные версии приложения и уведомлять клиентов об изменениях.

Перечисленные выше факторы стали определяющими в выборе архитектурного решения. Модель SaaS полностью соответствует потребностям современного мира, обеспечивая в равной мере доступность и универсальность, эффективность, производительность и удобство использования как для конечного пользователя, так и для разработчика.

Список использованных источников:

1. Ташлыкова-Бушкевич, И.И. Метод резерфордовского обратного рассеяния при анализе состава твердых тел: учебно-метод. пособие / И.И. Ташлыкова-Бушкевич. – Минск: БГУИР, 2003. – 52 с.
2. Солодкий, Д.М. Композиционный анализ состава твёрдых тел методом резерфордовского обратного рассеяния: программные комплексы. / Д.М. Солодкий, И.И. Ташлыкова-Бушкевич // Актуальные направления научных исследований XXI века: сб. научных трудов. – Воронеж: ВГЛУ, 2015. – №8 ч.1 – С. 275-278.
3. iRB.Space – Intelligence in IBA processing [Электронный ресурс]. – Режим доступа: <http://irb.space/>.

4. Солодкий, Д.М. Эффективное покрытие мобильных платформ с использованием технологии Apache Cordova / Д.М. Солодкий, В.Н. Козуб. – Материалы 51-я научной конференции. – Минск: БГУИР, 2015. – С. 35.
5. Солодкий, Д.М. Программный модуль агрегации расписаний с использованием нереляционной базы данных: дипломная работа / Д.М. Солодкий. – Минск: БГУИР, 2015. – 100 л.
6. About HTML5 WebSocket [Электронный ресурс]. – Режим доступа: <http://www.websocket.org/aboutwebsocket.html>
7. Nodemon [Электронный ресурс]. – Режим доступа: <http://nodemon.io/>.

ДЕКЛАРАТИВНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PROLOG

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

Чистяков Михаил

Луцик Ю. А. – к-т технич. наук, доцент

Декларативное программирование сильно отличается от того подхода, к которому привыкли многие программисты. Практически все популярные на сегодняшний день языки программирования реализуют **императивный** подход. Суть этого подхода заключается в том, что программист должен разработать алгоритм решения задачи и правильно оформить его в рамках языка. Однако само построение алгоритма уже является достаточно непростой творческой задачей. Парадигма **декларативного** программирования даёт возможность абстрагироваться от построения алгоритма и сосредоточиться на правильном формальном описании задачи. Рассмотрим основные моменты данного подхода на примере языка **Prolog**.

Программа на языке Prolog состоит из фактов, правил и вопросов. Правило представляет собой конструкцию вида “заголовок :- тело” и буквально означает “заголовок истина, если тело истина”. Факт представляет собой предикат с конкретным значением и имеет вид “имя(значение)”. Совокупность фактов и правил программы составляют её **базу знаний**. Она имеет ключевое значение в процессе решения задачи. Программа, представляющая собой базу знаний, может иметь много точек входа. Решение инициируется при получении некоторого запроса, по которому производится поиск в базе знаний. В основе поиска решения программы лежит метод **поиска с возвратом** и **унификация**. Если есть несколько вариантов дальнейшего выполнения (определено несколько правил с одинаковой сигнатурой), то будут по очереди проработаны все варианты. Если при выполнении программы будет получено противоречие, либо требуемый факт вернёт значение ‘ложь’, произойдёт возврат назад в то место, в котором возможно было другое поведение. Рассмотрим применение Prolog для решения следующей задачи.

В некотором поле шахматной доски 8x8 находится шахматный конь. Найти последовательность ходов, которая может привести его в другое указанное поле.

Ниже приведён один из возможных вариантов реализации данной программы на языке SWI-Prolog. Для того, чтобы получить желаемый ответ, необходимо сформировать запрос вида “findSolutions([x1,y1], [x2,y2], S, X).” Ответ будет представлен в виде X=[[x',y'],[x'',y''],...end], где [x',y'] – координаты следующего хода.

1. findSolutions(StartPos, Goal, MinSteps,X):-
2. solve(StartPos, Goal, MinSteps,X);
3. NewSteps is MinSteps +1,
4. findSolutions(StartPos, Goal, NewSteps,X).
5. solve(X,X,0,[end]).
6. solve(StartPos,Goal,StepsLost,[NewPos|Tail]):-
7. go(StartPos,NewPos,8),
8. NewSteps is StepsLost-1,
9. NewSteps >=0,
10. solve(NewPos,Goal,NewSteps ,Tail).
11. go([X,Y],[X1,Y1],M):- X1 is X+1, Y1 is Y+2, X1<M, Y1<M.
12. go([X,Y],[X1,Y1],M):- X1 is X+2, Y1 is Y+1, X1<M, Y1<M.
13. go([X,Y],[X1,Y1],M):- X1 is X+2, Y1 is Y-1, X1<M, Y1>=0.
14. go([X,Y],[X1,Y1],M):- X1 is X+1, Y1 is Y-2, X1<M, Y1>=0.
15. go([X,Y],[X1,Y1],M):- X1 is X-1, Y1 is Y-2, X1>=0, Y1>=0.
16. go([X,Y],[X1,Y1],M):- X1 is X-2, Y1 is Y-1, X1>=0, Y1>=0.
17. go([X,Y],[X1,Y1],M):- X1 is X-2, Y1 is Y+1, X1>=0, Y1<M.
18. go([X,Y],[X1,Y1],M):- X1 is X-1, Y1 is Y+2, X1>=0, Y1<M.

Данный декларативный подход отлично работает для задач, алгоритм решения которых не определён. Prolog успешно справляется с задачами по прокладыванию маршрута или поиску необходимой расстановки. Представление программы в виде базы знаний и набора запросов делает Prolog мощным инструментом для работы с различными базами данных, а также для работы с искусственным интеллектом. Таким образом, знакомство с Prolog-ом будет полезно для программистов любой квалификации для расширения арсенала методов решения задач и взглядов на программирование в целом.

Список использованных источников:

1. Сошников Д.В. Парадигма логического программирования / В.Д. Сошников // М.: «Вузовская книга», 2006. – 220 с.
2. Братко И. Программирование на языке Пролог для искусственного интеллекта / И. Братко // М.: «Мир», 1990 – 560 с.

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ИМИТАЦИИ ПРОЦЕССА ФУНКЦИОНИРОВАНИЯ РЛС ЗРК В УСЛОВИЯХ ВОЗДЕЙСТВИЯ АКТИВНЫХ ПОМЕХ

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

Лапука А. О.

Лукашевич М. М. – к. т. н.

Одной из бурно развивающихся в настоящее время областей реализации научно-технического потенциала в оборонном секторе отечественной экономики является развитие средств индивидуальной радиоэлектронной защиты летательных аппаратов от управляемого ракетного оружия (УРО) зенитных ракетных комплексов (ЗРК) и авиационных ракетных комплексов перехвата (АРКП). Проведение натурных испытаний таких систем сопряжено с достаточно серьезными организационными трудностями и существенными финансовыми затратами: требуется подготовка и выполнение исследовательских (испытательных) полетов, организация системы управления экспериментом, сбора объективной информации и т.д. Кроме того, в результате летных испытаний обычно не удается получить требуемый объем информации, характеризующий эффективность функционирования разрабатываемой аппаратуры в широком диапазоне условий ее боевого применения.

Для преодоления перечисленных проблем разработчиками используется подход, основанный на методе полунатурных испытаний. Он характеризуется разработкой программно-аппаратного комплекса сигнальной имитации внешних (по отношению к разрабатываемому оборудованию) систем, обеспечивающих воспроизведение условий его боевого применения. Основными требованиями, предъявляемыми к такому комплексу, являются:

- разработка и программная реализация сценариев имитации целевой и помеховой обстановки, учитывающих взаимное расположение и перемещение подавляемой РЛС и помехопостановщика, динамику изменения ракурса наблюдения,

- программно-управляемая имитация зондирующего сигнала подавляемой РЛС с наделением его необходимым видом модуляции, временными и спектральными характеристиками. При этом принципиально необходимыми условиями являются наличие излучения сформированного сигнала на литерной (несущей) частоте, с учетом временной задержки и масштабирования по мощности в соответствии с текущей дальностью расположения РЛС относительно помехопостановщика, а также с учетом угла поворота приемно-передающей антенны (Рисунок 1). Это необходимо для обеспечения корректности имитации сигналов, поступающих на вход приемника разрабатываемой аппаратуры;

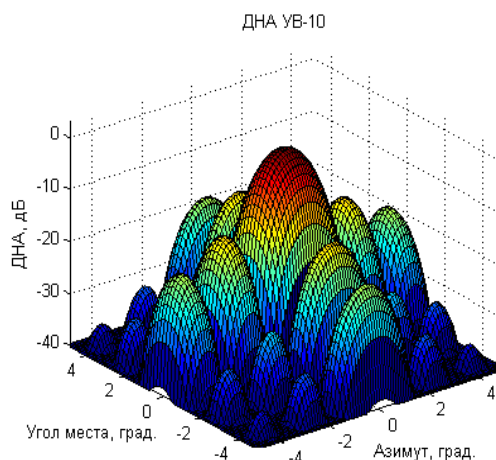


Рисунок 1 – диаграмма направленности приемно-передающей антенны УВ-10

- наличие имитатора полезного сигнала, обеспечивающего соответствующий заданной целевой обстановке уровень мощности и спектрально-временные характеристики отраженных сигналов от цели;
- наличие устройства приема и частотного преобразования полезных и помеховых сигналов,

имитирующих аналогичные процессы функционирования подавляемой РЛС в реальном масштабе времени. Конечным элементом такого устройства является аналого-цифровой преобразователь, выходной цифровой сигнал которого в дальнейшем используется для имитационного моделирования процессов обработки;

– реализованная на сигнальном уровне имитационная математическая модель процессов обработки сигналов в подавляемой РЛС. Данная модель реализует идеализированные процедуры фильтрации, автоматического обнаружения, взятия на сопровождение и измерения координат обнаруженных целей, а также визуализации результатов радиолокационного наблюдения целей и помех на соответствующих индикаторах. При этом принципиальной является реализация алгоритмов помехозащиты, используемых данной РЛС, что обеспечивает гарантированную корректность формируемых оценок эффективности воздействия на нее создаваемых помех.

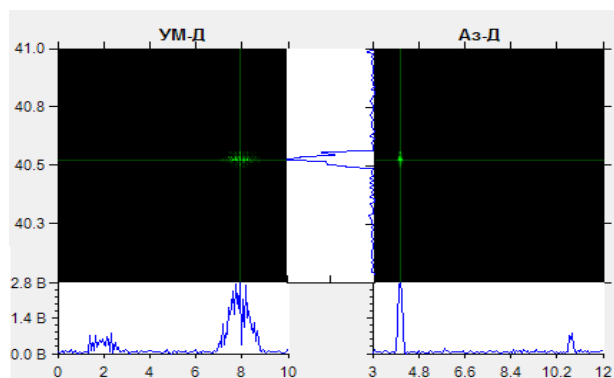


Рисунок 2 – развертки угол места - дальность и азимут - дальность СНР С-125

Описанный подход, основанный на рациональном сочетании аппаратных и программных средств, позволяет с высокой достоверностью оценить результаты воздействия помех на подавляемую РЛС при обеспечении наглядного отображения принимаемых сигналов на индикаторах РЛС (Рисунок 2) и высокой объективности. Его несомненным достоинством является минимизация требуемых материальных и организационных ресурсов в совокупности с возможностью получения требуемых оценок эффективности в широком диапазоне условий боевого применения разрабатываемой аппаратуры.

Список использованных источников:

1. Основы построения зенитного ракетного комплекса С-125М и СНР-125М: учебник / МО СССР – М: 1977. – 296 с.
2. Лапука О.Г. Анализ и синтез в классе дискретных конечномерных систем: монография / Лапука О.Г., Пащенко К.К. – Минск: ВАРБ 2010. – 372 с.

**СЕКЦИЯ
«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ

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

Кирвель Е.А.

Иванюк А.А. – д-р. техн. наук, профессор

В докладе рассмотрены проблемы и особенности реализации физически неклонируемых функций, а также описаны методы оценки качества физически неклонируемых функций. Рассматриваются технические проблемы, связанные с реализацией статических тестов NIST, и методы их решения.

В современном мире информационных технологиях все большее внимание стоит уделять защите информации. Криптографические методы применяются практически во всех информационных технологиях и охватывают полный стек промышленной разработки: начиная от цифровых устройств и заканчивая протоколами передачи данных в веб-технологиях. Физическая криптография является одним из наиболее современных достижений в области криптографии и защиты информации. Доминирующей категорией физической криптографии являются физически неклонируемые функции (ФНФ). Существуют варианты реализации физически неклонируемых функций. В данном докладе рассмотрен вариант реализации физически неклонируемой функции типа арбитр, для анализа которой и было применено описываемое программное средство. Для определения качества ФНФ обычно используют следующие параметры: уникальность, стабильность, случайность. Для расчета каждой из них имеются свои подходы и особенности. К примеру для расчета случайности часто прибегают к статическим тестам Д. Кнута, тестам Дж Марсальи, Национального института стандартов США (NIST). Описываемое программное средство реализует статические тесты NIST для расчета случайности. На данный момент имеются реализации тестов NIST на языке C, однако эти пакеты имеют ряд недостатков: имеют ошибки в реализации алгоритмов, алгоритмы не оптимизированы, нечитабельный код. При использовании потоков и элементарных оптимизаций, можно значительно улучшить производительность данных тестов. Для расчета уникальности и стабильности описываемое программное средство предоставляет возможность выбора метода расчета разницы между двумя векторами ответов от одной и той же ФНФ. В настоящее время не существует такого программного средства, которое агрегировало бы в себе тесты для ФНФ, позволяло бы в удобной форме получать результаты и проводить анализ.

Основными целями создания программного средства являются:

- 1) Обеспечить возможность расчёта основных характеристик ФНФ таких как уникальность, стабильность, случайность.
- 2) Реализовать статические тесты NIST используя .NET Framework.
- 3) Добиться улучшения производительности тестов NIST за счет распараллеливания операций.
- 4) Обеспечить анализ результатов исследования ФНФ типа арбитр на программируемых логических интегральных схемах (ПЛИС) типа FPGA.
- 5) Реализовать возможность получения результатов в виде графиков для удобства анализа результатов.

В качестве языка программирования для реализации программного средства был использован язык C#. Развитая инфраструктура платформы .NET и появившиеся в .NET 4.5 удобные конструкции асинхронного и параллельного программирования позволили создать надежное и высокопроизводительное программное средство.

В ходе реализации все поставленные цели были достигнуты. Были использованы различные средства параллельного программирования .Net Framework. При помощи PLINQ удалось добиться увеличения производительности тестов более чем в 2 раза. Программное средство было протестировано при подаче 100 блоков, состоящих из 100 000 бит каждый.

Таким образом, разработанное программное средство автоматизации тестирования физически неклонируемых функций позволяет осуществлять расчет основных характеристик ФНФ, а также упрощает анализ полученных результатов. Предлагаемое программное средство было использовано на практике для исследования ФНФ типа арбитр на программируемых логических интегральных схемах (ПЛИС) типа FPGA. При помощи данного программного средства исследовались характеристики как для нескольких экземпляров ФНФ на одном кристалле, так и на разных кристаллах.

Список использованных источников:

1. Ярмолик В.Н., Вашинко Ю.Г. Физически неклонируемые функции / Информатика, №2, 2011 г. 12с..
2. Клыбик В.П. Применение физически неклонируемой функции типа арбитр для решения задачи идентификации цифровых устройств / В.П Клыбик, А.А. Иванюк // Информатика. – 2015., - №3 – С 24-34
3. Zalivaka S.S, Puchkov A.V., Klubik V.P., Ivaniuk A.A., Chang C.H. Multi-valued arbiters for Quality Enhancement of PUF Responses on FPGA Implementation.
4. onses on FPGA Implementation.

ИССЛЕДОВАНИЕ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ С ИСПОЛЬЗОВАНИЕМ СОФТ-ПРОЦЕССОРА MICROBLAZE

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

Пучков А. В.

Иванюк А. А. – доктор технических наук, профессор

Одним из перспективных решений в области защиты цифровых устройств и систем от несанкционированного использования является применение решений, основанных на так называемых физически неклонировуемых функциях. В основе их функционирования лежит структурная сложность физических систем. В силу своей природы физически неклонировуемые функции наиболее полно исследуются в своих физических реализациях. Рассмотренное в работе программно-аппаратное решение использует программируемые логические интегральные схемы типа FPGA с работающим на базе них софт-процессором и высокопроизводительным сетевым интерфейсом для эффективной постановки экспериментов на реализациях физически неклонировуемых функций.

В условиях, когда несанкционированное использование цифровых устройств, систем и их проектных описаний становится все большей проблемой, широкое распространение получают методы физической криптографии, в частности использование физически неклонировуемых функций (ФНФ) [1]. При создании любых цифровых систем принципиально невозможно управлять величинами многих физических параметров, вследствие чего последние вследствие вариации техпроцесса принимают случайные, но уникальные для каждой цифровой системы значения. Идея извлечения подобных уникальных параметров из цифровых систем и является основой аппаратных реализаций ФНФ [1].

Эффективное использование ФНФ не представляется возможным без предварительного исследования их свойств на целевой аппаратной платформе. В настоящее время широко распространены программируемые логические интегральные схемы (ПЛИС) типа FPGA, которые предлагают чрезвычайную гибкость, и используются не только для прототипирования цифровых систем, но и для их конечной реализации [1]. Исследование различных аспектов реализаций физически неклонировуемых функций на FPGA неоднократно становилось предметом исследований [2].

Однако процесс исследования сопряжен с рядом трудностей, в числе которых можно выделить необходимость передавать с платы прототипирования на основе FPGA на рабочую станцию большой объем данных (в простейшем случае, запросов и ответов ФНФ), а также надежность используемого решения, напрямую влияющая на достоверность получаемых данных. В рамках разработанного решения были предприняты шаги к эффективному решению данных проблем, состоящие в использовании высокопроизводительного сетевого интерфейса, а также реализации значительной части логики системы программным способом, при этом выполнение программного кода осуществляет софт-процессор MicroBlaze, разработанный корпорацией Xilinx.

При помощи имеющихся плат быстрого прототипирования Xilinx Artix-7 и программного обеспечения Xilinx Vivado была спроектирована цифровая система, осуществляющая управление реализациями ФНФ и передачу данных от них к рабочей станции или иному устройству. Ядром системы стал софт-процессор MicroBlaze, соединенный с периферийными устройствами посредством шины AXI. К важнейшим из периферийных устройств относятся контроллер Ethernet (канальный уровень), внешняя память ОЗУ и сами реализации ФНФ. Использование Ethernet позволило улучшить производительность и надежность интерфейсного взаимодействия с рабочей станцией по сравнению с [2]. При этом физический уровень модели ISO/OSI присутствовал в виде аппаратного модуля на плате прототипирования, канальный реализован вышеупомянутым контроллером, а все вышележащие – программным способом (с использованием протокола TCP в качестве транспортного), что делает данное решение еще более гибким.

Из-за того, что логика управления ФНФ реализована в виде программного обеспечения для реализованного средствами FPGA софт-процессора MicroBlaze, она легко отлаживается и, что немаловажно, любые ее изменения не приводят к очень затратной по времени процедуре синтеза проектного описания цифровой системы. Вместо этого происходит лишь перекомпиляция исполняемых модулей встраиваемого ПО и их загрузка. Кроме того, сам софт-процессор, будучи конфигурируемой IP-компонентой, может быть легко настроен для наиболее эффективного решения поставленных задач.

Для выполнения на стороне рабочей станции было разработано клиентское программное средство с использованием технологии .NET на языке C#.

Использование новых решений позволило существенно улучшить эффективность исследования в части используемого времени, а также возможностей верификации, гибкости в использовании, что было отмечено уже в первых экспериментах.

Список использованных источников:

1. Иванюк, А. А. Проектирование встраиваемых цифровых устройств и систем: монография / А. А. Иванюк. – Минск: Бестпринт, 2012. – 337 с.
2. Заливако, С. С. Аппаратно-программный комплекс исследования физически неклонировуемых функций / С. С. Заливако, А. А. Иванюк, В. П. Клыбик, А. В. Пучков // Материалы международной научной конференции ITS 2015. – Минск: БГУИР, 2015. – С. 174–175.

АЛГОРИТМЫ И МОДЕЛИ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ ВЕБ-СЕРВИСОВ

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

Разбоев Д. О.

Кабак Е. В. – канд. техн. наук, инженер-программист

Нагрузочное тестирование – автоматизированное тестирование для определения, как быстро и надежно работает вычислительная система или ее часть. Во время тестирования может быть отслежено время отклика на запрос, потребление ресурсов системой.

Рассмотрим пример. Пусть тестовый сценарий предусматривает, что некоторое количество пользователей будут производить логин, переходить по основным страницам, заполнять формы на страницах. В соответствии с приведенным сценарием будет производиться нагрузочное тестирование для определения максимально возможного количества одновременно работающих пользователей. В качестве программного обеспечения будет использоваться открытое программное обеспечение JMeter. Данное программное обеспечение позволяет производить нагрузочное тестирование с использованием технологий JDBC/FTP/LDAP/HTTP/POP/TCP. Так же данное программное обеспечение может легко быть изменено. Имеет большой набор плагинов.

В данном примере после шага с логином, для последующих запросов сохраняются cookie. Запрос является прошедшим при следующих условиях 1) если он выполнен в заранее установленное время; 2) если в теле ответа сервера содержатся ключевые слова; 3) если код ответа сервера равен положительному. В последствии можно усовершенствовать условия, которые считаются успешными ответами на запросы. Для получения входных тестовых данных для тестирования используется технология JDBC.

На рисунках 1 и 2 приведены структурные схемы тестового плана и результат выполнения тестового плана для 510 виртуальных пользователей в течении 10 секунд:

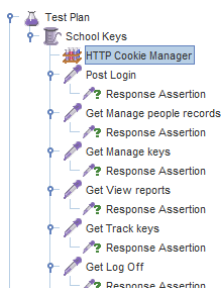


Рис. 1 – Структурная схема тестового плана

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	KB/sec
Post Login	510	4277	3511	9378	10209	10945	1014	11653	0,00%	4,7/sec	26,0
Get Mana...	510	2205	1523	6199	8232	8967	1002	9782	0,00%	4,7/sec	10,1
Get Mana...	510	2099	1531	3687	7761	9061	1002	9950	0,00%	4,7/sec	10,0
Get View r...	510	2039	1531	2573	7472	9300	1002	9949	0,00%	4,7/sec	10,1
Get Track...	510	2964	2522	3605	7872	11120	1002	11344	0,00%	4,7/sec	24,4
Get Log Off	510	2975	2547	3593	8113	10432	1005	11248	0,00%	4,7/sec	8,1
TOTAL	3060	2760	1585	6612	8721	10432	1002	11653	0,00%	26,0/sec	81,7

Рис. 2 – Результат выполнения тестового плана для 510 виртуальных пользователей в течении 10 секунд

В данном случае настройки выполнения тестового плана были выбраны случайным образом, в реальных системах данные берутся из спецификации системы.

Таким образом, были разработаны модели и алгоритмы нагрузочного тестирования веб-сервисов. Рассматриваемая система обеспечивает проверку веб-сервиса на работоспособность в условиях высокой нагрузки. Так же проверяется что время отклика и расходование ресурсов.

Список использованных источников:

1. Савин, Р. Тестирование Дот Ком / Р. Савин. – Минск : Дело, 2007 – 312с.
2. Винченко, И. Автоматизация процессов тестирования / И. Винченко. – Спб. Ж: Питер, 2008. – 211 с.

СИСТЕМА ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ В ПОМЕЩЕНИИ

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

Павлович Ю.Г.

Бранцевич П.Ю. - кандидат технических наук, доцент

В последнее время наблюдается рост интереса к точным методам определения положения, которое используется для написания различных. Глобальная система позиционирования и беспроводные сервисы решают данную задачу, тем не менее, такие технологии не могут обеспечить точную геолокацию внутри помещения.

Двумя основными проблемами для точного определения местоположения в помещении являются сложность распространения радиоволн и различная планировка помещений. Из-за этих трудностей необходимо тщательно подбирать метод передачи сигнала, общую системную архитектуру, а также алгоритм определения положения. В данном докладе представлен обзор метода и модели для беспроводных систем определения местоположения.

Целью данной работы является разработка модели системы местонахождения локации внутри помещения. Эта система должна предоставлять и визуализировать информацию о локации устройства с достаточно большой точностью. Также эта система должна быть доступной: технические средства для построения системы не должны быть дорогими.

Существует два основных подхода к разработке беспроводной системы геолокации устройств. Первый подход заключается в разработке сетевой инфраструктуры датчиков местоположения. Второй подход заключается в использовании существующей беспроводной сетевой инфраструктуры. Преимущество первого подхода состоит в том, что физические характеристики, и, следовательно, качество результатов определения местоположения, может быть улучшено под конкретную ситуацию. Радиопередатчик может быть выполнен в виде небольшого устройства или наклейки, а плотность инфраструктуры таких датчиков может быть отрегулирована до требуемой точности определения положения применения. Преимущество второго подхода заключается в том, что он является более дешевым, так как не требует времени на подготовительную фазу инфраструктуры.

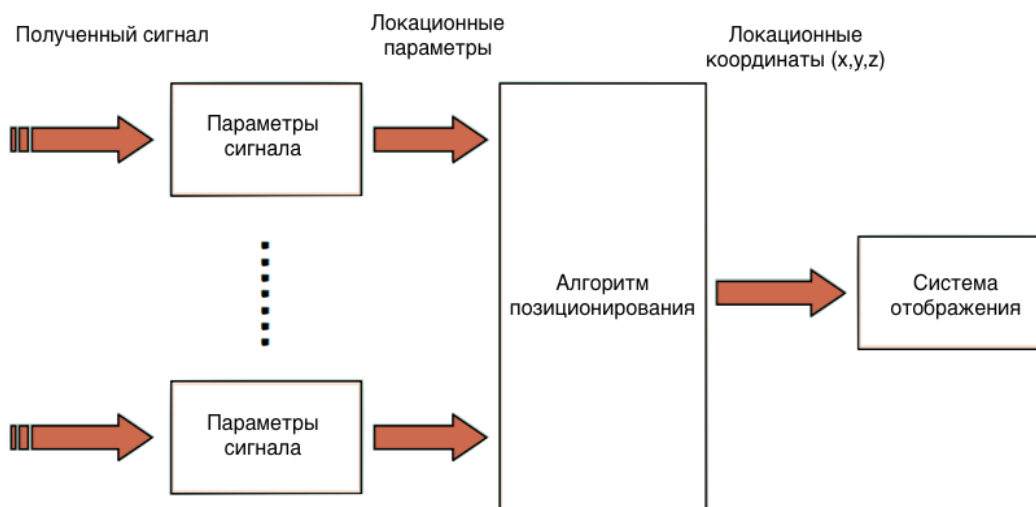


Рис. 1 – Функциональная блок-диаграмма системы беспроводной локации

Основными элементами системы являются датчики для измерения параметров радиосигнала, которые преобразуются в относительные параметры положения устройства в пространстве по отношению к заранее известным опорным точкам, алгоритм определения местоположения, который обрабатывает измерения для расчета координаты местоположения устройства в пространстве, а также устройство отображения, которое визуализирует расположение устройства в пространстве. Относительными параметрами положения устройства могут быть: приблизительное направление полученного сигнала, приблизительное расстояние устройством и датчиками. Угол принятого сигнала является основным показателем, используемым в системах на основе направления. Сила принимаемого сигнала, фаза сигнала, а также скорость получения сигнала являются параметрами для оценки расстояния. Система отображения отображает координаты устройства, или местоположение устройства на карте.

Расположенные датчики для получения параметров страдают от сильных помех, создаваемых другими датчиками и сильной радиотени, поэтому измерения силы сигнала и угла обстрела обеспечивают

менее точные показатели, чем скорость получения ответного сигнала. В результате, системы, подобные GPS, предназначенные для внутренней геолокации должны использовать более точные параметры скорости ответа в качестве основного параметра. Системы, использующие существующие инфраструктуры беспроводных сетей или мобильные сети 3G, в закрытых системах могут использовать параметры силы сигнала и угла обстрела, используя возможности данных технических средств.

Благодаря надежным измерениям расстояний между датчиком и устройством, а также геометрическим методам триангуляции, можно найти местоположение устройства. Для достижения более высокой точности местоположения, ошибки, возникающие в процессе измерений должны быть смягчены в процессе нахождения позиции.

Геолокация внутри зданий характеризуется тем, что зона обслуживания ограничивается внутренней и непосредственной близости от здания. В настоящее время план сооружения, как правило, доступен в виде электронного документа. Наличие электронных чертежей является важной составляющей в алгоритмах нахождения позиции.

Ошибки, возникающие при отслеживании устройства, при переходе в между помещениями или между этажами, могут быть легко распознаны и устранены. Еще одной уникальной чертой локации внутри помещений является то, что размер зоны покрытия значительно меньше, чем на открытом воздухе. Это дает возможность проводить тщательное планирование о размещении датчиков. Такое планирование инфраструктуры датчиков может значительно снизить погрешности измерения местоположения. Данная информация используется в алгоритмах нахождения позиции.

Таким образом, была предложена модель определения местоположения в здании относительно мобильных датчиков внутри помещений. Расчет координат осуществляется с помощью метода триангуляции с использованием позиции датчиков относительно плана помещений. Данный метод позволит разработать систему, которая является недорогим для проектирования и развертывания, соответствует нормам частот, и обеспечивает необходимую точность.

Список использованных источников:

1. K. Pahlavan and P. Krishnamurthy, Principles of Wireless Networks — A Unified Approach, Prentice Hall, 2002.
2. P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," IEEE INFOCOM, Israel, Mar. 2000.

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЙ НА REACT.JS, С ПРИМЕНЕНИЕМ FLUX АРХИТЕКТУРЫ

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

Пунько В. В.

Парамонов А. И. – канд. техн. наук, доцент

Современные требования к пользовательским интерфейсам веб-ориентированных систем накладывают повышенные требования к интерактивности и удобству использования функций приложения. В связи с этим язык программирования JavaScript активно используется для работы на стороне клиента, а вместе с ним и весь пул связанных с языком технологий.

Из-за специфики веб-приложений, разработка для веба более трудоёмка чем написание десктопных приложений. Для того, чтобы нивелировать эту проблему используют различные веб-фреймворки, основанные на архитектурах mvc [1] и flux [2]. На данный момент известно несколько [3] хорошо себя зарекомендовавших себя решений: Angular, Ember, Backbone, React. Полноценными фреймворками являются Angular и Ember, Backbone и React – библиотеки для построения пользовательских интерфейсов. Для больших веб-приложений остро встаёт проблема зависимости частей приложения между собой, что влечет значительное возрастание сложности архитектуры веб-системы и падения производительности ее работы.

Одним из путей решения данного вопроса может быть использование связки React.js и архитектуры Flux. Новая, активно развивающаяся и перспективная разработка компании фейсбук React в последнее время привлекает все большее внимание. В сочетании с применением архитектуры Flux библиотеку React.js можно считать полноценным фреймворком. Основной идеей React.js [4] является компонентно-ориентированный подход, опирающийся на революционную идею сочетать JavaScript код и разметку компонентов. Flux – это архитектура базирующаяся на однонаправленном потоке данных и подходе управлением изменением состояния через события. На рисунке 1 приведены структурная схема архитектуры.

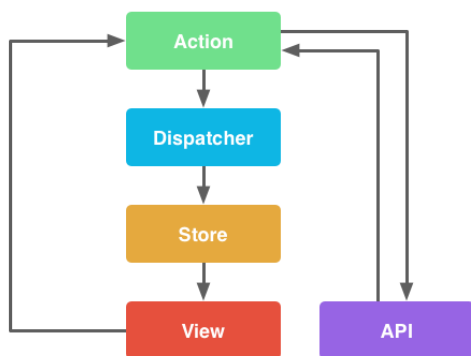


Рис. 1 – Структурная схема Flux архитектуры.

Основными элементами Flux являются:
 View – пользовательский интерфейс состоящий из react-компонентов;
 Dispatcher – система регистрации обратных вызовов и реагирования на события;
 Store – хранилище данных на клиенте, которое взаимодействует с View через Dispatcher;
 Action – прослойка между Dispatcher и View для регистрации обработчиков событий действий пользователя.
 Для простоты далее будем называть Store – хранилище, Dispatcher – диспетчер.
 Типичным подходом в Flux является единственный диспетчер, в котором регистрируются все события приложения и многочисленные хранилища в соответствии с доменными областями приложения.

В контексте средних веб-приложений один диспетчер способствует уменьшению количества зависимостей. Однако для больших приложений появляется проблема не разграниченности доменных областей и возрастания сложности. Единственный диспетчер легко может стать узким местом для производительности приложения, потому что в нём регистрируется все обработчики событий приложения.

Если представить большое веб-приложение как совокупность приложений меньшего объёма, то целесообразным является использовать несколько диспетчеров на каждое такое подприложение. Это способствует разграничению и упрощению кодовой базы и решает возможную проблему производительности. Для связи подприложений между собой можно использовать общий диспетчер. Важно отметить, что данный подход актуален для больших по объёму веб-приложения (преимущественно single page application), в которых можно выделить независимые под части.

Таким образом, был рассмотрен подход к решению проблемы возрастания сложности и падения производительности в веб приложениях, которая связана с особенностью архитектуры платформы, предполагающей единственный диспетчер событий. Подход разделения приложения на зоны ответственности нескольких диспетчеров обеспечивает упрощение кода приложения, что сокращает время на расширение и сопровождение кода, а также предотвращает возможную проблему производительности.

Список использованных источников:

1. «MVC в JavaScript» [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://designformasters.info/posts/mvc-javascript/>
2. «Flux официальный сайт» [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://facebook.github.io/flux/docs/overview.html#content>
3. «Сравнение JavaScript фреймворков» [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://apps4all.ru/post/03-10-15-sravnenie-samyh-populyarnyh-javascript-mv-frejmworkov-ch2>
4. «Документация по React.js» [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://facebook.github.io/react/>

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ КВАЛИФИКАЦИЕЙ ПЕРСОНАЛА

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

Шурова Е. О.

Глухова Л. А. – канд. техн. наук, доцент

Стратегическое управление профессиональным потенциалом компании является залогом эффективного функционирования организации. В докладе рассматриваются принципы автоматизации управления квалификацией персонала, позволяющие облегчить работу как руководителя в выявлении талантов и направлении их на благо компании, так и сотрудника в проявлении навыков и в планировании своего профессионального развития.

В условиях глобализации экономики ресурсы компаний в финансовых и технических сферах становятся доступными и в результате более схожими, что в значительной мере понижает способность к конкуренции. Поэтому преимуществом компании становится ее персонал, его знания, навыки и заинтересованность в работе. В связи с этим повышается актуальность концепции управления талантами. Она заключается не только в поиске квалифицированных сотрудников, но и в их профессиональном развитии, в раскрытии их талантов и правильном использовании их потенциала. Также учитывается, что реализация конкретного навыка зависит не только от квалификации и знаний сотрудника, но и от его навыков и отношения к работе. Таким образом, концепция управления талантами включает в себя раскрытие талантов сотрудников и их последующее использование в развитии компании.

Одним из инструментов управления человеческими ресурсами является модель компетенций. Модель компетенций – это общий перечень компетенций, которыми должен обладать персонал организации, с их подробным описанием. Компетенция – это личная способность сотрудника, определяющая наличие знаний и опыта для результативной работы в области, заданной компетенцией. На основе компетенций строятся планы развития персонала, оценивается степень подготовки специалиста и производится отбор талантов.

Сам процесс управления квалификацией персонала сложен и многогранен, а его ручное проведение практически невозможно. С целью облегчения этой задачи предлагается система автоматизации управления квалификацией персонала. Система состоит из трех модулей: модуль сотрудника, модуль компании и модуль оценки персонала. В основу автоматизации положена идея принятия компетенции за единицу управления и оценки.

Основные процессы системы отображены на рисунке 1.

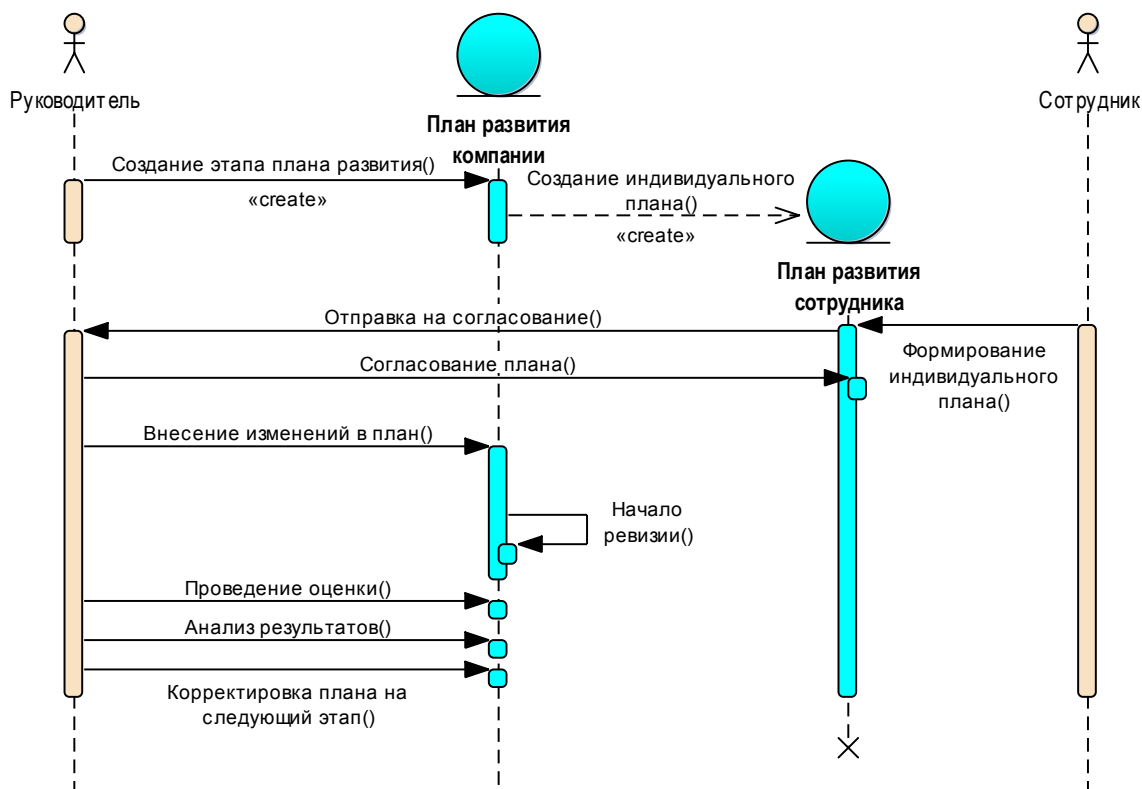


Рис. 1. Связь плана развития компании и индивидуального плана развития сотрудника

С помощью этой системы руководитель создает план развития компании на текущий этап. На основе данного плана сотрудники строят свои индивидуальные планы. По завершении этапа развития проводится оценка персонала и анализ достижений компании на этом этапе. Результаты оценки и анализа используются в качестве базы для формирования плана на следующий этап.

Модуль сотрудника дает персоналу возможность создавать свои собственные планы развития навыков, оценивать свои текущие знания, задавать планируемый уровень развития и дату, к которой сотрудник предполагает повысить владение данной компетенцией, а также мероприятия, с помощью которых это будет осуществлено.

Модуль компании предназначен для руководителей компании.

Во-первых, в нем создается корпоративная модель компетенций, которая отражает стратегические ориентиры компании. Для грамотного построения модели компетенций существует возможность привлечения экспертов для принятия решения о включения компетенции в модель.

Во-вторых, руководитель создает план развития компании, в котором выделяет компетенции и группы компетенций, развитие которых наиболее приоритетно на данном этапе.

В модуле компании также задаются целевые уровни владения, которых должна достичь организация, и планируемое увеличение количества специалистов по компетенциям или их группам.

По завершении этапа развития начинается ревизия плана и периодическое оценочное мероприятие, с помощью которого выявляется успешность осуществления плана развития компании. По результатам ревизии корректируется план на следующий этап развития для достижения максимальной эффективности.

В-третьих, руководитель согласовывает индивидуальные планы сотрудников, утверждая или отправляя на доработку целевые уровни компетенций.

Модуль оценки персонала дает возможность проведения оценки, которая выявит текущие уровни знания сотрудников, их прогресс в обучении и перспективы в дальнейшей работе. Для оценки применяется метод «Оценка 360°», где человека оценивает не только его руководитель, но и его коллеги или эксперты в

определенных сферах. Автоматизация позволяет во многом сократить время, затрачиваемое на оценку специалистов, систематизировать все оценки, упростить выставление итоговой оценки.

Использование предлагаемой системы автоматизации управления квалификацией персонала позволит существенно повысить конкурентоспособность компании в своей сфере деятельности.

Список использованных источников:

1. Спенсер Лайн М, Спенсер Сайн М. Компетенции at work. Модели максимальной эффективности работы/ Спенсер Лайн М, Спенсер Сайн М. – М.: Гиппо, 2005. – 384 с.
2. Карпова А.В., Ключевой Н.В. Технологии управления развитием персонала. Учебник/ Карпова А.В., Ключевой Н.В. – М: Проспект, 2016. – 408 с.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АНАЛИЗА ВИБРАЦИОННЫХ СИГНАЛОВ НА ПЛАТФОРМЕ ANDROID

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

Алексеев Ю.И

Бранцевич П.Ю. – доцент, к-т техн. наук

Применение мобильных технологий и устройств для профессионального применения является современным направлением развития техники. Вибрационная диагностика является перспективной областью, в которой внедрение данных технологий может принести существенные выгоды.

Измерение и анализ вибрационных колебаний промышленного оборудования является важной мерой контроля и оценки технического состояния производственных машин. Повышение интенсивности вибрации является важным признаком неполадок и дефектов оборудования. Техническое состояние машины можно определить, сравнивая текущий уровень вибрации с пороговым значением. Пороговое значение вычисляется на основе предыдущих измерений вибрационного уровня, производственной статистики, спецификации и технических условий оборудования. Важным параметром вращательных машин является скорость вращения вала. От данной скорости зависит выбор типа датчиков и анализаторов.

Система вибрационного анализа обычно состоит из 4 базовых частей:

1. Вибрационные датчики на различном промышленном оборудовании.
2. Анализатор сигнала
3. Программное обеспечение для анализа.
4. Компьютер для анализа данных и их сохранения.

Мобильные технологии все более прочно проникают в нашу жизнь, вычислительные мощности планшетов и смартфонов позволяют успешно заменить в повседневном использовании персональные компьютеры и ноутбуки. Такими же быстрыми темпами они находят применение в профессиональной и промышленной областях. Использование мобильных устройств имеет ряд важных преимуществ: портативность и легкий вес, автономность, возможность быстрой смены функциональности устройства путем установки либо замены программного обеспечения, поддержка беспроводных коммуникаций. В сфере вибрационной диагностики мобильные технологии пока ещё не получили широкого распространения, имеется явная нехватка программных средств для профессиональных нужд.

На данный момент наиболее популярными мобильными платформами являются Android, Apple iOS. Платформа Android была выбрана благодаря более гибким возможностям для программирования, широкому ассортименту совместимых устройств от различных производителей, большому количеству литературы и учебных материалов, открытому программному коду и отсутствию необходимости использования проприетарного оборудования.

Программное средство для анализа вибрационных сигналов на платформе Android имеет достаточную функциональность для эффективного анализа вибрационных сигналов. Реализованы следующие функции: вычисление среднего квадратического значения (СКЗ), пик-фактора, вейвлет-анализ, построение моделей сигнала, определение амплитудного спектра сигналов, выделение гармонических составляющих сигнала, вычисление ряда других параметров. Для более эффективного использования площади экрана программа оптимизирована для работы в ландшафтном режиме отображения.

Программа может загружать файлы из локального хранилища (внутренней памяти либо карты памяти) или получать данные с сервера. Собранные сигналы сохраняются в базу данных для последующего сравнения и анализа тенденций изменения состояния оборудования. В отличие от большинства программных средств, поставляемых в составе программно-аппаратных комплексов, она совместима с различными форматами файлов вибрационных сигналов, что расширяет список поддерживаемой аппаратуры по снятию данных с оборудования и расширяет область применения.

Программа совместима с устройствами, работающими на платформе Android версии 4.0 и выше.

Планируется разработка полноценного программно-аппаратного комплекса, включающего в себя аппаратуру для снятия и беспроводной передачи сигналов с вибрационных датчиков на сервер, а затем анализа и обработки данных с помощью данного программного средства.

Применение данной программы позволит пользователям уменьшить эксплуатационные издержки, повысит гибкость использования и снизит потребности в специализированных аппаратно-программных комплексах.

Список использованных источников:

1. Sheffer C. Machinery Vibration Analysis & Predictive Maintenance / C. Sheffer, P. GirdHar. – Newnes, 2004

VIPER АРХИТЕКТУРА В МОБИЛЬНОЙ РАЗРАБОТКЕ

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

Антоненко Д.А.

Куликов С.С. – к-т. наук, доцент.

В последнее время наблюдается тенденция появления новых архитектур построения мобильных приложений. Это вызвано тем, что не все имеющиеся архитектуры позволяют достичь необходимой гибкости построения приложений, необходимого уровня тестируемости, эффективного распределения функциональности между компонентами. В результате часто приходится создавать свои собственные архитектурные решения, что занимает достаточно много времени. В связи с этим начался поиск некоторого универсального решения, решающего большинство проблем.

Исторически, базовой архитектурой при построении мобильных приложений являлась архитектура Модель-Вид-Контроллер (англ. - MVC). Со временем появилось много её вариаций, среди которых наиболее известными являются: Модель-Вид-Презентер (англ. MVP), Модель-Вид-Вид-Модель (англ. MVVM). Они устраняют определённые недостатки архитектуры Модель-Вид-Контроллер, но не решают всех её проблем и недостатков.

VIPER – новая архитектура построения приложений (в частности – мобильных), базирующаяся на принципах «чистой архитектуры» (англ. clean architecture). Она призвана решить большинство (если не все) проблем MV(X)-подобных архитектур и заменить их полностью. Ниже приведена схема различных компонентов VIPER-архитектуры и связей между ними.

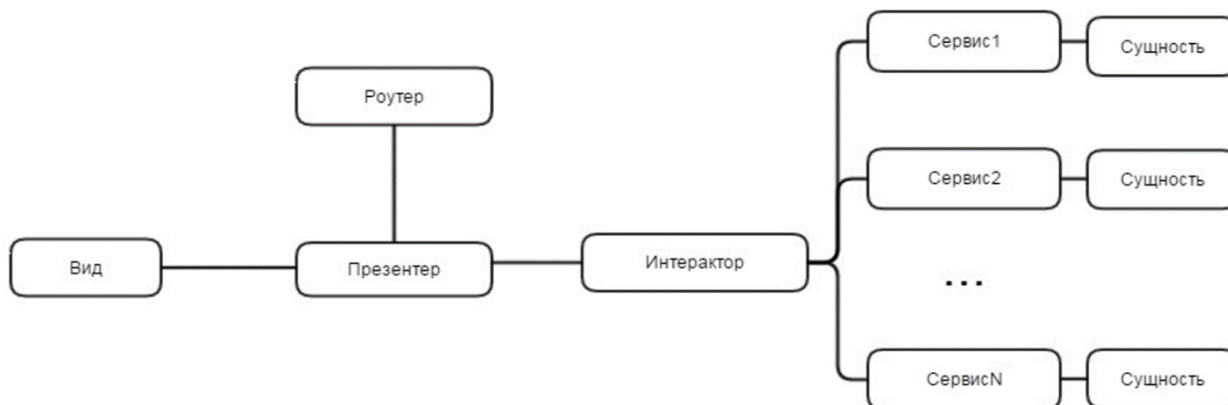


Рисунок 1 – VIPER архитектура, схема различных компонентов и связей между ними.

Основными понятиями, используемыми в данной архитектуре, являются:

- Вид – отображает, что сообщил Презентер и передаёт ввод данных пользователем назад Презентеру.
- Интерактор – содержит бизнес-логику, связанную с данными (сущность) или сетевыми операциями, как например создание новых экземпляров сущностей или загрузка их с сервера.
- Презентер – отвечает за пользовательский интерфейс, связанный с (но независимый от графических библиотек конкретной платформы) бизнес-логикой, вызывает методы на Интеракторе.
- Сущности – простые объекты данных, но не уровень доступа к данным, так как ответственность за это лежит на Интеракторе.
- Роутер – отвечает за непосредственно переходит между VIPER модулями.
- Сервис – единица Интерактора, выполняющая определённый вид работ.

Исходя из рассмотренной архитектуры, можно сделать следующий вывод:

а) Распределение обязанностей – несомненно, VIPER является чемпионом в распределении обязанностей между своими компонентами.

б) Тестируемость – обеспечивается отличная тестируемость благодаря имплементации принципа единственной обязанности.

в) Удобство использования – высокая стоимость «ремонтпригодности». Необходимо написать большое количество интерфейсов для классов с очень малыми обязанностями.

Таким образом, VIPER помогает нам быть более точными, что касается разделения проблем, разделяя большое количество кода одного класса на несколько меньших классов. За счёт поддержания единственной ответственности в каждом классе это упростит разработку классов, используя разработку через тестирование (англ. TDD), которое позволяет нам более быстро реагировать на изменяющиеся требования и создавать лучшее программное обеспечение. В тоже время использовать данную архитектуру неудобно при отсутствии большого количества бизнес-логики на экран. Возникают проблемы при необходимости использования повторно одних и тех же экранов при разных сценариях, так как необходимо хранить зависимости между модулями, который сейчас активен и который будет показан следующим. Сложно «наследоваться» от других модулей приложения.

БЫСТРОЕ ОПРЕДЕЛЕНИЕ ПРИНАДЛЕЖНОСТИ ПОЛЬЗОВАТЕЛЯ К СЕГМЕНТУ ПО IP-АДРЕСУ

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

Базаревский В. Э.

Бранцевич П. Ю. – кандидат технических наук

Рассматриваются алгоритмы быстрого определения принадлежности пользователя к сегменту IP-адресов, особенности хранения и организации сегментов для ускорения работы. Рассмотрена структура данных дерево отрезков, особенности работы с ним и построение.

В современных интернет-технологиях одним из основных способов монетизации приложений является продажа интернет-рекламы, тем самым делая её одним из важнейших элементов экосистемы сети интернет. Вместе с тем следует отметить следующий тренд вместо показа рекламы на определенных сайтах, становится возможным эффективно показывать рекламу её целевой аудитории за счет четкого таргетирования по интересам. Одним из наиболее распространенных методов таргетирования рекламы, является сегментации пользователей на основе их интересов, географического местоположения, используемых устройств, гендерных и иных признаков.

Среди различных средств сегментирования пользователей следует выделить метод сегментирования пользователей по IP-адресам. Так как фактически имеет место географическая сегментация пользователей, то на основе дополнительных сведений об отличительных особенностях абонентов конкретных IP-адресов результирующие сегменты несут в себе более сложную информацию (например, интересы, возраст, пол, профессию и т.д. пользователей). В качестве примера, всех абонентов, выходящих с IP-адресов, принадлежащих университету, можно отнести к сотрудникам университета или студентам.

Зачастую, сегментация по IP является не тривиальной задачей, так как пространство IP-адресов сильно разрежено, представляет собой большое количество интервалов разной длины, при этом сильно пересекающихся друг с другом.

Ввиду описанных выше требований, необходима реализация эффективной структуры данных для определения сегмента по отдельно взятому IP-адресу, оптимальной с точки зрения поиска, масштабируемости и потребления ресурсов. Для решения этой задачи был предложен ряд решений, которые помогли оптимизировать процесс определения принадлежности пользователя к тому или иному сегменту.

Сегмент представляется в виде областей IP-адресов с начальным и конечным IP-адресами. IP-адрес в 4-ой версии представляет собой 32-битовое число. Удобной формой записи IP-адреса (IPv4) является запись в виде четырёх десятичных чисел значением от 0 до 255, разделённых точками. Для того чтобы оптимизировать операцию сравнения адресов и ускорить проверку на вхождение адреса в сегмент, IP-адрес из строкового представления был конвертирован к типу Long. В таком численном виде операция сравнения занимает меньше процессорного времени.

Кроме задачи сравнения адресов существует задача организации сегментов в структуру, оптимизирующую операцию поиска и определения принадлежности IP-адреса к конкретному сегменту. Самый простой способ решать представленную задачу, это завести линейный массив. При такой реализации время нахождения ответа на запрос имеет порядок $O(n)$. Изменение же значения какого-либо сегмента требует $O(1)$ времени. Операция изменения значения для выбора архитектуры представления сегментов в данном случае является не определяющей, так как изменение сегментов происходит редко, а операция поиска в свою очередь выполняется при каждом обращении клиента к сервису. Альтернативной структурой организации данных о сегментах пользователей для оперативного поиска является дерево отрезков.

Дерево отрезков представляет собой дерево, листьями которого являются элементы исходного массива. Остальные вершины являются результатом какой-либо операции над детьми. В нашем случае удобнее всего выбрать операцию объединения сегментов. Таким образом, корень хранит результат

объединения всех сегментов, его левый ребенок - результат объединения элементов с индексом от 0 до $(n / 2 - 1)$, а правый - результат объединения элементов с индексом от $(n / 2)$ до n и так далее до листьев. Существует две реализации построения дерева отрезков, но обе они, как от корня к листьям, так и от листьев к корню, реализуются на базе массива, размерностью ближайшей следующей степени 2. При построении снизу алгоритм поднимается от листьев к корню. Для этого просто начинаем заполнять элементы массива от большего индекса к меньшему, таким образом при заполнении элемента i его дети $2 * i + 1$ и $2 * i + 2$ уже будут заполнены, а при построении сверху спускается от корня к листьям.

Алгоритм построения дерева сегмента является коротким и действенным способом решения подобных задач. Временная сложность $O(\log N)$, что значительно лучше, чем $O(N)$. Например, при массиве длиной 10^9 элементов, необходимо примерно 32 сравнения. Изменить отрезок в этой структуре так же просто — надо пройти по всем вершинам от заданной до 1-ой, и заменить его. Это также занимает $O(\log N)$ времени.

Список использованных источников:

1. Н. Вирт Алгоритмы и структуры данных. СПб.: Невский диалект, 2001. – 352 с.
2. Т. Кормен и др. Алгоритмы: построение и анализ, М.: МЦНМО, 2002.

МОДЕЛЬ СИСТЕМЫ РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ ДАННЫХ

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

Бернацкий В.В.

Бранцевич П.Ю. - кандидат технических наук, доцент

Надежность, доступность, согласованность данных являются основными требованиями предъявляемыми к базам данных. Рост объема и трафика данных, а также использование распределенных вычислений требуют возможности горизонтальной масштабируемости баз данных. В докладе предложена модель системы распределенного хранения данных удовлетворяющая данным требованиям.

Целью данной работы является разработка модели системы распределенного хранения данных с сохранением основных требований к базам данных. Эта система должна быть отказоустойчивой, выход из троля отдельных узлов системы не должны приводить к остановке работы всей системы. Также эта система должна быть доступной: операции чтения и записи должны быть быстрыми. Операции записи не должны приводить систему в несогласованное состояние. Вся нагрузка в системе должна быть равномерно распределенной между всеми узлами (вершинами).

Исходя из требований к данной системе были выделены следующие задачи: определение структуры хранимых данных, обеспечение равномерного распределения нагрузки по всем узлам, обеспечение высокой доступности и согласованности данных.

Разрабатываемая система будет представлять собой коллекцию элементов ключ, имя признака, значение признака, где ключ является уникальным идентификатором логической сущности. Имя признака является именем атрибута этой логической сущности, а значение признака является значением атрибута заданного в поле имя признака. Данный подход позволяет обеспечить равномерное распределение данных основываясь на значении ключ, а добавление понятия признак позволяет уменьшить размер передаваемых данных и уменьшить количество неудавшихся транзакций (за счет применения изменений не ко всей сущности, а лишь к ее части).

Ключевым моментом для обеспечения высокой доступности и надежности является реплицируемость данных - хранение копий одних и тех же данных на нескольких вершинах. Это позволяет распределить запросы для работы с этими данными по вершинам, а в случае выхода из троля отдельных вершин продолжить работу с этими данными используя оставшиеся копии. В данной системе репликации будут подвержены сущности.

В распределенных системах хранения добиться полной согласованности данных практически невозможно, поэтому здесь применяют концепцию согласованности в конечном счете. Данная концепция является менее строгой формой согласованности. Согласно ней данные будут согласованы если никаких новых изменений не происходило. Для обеспечения выполнения данного условия можно использовать подход основной копии. Согласно нему есть 2 категории копий данных (сущности): основная и неосновная. Операции по изменению данных происходят лишь над основной копией данных, а все изменения затем транслируются на оставшиеся копии. Таким образом мы получаем надежность изменения данных как в синхронной модели, но скорость чтения увеличивается пропорционально количеству неосновных копий.

Равномерное распределение значений по узлам осуществляется за счет подсчета хеша ключа сущностей. В зависимости от того, в какой диапазон попало значение, определяется вершина для хранения сущности. Аналогичным образом определяются вершины для хранения других копий значений сущности.

Для обеспечения отказоустойчивости необходимо минимизировать вред от выхода из строя (недоступность) узлов системы. Это можно достичь путем равномерного распределения основных и неосновных копий сущностей и ведением списка приоритета узлов для каждой сущности. Данный список

позволяет для каждой сущности в случае отключения или выхода из строя узла с основной копией выбрать другой узел с копией, который будет использоваться в качестве основного для данной сущности. В совокупности с равномерным распределением основных и неосновных копий по узлам, в случае отключения одной из вершин, произойдет автоматическое перераспределение ролей копий и работа всей системы продолжится. Отдельным случаем стоит рассмотреть вариант при разделении сети на 2 части, где связь в каждой половине будет работать, но соединение между вершинами двух половин будет утеряно. Для того, чтобы у нас не было параллельного функционирования двух основных копий для некоторых сущностей, можно добавить условие, что копия может являться основной лишь при наличии связи с большей половиной узлов, содержащих копию данной сущности.

Таким образом, была предложена модель системы для обеспечения распределенного хранения данных. Выбранная структура данных позволила обеспечить возможность распределенного хранения данных, а также увеличить надежность системы. В качестве базы для обеспечения доступности и надежности использован принцип реплицируемости данных. А использование метода основной копии, хеш-функции для определения вершин для хранения сущностей и алгоритма поведения системы в случае потери связи между вершинами позволили добиться согласованности, доступности, распределения нагрузки и надежности.

Список использованных источников:

1. Mustaque Ahmad , Mostafa H. Ammar , Shun Yan Cheung - "Replicated Data Management in Distributed Systems" - Emory University, 1992
2. Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, Werner Vogels - "Dynamo: Amazon's Highly Available Key-value Store" - Amazon.com, 2007

МЕТРИКИ ОЦЕНКИ ФУНКЦИОНИРОВАНИЯ ПРИЛОЖЕНИЙ ПОД УПРАВЛЕНИЕМ JAVA ЕЕ СЕРВЕРОВ

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

Бродницкий В.В.

Куликов С. С. – к. т. н., доцент

Мониторинг состояния и производительности приложений является критически важной составляющей при поддержке работы приложений, и чем больше возрастает сложность и масштаб системы, тем большую важность приобретает анализ её работы для сохранения стабильности функционирования и целостности данных. Большинство Java-приложений являются не одиночными программными средствами, а состоят из множества различных компонентов, обладают масштабируемой структурой, что значительно увеличивает зависимость состояния системы от ряда внешних факторов.

Количество показателей и метрик работы приложения, доступных для мониторинга, зависит от типа анализируемого программного средства, типа его компонентов и использованных сервисов, а также от конкретной реализации виртуальной машины JAVA. Важной задачей является группировка их по критерию описываемой области, что позволит увеличить эффективность их выборки и анализа. Можно выделить несколько ключевых областей, характеризующих стабильность и производительность любой программной системы:

Показатели виртуальной машины. В данную группу выделяют частоту событий виртуальной машины (количество вызовов и откликов методов, ошибок), степень параллелизма потоков, информация о блокировках, данные о компиляции.

Показатели загруженности памяти. Основными критериями данной группы являются показатели отношения свободной и занятой памяти к общему её количеству. Однако важными показателями являются также сообщения менеджеров памяти о превышении пороговых значений загруженности памяти для определённых пулов и необходимости её увеличения.

Показатели сборщика мусора. Для платформ с автоматическим управлением памятью сборка мусора является ключевым моментом, от эффективности работы которого зависит общая производительность системы. Здесь следует выделить показатели количества произведённых сборок, освобождённой в них памяти, подробные данные о состоянии поколений.

Показатели операционной системы. Так как приложения, выполняемые виртуальной машиной, располагаются на конкретном аппаратном и программном обеспечении, то характеристики их эффективности и готовности непосредственно влияют на показатели производительности самой программной системы. Очевидно, что мониторинг их состояния является важной частью диагностики любого программного средства. В основном здесь выделяют уровень загрузки процессора, доступность и скорость передачи по сети, операций ввода-вывода информации.

Виртуальная машина Java представляет собой сложную многоуровневую систему, включающую в себя множество компонентов и крайне тяжело выделить единую группу показателей, полностью характеризующую её производительность и надёжность. В 5 версии был добавлен специальный пакет объектов и интерфейсов, который позволяет выделить определённые группы метрик, сгруппированные по описанию определённых подсистем виртуальной машины. Мониторинг

производится за счёт вызовов методов MX-объектов, реализующих предоставленные интерфейсы. Пользователь не имеет прямого доступа к их жизненному циклу, инициализацией, контролем и уничтожением данных объектов занимается сама виртуальная машина.

Интерфейс системы выполнения обеспечивает доступ к большому списку информации, описывающей работу виртуальной машины по обеспечению выполнения приложений. Среди них выделяют:

- Пути расположения данных: расположение классов системы, сторонних библиотек, загрузчика классов и системных свойств инициализации виртуальной машины, которые используются запущенными приложениями.
- Информация виртуальной машины: содержит наименование, версию, имя поставщика текущей реализации виртуальной машины, а также аналогичную информацию о её спецификации.
- Время запуска: предоставляет время запуска виртуальной машины, приостановок и возобновлений работы в миллисекундах.
- Информация о текущем потоке: содержит уникальный идентификатор текущего потока, процессорное время, затраченное на его работу, время, которое поток находился в режиме ожидания и его текущий статус.
- Количество активных потоков: общее количество незавершённых потоков виртуальной машины, включая потоки-демоны.
- Количество потоков в режиме ожидания: включает также общее время их простоя и процессорное время, затраченное на их выполнение. Как только процесс переходит в активный статус, счётчик времени ожидания сбрасывается.

Таким образом мониторинг данных видов загрузки позволяет составить цельную картину работы программных средств и во многих случаях диагностировать способы оптимизации производительности выполнения приложений в рамках виртуальной машины.

Список использованных источников:

1. Alistair Croll Complete Web Monitoring/ Alistair Croll, Sean Power: O'Reilly Media, 2009.
2. D. Jones Creating Unified IT Monitoring and Management in Your Environment/ D. Jones: Realtime Publishers, 2012.
3. EMC Smarts: интеллектуальный мониторинг ИТ – среды и бизнес – процессов / Комптек: Москва.

АВТОМАТИЗИРОВАННЫЕ МЕТОДЫ ОБРАБОТКИ ПСИХОЛОГИЧЕСКИХ ТЕСТОВ С ВОПРОСАМИ ОТКРЫТОГО ТИПА

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

Быковский А. С.

Серебряная Л. В. – кандидат техн. наук, доцент

Условием успешного проведения психодиагностики является необходимость принятия решений в сфере измерения индивидуально-психологических особенностей человека. Основным источником информации, отражающей указанные особенности, является психологический тест и его последующая обработка.

Основной целью данной работы является разработка автоматизированных методов обработки психологических тестов с вопросами открытого типа. Методы должны давать правильные результаты при обработке нечисловых данных и быть применимы при разработке программного обеспечения. Методы должны охватывать стандартизированные методики и анкетные опросы, а так же способствовать уменьшению количества времени, необходимого для выведения психологического заключения.

Исходя из этих требований и особенностей интерпретации ответов на вопросы, было выделено три метода обработки психодиагностической информации на компьютере: автоматический, полуавтоматический и ручной. Автоматический метод хорошо подходит для закрытых вопросов, так как выбор ответа не связан с непосредственным его вводом в свободной форме и отсутствует фактор внесения непреднамеренной ошибки в ответ. В процессе автоматической обработки принимается однозначное решение о правильности ответа. После автоматической обработки будет получено психологическое заключение с указанием вычисленных выходных параметров теста. Так же данный метод подходит для открытых вопросов.

Полуавтоматический метод применяется, если в вопросе, для которого есть правильное решение, предлагается свободный ответ. Это связано с вводом текстовой информации и существует вероятность ввода правильного ответа с орфографической ошибкой, или опечаткой, а они не должны влиять на результат психологического теста. В данном варианте программная система сопоставляет ответ с эталонным, вычисляя процент соответствия. Такой процент по каждому спорному вопросу выдается, психологу при обработке результатов и предлагается указать: является ли данный ответ верным. Если между ответом респондента и эталоном найдено полное соответствие, решение о правильности ответа система принимает автоматически.

Для анкет, вопросы которых предлагают ответ-описание или ответ, выражающий отношение,

применяется ручной метод. Ручной метод полностью исключает участие компьютера в формировании психологического заключения. Обработка результатов производится следующим способом. Ответы на вопросы анкеты разбиваются на заранее известные категории и предлагаются психологу по группам для удобства ручной обработки. На основе этой информации по каждой группе вопросов психолог заполняет итоговые параметры, из которых складывается психологическое заключение. Ручной метод предлагает удобный способ представления результатов теста для выведения заключения посредством вывода сгруппированных ответов респондента и подсказок для обработки этих ответов.

Самым эффективным подходом к обработке тестов с точки зрения производительности является автоматический метод. При его использовании результаты можно получить сразу после прохождения теста испытуемым, однако данный метод не возможно применить к большинству тестов с открытым типом вопроса, что сужает количество применяемых психодиагностических методик. Полуавтоматический метод сочетает в себе эффективность автоматического метода и высокую гибкость относительно типов вопроса и обработки результатов. Данный метод охватывает большее число методик, но требует непосредственного участия психолога при обработке результатов, что увеличивает временные затраты по сравнению с автоматическим методом. Достоинствами ручного метода, является его применимость к специализированным методам исследования и возможность использования при обучении студентов-психологов. Из недостатков стоит отметить требовательность ручного метода к квалификации психолога, и высокую трудоемкость обработки.

Таким образом, было предложено три автоматизированных метода обработки психологических тестов с открытым типом вопроса. Каждый из предложенных методов справляется с задачей определения правильности ответа для стандартизованных методик и анкетных опросов. Благодаря уменьшению ручных операций в автоматическом и полуавтоматическом методах удается значительно сократить время на обработку тестов и соответственно увеличить производительность труда. Логическое описание представленных методов позволяет применить их при разработке программного обеспечения. В совокупности разработанные методы образуют подход, который охватывают широкий спектр психодиагностических методик и обеспечивают психологам новые возможности проведения исследований. Вследствие распространенности применения тестов в разных областях науки, описанные методы могут быть спроецированы на такие области знаний как педагогика и социология. Полученные методы систематизируют знания по обработке тестов с вопросами открытого типа и представляют основу для разработки алгоритмов обработки тестов.

Список использованных источников:

1. Червинская, К.Р. Компьютерная психодиагностика – СПб, 2003. – 336 с.
2. Немов, Р. С. Психология. Кн. 3: Психодиагностика. Введение в научное психологическое исследование с элементами математической статистики. – Москва, 2001. – 640 с.

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ФРОНТЕНДА ВЫСОКОНАГРУЖЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ

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

Вакульчик Е.Н.

Куликов С. С. – к.т.н., доцент

Для большинства веб-страниц менее чем 10-20% от времени отклика конечного пользователя тратится на получение HTML-документа с веб-сервера в браузер. Если необходимо резко уменьшить время отклика веб-страниц, инженер должен сосредоточиться на остальных 80-90% от времени отклика конечного пользователя.

При любой оптимизации высокой важностью обладает выяснение профиля текущей производительности с целью определения областей наибольшего улучшения. Для того, чтобы знать, что улучшать, необходимо знать, где пользователь тратит на ожидание больше всего времени. Загрузка HTML-документа занимает небольшой процент от общего времени отклика приложения, и пользователь тратит большую часть в ожидании обработки компонентов страницы: скриптов, таблиц стилей, изображений и др. Из этого следует вывод, что место фокуса – это производительность клиентской части (фронтенда). Целью данного исследования стало создание двух изначально идентичных веб-приложений, применение методов повышения производительности к одному из них и сравнение итоговых результатов.

С помощью HTTP-протокола браузер и сервер производят обмен данными, который можно ускорить путём уменьшения количества HTTP-запросов и объёма передаваемых в ответ на них данных. В данном исследовании оценивалась возможность уменьшения числа HTTP-запросов и ответов путём уменьшения числа компонентов HTML-страницы.

Размер HTTP-ответа при прочих равных условиях можно снизить, используя сжатие, если браузер и сервер поддерживают соответствующую технологию (gzip, deflate). Браузеры объявляют о своей поддержке сжатия с использованием Accept-Encoding заголовка. Серверы определяет сжатые ответы с помощью заголовка Content-Encoding.

Для случая, если браузер уже имеет копию компонента HTML-страницы в своём кэше, но по логике работы приложения мы не можем быть уверены, что она не устарела, будет выполнен условный get-запрос. Условные get-запросы и HTTP-ответы с кодом 304 (not modified) помогают ускорить загрузку страниц, но они всё ещё требуют взаимного обмена между клиентом и сервером для выполнения проверки на актуальность имеющейся у клиента информации. Expires-заголовки устраняет необходимость в такой проверке, давая понять, может ли браузер использовать свою закешированную копию компонента. Когда браузер получает Expires-заголовок в HTTP-ответе, он сохраняет дату истечения срока актуальности полученного элемента HTML-страницы и использует локальную копию на протяжении всего указанного срока.

Использование карт изображений и CSS-спрайтов позволяет объединить несколько изображений в одно, что позволяет получить все такие объединённые изображения одним HTTP-запросом, тем самым уменьшив накладные расходы на HTTP-взаимодействие.

Есть у данного метода и недостатки: определение координат области карты изображений или CSS-спрайта, выполняемое вручную при программировании клиентской части приложения, утомительно и чревато ошибками, а также почти неприменимо для любых фигур, кроме прямоугольников.

Однако в данном исследовании применение карт изображений и CSS-спрайтов оказалось на 56% быстрее, чем обработка отдельных изображений (365 миллисекунды против 822 миллисекунд).

Для компонентов, которые не являются критическим для первоначального отображения страницы, применяется отложенная загрузка, позволяющая пользователю начинать взаимодействие с клиентской частью приложения ещё до завершения передачи с сервера полного набора запрошенных данных.

В данном исследовании рассматриваются методы устранения избыточных HTTP-запросов, которые не включают сложные компромиссные решения между производительностью и дизайном: использование карт изображений, CSS-спрайтов, встроенных изображений, сочетание скриптов и таблиц стилей, настройка HTTP-запросов и ответов. Использование перечисленных методов позволяет снизить время отклика клиентской части приложения на 60-70%.

Краткий вывод по результатам данного исследования: уменьшение числа компонентов HTML-страницы и числа HTTP-запросов позволяет существенно повысить скорость отклика клиентской части приложения.

Список использованных источников:

1. Steve Souders High Performance Web Sites. – O'Really. – 2007.
2. Свободная энциклопедия Wikipedia. [Электронный ресурс]. Дата обновления: 23.01.2016. URL: <https://ru.wikipedia.org/wiki/HTTP>

МОДЕЛЬ ПРОЦЕССА УПРАВЛЕНИЯ

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

Данилова Г.В., м.т.н.

Управление сложными процессами имеет общие закономерности. В процессе управления участвуют: объект управления, орган управления и методы воздействия. На помощь процессу управления приходят программные средства, помогающие решать задачу принятия решения. Системы пошагового управления широко распространены не только на верхних уровнях управления, но и в процессах управления.

Управление сложными системами и процессами интересует человечество с давних пор. Наблюдение за многочисленными процессами выявило общие закономерности, что даёт возможность определить и описать единый базис.

Управление можно сравнить с замкнутым информационным циклом, направленным на достижение поставленной цели. В процессе управления участвуют: объект управления, орган управления и методы воздействия (рис.1).

Рис. 1 – Модель процесса управления

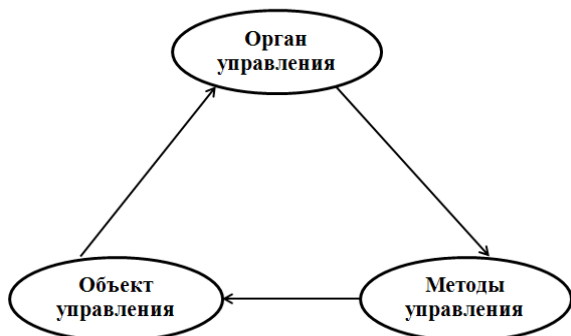
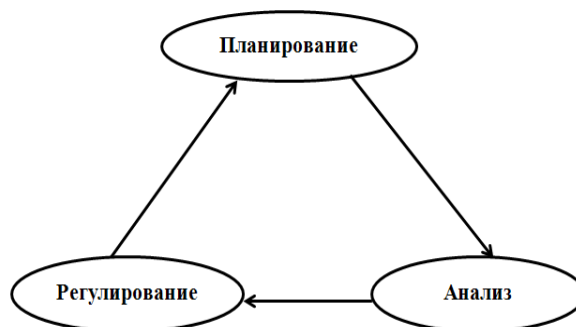


Рис. 2 – Модель достижения поставленной цели



Для того чтобы приблизиться к цели на объект управления оказывается определённое воздействие. Сведения о полученном состоянии объекта управления в виде обратной связи поступают в орган управления, который определяет следующее воздействие.

Сутью процесса обработки информации является принятие решения. Функция органа управления заключается в трансформации поступающих от объекта управления сведений и определения в связи с поставленной задачей дальнейших действий, что позволяет достигать нужных критериев эффективности.

Для нормального функционирования система должна иметь:

- цель;
- идеальную модель;
- фактическую модель;
- методы коррекции.

В связи с этим огромное значение приобретает планирование, которое определяет цель, сроки, методы управления (рис.2).

Для того чтобы приблизиться к цели на объект управления оказывается определённое воздействие. Сведения о полученном состоянии объекта управления в виде обратной связи поступают в орган управления, который определяет следующее воздействие. Принятие решения осуществляется по определенным правилам – алгоритмам. Формализация процесса заключается в построении модели, связывающую цель и исходными данными через управляющие команды.

Для того чтобы получить желаемое изменение объекта управления нужно знать характер связей между внутренними и входными данными, а также возможные отклонения как во входных данных, так и в выходных. Формальные зависимости, отражающие сложность структурных связей между входными, внутренними и выходными параметрами, определяют математическую модель объекта управления.

Система управления – понятие не материальное. Это совокупность математических моделей реального объекта управления и модели элемента управления – алгоритма или закона управления. Основная задача этого понятия – формальная или формализованная разработка закона или алгоритма управления по известной модели объекта.

На помощь процессу управления приходят программные средства, помогающие решать задачу принятия решения. В последнее время они активно входят в разные сферы человеческой деятельности. В зависимости от поставленной задачи такие системы могут работать либо в режиме реального времени, либо по заранее разработанному плану. Системы пошагового управления широко распространены не только на верхних уровнях управления, но и в процессах управления.

МОДЕЛИ И АЛГОРИТМЫ ОЦЕНКИ ЭФФЕКТИВНОСТИ БАЗ ДАННЫХ

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

Демидович Н. А.

Глухова Л. А. – канд. техн. наук, доцент

Сегодня существует множество баз данных, использующихся в различных аспектах жизнедеятельности. Актуальной является эффективность функционирования работы баз данных. В связи с этим постоянно разрабатываются модели эффективности функционирования баз данных, включающие новые меры (метрики) эффективности. В докладе рассматриваются действующие стандарты в области оценки качества программного средства, в частности характеристика эффективности и ее подхарактеристики в области разработки баз данных.

В настоящее время широко используются базы данных в различных сферах жизненной деятельности, т.к. необходимо хранить большое количество различной информации. Важной характеристикой баз данных является их эффективность функционирования, определяющая как скоростные, так и ресурсные параметры базы данных. Эффективность функционирования – зависимость функционирования от количества ресурсов, используемых в заданных условиях.

Для того чтобы оценить эффективность функционирования той или иной базы данных, необходимо исследовать существующие модели и алгоритмы оценки эффективности баз данных.

На основе оценки эффективности функционирования заказчик может сравнить несколько баз данных и выбрать из них более подходящую по скоростным и ресурсным параметрам. Также оценка эффективности позволяет определить характеристики и возможности уже используемой базы данных для дальнейшей ее оптимизации.

В настоящее время в области оценки качества программного средства на территории Республики Беларусь действуют следующие основные стандарты: межгосударственный стандарт стран СНГ ГОСТ 28806-90, межгосударственный стандарт стран СНГ ГОСТ 28195-99, национальный стандарт Беларуси СТБ ИСО/МЭК 9126-2003 и другие. Эти стандарты регламентируют выполнение оценки качества программных средств и систем на основе модели качества, представленной в виде иерархической схемы. В соответствии с

данной моделью множество свойств, которые отражают качество программного средства, представляется в виде структуры с несколькими уровнями [1, 2].

В соответствии со стандартом ISO/IEC 25010:2011 эффективность является характеристикой качества и разделяется на следующие три подхарактеристики: *поведение во времени*, *использование ресурсов* и *максимальные возможности* [1].

Для баз данных подхарактеристика *поведение во времени* позволяет оценить, насколько быстро обрабатываются запросы к базе данных. Зачастую скорость обработки запросов к базе данных является ключевым параметром оценки качества базы данных, т.к. эта подхарактеристика особенно важна для пользователя: чем быстрее обрабатывается запрос, тем быстрее пользователь получает необходимую ему информацию.

Подхарактеристика *использование ресурсов* показывает, насколько эффективно организована структура базы данных, а также насколько оптимально используется память при выполнении запросов к базе данных и при выполнении хранимых процедур. Меры этой подхарактеристики позволяют показать, где можно или нужно сократить или приумножить ресурсы для работы базы данных.

Последняя подхарактеристика (*максимальные возможности*), позволяет определить максимальные возможности баз данных по различным параметрам. Для данной подхарактеристики нет стандартизированных метрик, поэтому разработка и предложение актуальных метрик является важным для определения максимальных возможностей базы данных.

Каждой подхарактеристике соответствуют свои метрики, которые позволяют численно определить эффективность функционирования базы данных.

В докладе рассматриваются модели и алгоритмы для оценки эффективности функционирования баз данных, использующие существующие модифицированные модели. Предлагаются также новые метрики эффективности функционирования баз данных, модели эффективности функционирования и алгоритмы оценки эффективности функционирования баз данных.

Экспериментальная оценка разработанных моделей и алгоритмов обеспечивается с помощью разработанного программного обеспечения и существующей базы данных.

Список использованных источников:

1. Бахтизин В.В. Метрология, стандартизация и сертификация в информационных технологиях / В.В. Бахтизин, Л.А. Глухова, С.Н. Неборский. - Минск, БГУИР 2013. – 60 с.
2. ISO/IEC 25010:2011. Системная и программная инженерия. Требования к качеству и оценка программного продукта (SQuaRE). Модели качества систем и программных средства. – Введ 2011-03-01. – Женева Ж ISO/IEC, 2011

ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ СВЯЗИ МЕЖДУ КОМПЬЮТЕРАМИ

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

Дрозд А.Д.

Глухова Л. А. – канд. техн. наук, доцент

В докладе рассмотрены достоинства и недостатки существующих программных средств организации связи между компьютерами. Сформулированы цели и особенности реализации предлагаемого программного средства.

У современных людей существует необходимость живого общения с друзьями, деловыми партнерами, родственниками с возможностью видеть и слышать друг друга, проводить практические занятия на больших расстояниях или вести переписку в режиме реального времени. По этой причине всё большую популярность стали приобретать различные программы, обеспечивающие связь через интернет.

Для существующих прототипов, таких как TeamSpeak, WhatsApp и Skype, характерны следующие основные недостатки: отсутствие возможности совершать видео звонки (TeamSpeak), отсутствие клиента для персональных компьютеров (WhatsApp) и наличие назойливой рекламы (Skype).

Предлагаемое программное средство организации связи между компьютерами предназначено для решения комплекса задач, связанных с передачей текстовой, голосовой и видео информацией по сети Интернет.

Основными целями создания программного средства являются:

- 1) Упростить процесс общения людей – предоставить интуитивно понятный для пользователя интерфейс с возможностью совершения голосовых и видео звонков, отправки друг другу текстовых сообщений.
- 2) Предоставить пользователю возможность ввода личной информации, редактирования личной информации, удаление личной информации и настройки уровня конфиденциальности.

3) Предоставить пользователю возможность просмотра личной информации других пользователей, если это позволяет настройка конфиденциальности.

4) Предоставить пользователю информацию о других пользователях, которые находятся в сети.

5) Исключить основные недостатки существующих аналогов, приведенные выше.

Программное средство организации связи между компьютерами предусматривает выполнение основных функций:

- отправка текстовых сообщений – пользователю предоставляется список пользователей, которым он может написать;

- совершение голосового звонка – пользователь может совершать звонки только тем пользователям, которые находятся в сети;

- совершение видеозвонка – пользователь может совершать звонки только тем пользователям, которые находятся в сети;

- просмотр личной информации о другом пользователе;

- ввод своей личной информации, которая включает в себя следующие пункты:

1) фамилия, имя, отчество пользователя;

2) дата рождения;

3) место жительства пользователя;

4) место работы и должность;

5) языки, на которых общается пользователь;

6) семейное положение;

7) фотография пользователя;

- редактирование своей личной информации;

- удаление своей личной информации;

- настройка конфиденциальности;

- просмотр личной информации других пользователей.

Програмное средство организации связи между компьютерами реализовано на языке Java под управлением кроссплатформенной среды разработки Java Runtime Environment. Этим выбором достигается поддержка работоспособности программного средства под управлением 32- и 64-битных операционных систем Windows и MacOS.

Разработанное программное средство организации связи между компьютерами позволяет людям, которые находятся на большом расстоянии, слышать и видеть друг друга и отправлять друг другу текстовые сообщения. Программное средство лишено большинства недостатков существующих аналогов.

Список использованных источников:

1. Хорстманн, К. С., Корнелл, Г. Java 2. Библиотека профессионала, том 1. Основы / К. С. Хорстманн, Г. Корнелл. –М.: Вильямс, 2007. – 896 с

2. Танненбаум, Э. Компьютерные сети. – 5-е изд. / Э. Танненбаум, Д. Уззеролл. – СПб.: Питер, 2012. – 960 с.

АЛГОРИТМ ПРОГНОЗА РЕЗУЛЬТАТОВ СПОРТИВНЫХ МАТЧЕЙ

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

Евтухович В.Ю.

Фадеева Е.Е. – ассистент

На сегодняшний день весьма популярны букмекерские конторы и другие фирмы, основанные на ставках и удаче игроков. Основное направление среди них занимает спорт, а среди спортивных категорий лидирует футбол. Основной проблемой как организаторов, так и игроков является выбор победителя, либо, если выразаться точнее, расчет вероятности результатов матча.

При разработке алгоритма следует сразу учесть, что требуется обширная и довольно подробная статистика предыдущих состязаний чемпионата. Следует иметь данные о результатах матчей, о основных действиях игроков и другие важные сведения, способные повлиять на выигрыш той или иной команды.

Самым действенным и популярным методом расчета вероятности результатов спортивных состязаний считается метод взвешенных параметров (коэффициентов). Он подразумевает, что существуют некоторые важные параметры команд, которые в большей или меньшей степени влияют на победу команды в матче.

Исходя из статьи Джона Годдарда и Иоаниса Асимокопулоса «Прогнозирование футбольных результатов и эффективные фиксированные ставки» для «Journal of Forecasting» можно выделить 5 основных параметров для расчета вероятности:

1) Вероятность победы – данный параметр показывает, отношение победных матчей команды к общему количеству состязаний за последние k лет. Статистические исследования показали, что наиболее оптимальным значением для k является два года, т.е. текущий и предыдущий сезон. Большее количество лет ведет к ухудшению точности прогноза;

2) Результаты m последних домашних и n гостевых игр – данный параметр показывает, как команда играет непосредственно перед матчем. В вышеуказанной статье уточняется, что оптимальным значением для m и n является 9;

3) Географическое положение команд – данный параметр включается в выборку исходя из того, что по статистике команды, преодолевающие большие расстояния перед игрой играют хуже, чем команды, играющие на домашнем поле или на поле, расположенном не очень далеко;

4) Вместимость домашних стадионов – данный параметр учитывает такой фактор, как «доминирование большой команды». По статистике команды, играющие лучше, имеют большее количество фанатов. Кроме того, они привлекают лучших инвесторов и спонсоров. Вследствие чего, они имеют возможность строить большие стадионы и привлекать еще большее количество фанатов. Из всего это следует, что команды, с большей вместимостью домашних стадионов, выигрывают чаще.

5) Результаты предыдущих встреч команд – данный фактор указывает, как команды играли между собой в предыдущих матчах;

Для расчета коэффициентов данных параметров была использована нейронная сеть типа персептрон. Данная нейронная сеть между слоями входных и выходных нейронов имеет связи, проходя по которым параметры умножаются на некоторый коэффициент. При обучении нейронной сети данный коэффициент принимает такое значение, что при прогоне тестовых выборок, на выходе получается значение, наиболее приближенное к правильному.

По представленным выше параметрам из статистических данных были выбраны необходимые значения. Тестирование системы производилось на данных Английской Премьер Лиги. После просчета вероятности большого количества матчей и сравнения его с реальными результатами была получена точность прогноза в 63,5%. Данный результат уже превышает случайное распределение, однако он недостаточно эффективен для профессиональных систем, в которых необходима точность не менее 65%.

Для улучшения алгоритма было решено добавить в систему еще два фактора:

1) Класс игроков команды – влияние отдельных игроков на выигрыш команды достаточно велик. Однако просчет статистических данных всех заслуг игроков за предыдущие матчи – это слишком нетривиальная задача. Чтобы несколько ее облегчить было решено использовать рейтинги игроков, которые составляются специализированными агентствами. Более того, было решено использовать средний рейтинг игроков команды.

2) Разница мячей – данный параметр используется исходя из того, что забитый мяч в каждой игре повышает вероятность победы команды в матче, а пропущенный – понижает. При большом количестве забитых мячей и малом количестве пропущенных команда имеет больше шансов на победу.

При добавлении данных параметров в систему, переобучении нейронной сети и проверке ее снова на тестовой выборке был получен результат в 65,9% точности прогноза.

Таким образом была разработана модель прогноза результатов спортивных матчей, которая может использоваться в целях расчета вероятности победы той или иной команды.

Список использованных источников:

1. J. Goddard and I. Asimakopoulos. Forecasting football results and the efficiency of fixed-odds betting. Journal of Forecasting, 23(1):51–66, 2004.
2. Искусство ставить на спорт: первое пособие на русском языке, Альпина Паблицер. – Москва, 2016. – 167 с.

РАСПОЗНАВАНИЕ ВОЗДУШНЫХ ЦЕЛЕЙ ПО РАДИОЛОКАЦИОННЫМ ПОРТРЕТАМ С ПОМОЩЬЮ ЭВРИСТИЧЕСКОЙ ВОЗМОЖНОСТНОЙ КЛАСТЕРИЗАЦИИ

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

Ерома А. П.

Вятченин Д. А. – канд. филос. наук, доцент

В условиях современного воздушного боя оказывается важным своевременное обнаружение и идентификация воздушных целей средствами РТВ и ПВО для последующего целеуказания и наведения ракет.

Высокая эффективность нечетких и возможностных методов кластеризации при обнаружении и идентификации воздушных целей была продемонстрирована в работах [1] и [2]. Для генерирования радиолокационных портретов воздушных целей был выбран программный комплекс, предложенный в [3].

На рисунке 1 в качестве примера приведен сглаженный дальностный портрет самолета В-52Н, который представляет собой нормированное распределение мощности отраженного сигнала от цели в ракурсе размером 20 дискрет, что позволяет интерпретировать его как нечеткое множество, заданное на соответствующем универсуме.

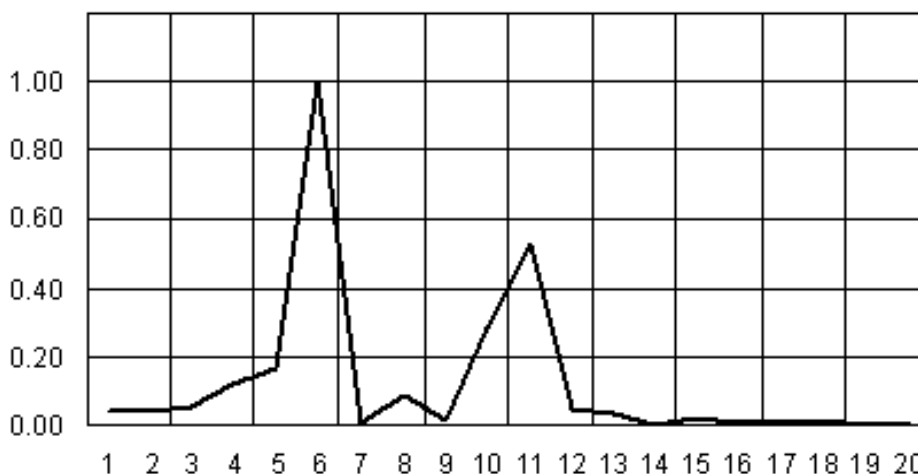


Рис. 1 – Сглаженный дальностный радиолокационный портрет самолета В-52Н в ракурсе 80°

В порядке вычислительного эксперимента была проведена иерархическая классификация 18 радиолокационных портретов самолета В-52Н с помощью эвристического H-AFC-TC-алгоритма возможностной кластеризации [2,3]. Кроме того, в работе [5] был предложен прямой реляционный D-AFC(u)-алгоритм, не требующий априорного задания классов

Таким образом, была разработана методика обнаружения целей по их радиолокационным портретам без априори задаваемого числа классов в исследуемой совокупности. Результаты кластеризации могут использоваться как самостоятельно для идентификации целей, так и для обучения нечеткого классификатора типа Мамдани, механизм генерирования которого предложен в [4].

Список использованных источников:

1. Вятчин Д.А., Вятчин В.А. . Перспективы применения методов нечеткой кластеризации в военных целях / Д.А. Вятчин, В.А. Вятчин. // Военная мысль – 2011. – № 1. – С. 46-55.
2. Вятчин Д.А., Хижняк А.В., Шевяков А.В. Нечеткая кластеризация и нечеткая математическая морфология в задачах обработки изображений. / Д.А. Вятчин, А.В. Хижняк, А.В. Шевяков // Мн: Издательство Военной Академии Республики Беларусь, 2012. – 289 с.
3. Radar Target Backscattering Simulation: Software and User's Manual / S.A. Gorshkov, S.P. Leshchenko, V.M. Orlenko, S.Yu. Sedyshev, Y.D. Shirman // Artech House: London, 2001. – 71 p.
4. Viattchenin, D.A. A Heuristic Approach to Possibilistic Clustering: Algorithms and Applications / D.A. Viattchenin. – Springer-Verlag: Berlin, 2013. – 227 p.
5. Viattchenin, D.A., Yaroma, A., Damaratski, A. A novel direct relational heuristic algorithm of possibilistic clustering / D.A. Viattchenin, A. Yaroma, A. Damaratski // International Journal of Computer Applications. – 2014. – Vol. 107, No. 18. – P. 15-21.

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ПРОСТРАНСТВЕННЫХ ДАННЫХ

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

Жукович Я.П.

Блинов И.Н. – кандидат физ.-мат. наук, доцент

В общем случае, под интеллектуальным анализом пространственных данных подразумевается процесс извлечения неявных знаний, пространственных отношений и обнаружения характеристик и шаблонов, которые не представлены явно в базах данных. Данные техники могут сыграть важную роль в понимании пространственных данных и в обнаружении сложных взаимоотношений между пространственными и непространственными данными. Более того, такие взаимоотношения могут быть использованы для представления данных в сжатом виде и для реорганизации пространственных баз данных. Интеллектуальный анализ пространственных данных применяется во многих областях, среди которых можно упомянуть геоинформационные системы, анализ баз данных изображений и другие.

Пространственные данные представляют собой данные о пространственных объектах, которые состоят из координатной и атрибутивных частей. Установление связи между этими частями называется геокодированием. Координатные данные определяют позиционные характеристики пространственного

объекта. Атрибутивные данные представляют собой совокупность непозиционных характеристик пространственного объекта и определяют смысловое содержание объекта и могут содержать качественные или количественные значения.

Главным отличием интеллектуального анализа пространственных данных от анализа данных, не имеющих пространственных атрибутов, является пространственная автокорреляция – пространственные объекты могут оказывать влияние друг на друга, что требуется учитывать. Важнейшим критерием выполнения интеллектуального анализа пространственных данных является эффективность алгоритмов, на которую влияют значительный объем пространственных данных, сложность пространственных типов данных и доступа к ним. Таким образом выполнение интеллектуального анализа пространственных данных представляет собой намного более сложную вычислительную задачу, которая может быть решена с большей эффективностью с использованием распределенных методов.

Распределенное окружение, выполняющее интеллектуальный анализ распределенных данных, является нетривиальной проблемой из-за большого количества ограничений, таких как: ограниченная пропускная способность сети (например, беспроводные сети), распределенные вычислительные ресурсы, безопасность данных и многие другие. Можно сделать предположение, что традиционные методы интеллектуального анализа будут работать неэффективно в распределенном окружении, где централизация данных является проблемой.

На данный момент некоторые компании занимаются разработкой приложений для распределенного интеллектуального анализа (РИА) данных в различных сферах, включая распределенные базы данных, основанные на сенсорных сетях, диагностика состояния автомобилей и другие. Однако, в области РИА до сих пор остается ряд нерешенных проблем, которые нуждаются в дальнейшем исследовании. В первую очередь, многие приложения в реальном мире имеют дело со сценариями распределения данных, которые не являются ни гетеро-, ни гомогенными: могут иметь место гетерогенные узлы данных, которые используют более, чем одну колонку, может не иметься явно определенного ключа, который связывает несколько записей в различных узлах. Требуется более детальная проработка алгоритмов для гетерогенных сценариев. Также требуются дальнейшие исследования в области предварительной обработки распределенных данных, основанной на метаописании.

Зачастую для РИА необходим обмен шаблонами между участвующими узлами. Таким образом, бесшовная и прозрачная реализация технологии РИА потребует стандартизованных схем для представления обмена шаблонами. В качестве отправной точки могут быть рассмотрены язык разметки прогнозного моделирования (PMML), межотраслевой стандарт для интеллектуального анализа данных (CRISP-DM) и другие связанные проекты.

Также в литературе упоминаются следующие возможные направления исследований в области интеллектуального анализа пространственных данных:

1. Использование объектно-ориентированных пространственных баз данных: многие исследователи упоминают, что объектно-ориентированные модели могут представлять собой лучший выбор для работы с пространственными данными, чем традиционные реляционные модели.

2. Чередующееся обобщение: интересным является рассмотрение чередующихся пространственных и непространственных обобщений, для получения результата более эффективным образом.

Список использованных источников:

1. Цветков В.Я. Пространственные знания // Международный журнал прикладных и фундаментальных исследований. 2013. No 7. С. 43–47.
2. Маркелов В.М., Цветков В.Я. Модели получения знаний в геоинформатике // Славянский форум. 2015. No 1 (7). С. 177–182.
3. Benjamin Kuipers. Modeling Spatial Knowledge // Cognitive Science. – 1978. – No 2. – P. 129–153.

ПЕРЕХОД К ИНТЕРАКТИВНОМУ ОБУЧЕНИЮ

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

Каранцевич К.И.

Данилова Г.В., м.т.н.

Потребность развития цивилизации в древние времена воплотилась в жизнь в виде школ. Первые школы появились в Древнем Египте, и посещать их могли только юноши из влиятельных семей. По некоторым данным в V в. до н.э. все свободные греки имели образование, т.е. школьное обучение было обязательным. В древних школах обучали математике, географии, астрономии, медицине и др. Так же, как и в наше время, имелись разные ступени обучения: младшая школа, средняя. Высшее образование же, например, в Древней Греции представляло собой некие группы людей, которые следовали за своим учителем. В качестве учителя выступали образованные люди, имеющие за плечами огромный опыт.

С течением времени школьное образование начали систематизировать, анализировать, вследствие чего появились различные способы обучения, которые используются и по сей день. В практике существуют большое количество сценариев определения методов обучения, которые основаны на степени осознанности восприятия учебного материала:

- пассивные;
- активные;
- интерактивные и прочие.

До недавнего времени при обучении самой распространённой формой являлся пассивный метод. Пассивный метод – это форма взаимодействия учителя и учащихся, в которой учитель является основным действующим лицом и управляющим ходом урока, а учащиеся выступают в роли пассивных слушателей. Несмотря на то, что данный метод имеет достаточную распространённость, современное образование старается шагнуть вперед и перейти к активному и интерактивному обучению. Активный метод – это форма взаимодействия учащихся и учителя, при которой учитель и учащиеся работают друг с другом в ходе урока и учащиеся здесь не пассивные слушатели, а активные участники урока. И, наконец, последний метод – интерактивный – означает взаимодействовать, находиться в режиме беседы, диалога с кем-либо. Другими словами, в отличие от активных методов, интерактивные ориентированы на более широкое взаимодействие в режиме реального времени учеников не только с учителем, но и друг с другом и на преобладании активности учащихся в процессе обучения.

Пассивность обучающегося во время занятия заключается в том, что он только слушает преподавателя, в то время как педагогические и социологические исследования показывают, что от пассивного участия в процессе обучения очень скоро не остается и следа. Существует определенная закономерность запоминания предоставленной информации, описанная американскими исследователями Р. Карникау и Ф. Макэлроу: человек помнит 10% прочитанного; 20% – услышанного; 30% – увиденного; 50% – увиденного и услышанного; 80% – того, что впоследствии скажет сам; 90% – того, что он понял в процессе деятельности. В связи с этим задача преподавателя инициировать процесс обучения (самообучения), поддерживать интерес и внимание в заданных областях, а также создавать условия для взаимодействия в учебной среде [1].

В рамках дипломного проекта будет разработано программное средство, основной функцией которого является взаимодействие преподавателя и студента, как на лекции, так и во внелекционное время. В основу заложены две главные идеи: обратная связь – студент-лектор; отслеживание активности студента и его знаний преподавателем.

Обратная связь решает проблему пассивного обучения, предоставляет студентам возможность задать интересующие вопросы и получить на них достоверные ответы от студентов либо самого преподавателя, не нарушая хода лекции. Таким образом, данный способ образования станет объединением пассивного и интерактивных методов обучения.

В свою очередь, преподаватель будет иметь возможность оценить в полной мере знания студента, с помощью анализа задаваемых вопросов в ходе лекции и ответов на поставленные вопросы, в том числе за счет проверки результатов прохождения тестов студентами в конце каждой лекции. Данное программное средство позволит преподавателю оценить, насколько студент усвоил пройденную тему и насколько он был активен во время семестра.

В результате дипломного проектирования будет разработано программное средство, которое поможет лучше организовать учебный процесс со стороны преподавателя и вызвать активность студентов. Следствием такого обучения студентами будет лучше усваиваться полученный материал.

Список использованных источников:

1. Данилова, Г. В. Средства формирования компетенций в IT-сфере./ Сб. материалов VII Международной научно-методической конференции «Высшее техническое образование: проблемы и пути развития» – Минск, БГУИР, 2014.

ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ

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

Кирвель Е.А.

Иванюк А.А. – д.т.н, профессор

В докладе рассмотрены проблемы и особенности реализации физически неклонируемых функций, а также описаны методы оценки качества физически неклонируемых функций. Рассматриваются технические проблемы, связанные с реализацией статических тестов NIST, и методы их решения.

В современном мире информационных технологиях все большее внимание стоит уделять защите информации. Криптографические методы применяются практически во всех информационных технологиях и охватывают полный стек промышленной разработки: начиная от цифровых устройств и заканчивая протоколами передачи данных в веб-технологиях. Физическая криптография является одним из наиболее современных достижений в области криптографии и защиты информации. Доминирующей категорией физической криптографии являются физически неклонлируемые функции (ФНФ). Существуют множества вариантов реализации физически неклонлируемых функций. В данном докладе рассмотрен вариант реализации физически неклонлируемой функции типа арбитр, для анализа которой и было применено описываемое программное средство. Для определения качества ФНФ обычно используют следующие параметры: уникальность, надежность, случайность. Для расчета каждой из них имеются свои подходы и особенности. К примеру для расчета случайности часто прибегают к статическим тестам Д. Кнута, тестам Дж Марсальи, Национального института стандартов США (NIST). Описываемое программное средство реализует статические тесты NIST. На данный момент имеются реализации тестов NIST на языке C++. Однако эти пакеты имеют ряд недостатков: имеют ошибки в реализации алгоритмов, алгоритмы не оптимизированы, нечитабельный код, данные тесты рассчитаны под решение специфических задач и не могут быть применены в качестве универсального средства проверки уникальности. При использовании потоков, можно значительно улучшить производительность данных тестов. В настоящее время не существует такого программного средства, которое агрегировало бы в себе тесты для ФНФ, позволяло бы в удобной форме получать результаты и проводить анализ.

Основными целями создания программного средства являются:

- 1) Обеспечить возможность расчёта основных метрик ФНФ таких как уникальность, надежность.
- 2) Реализовать статические тесты NIST используя .NET Framework.
- 3) Добиться улучшения производительности тестов NIST за счет распараллеливания операций.
- 4) Реализовать возможность получения результатов в виде графиков.

В ходе реализации все поставленные цели были достигнуты. Таким образом, разработанное программное средство автоматизации тестирования физически неклонлируемых функций позволяет осуществлять расчет основных метрик ФНФ, а также упрощает анализ полученных результатов.

Список использованных источников:

1. Ярмолик В.Н., Вашинко Ю.Г. Физически неклонлируемые функции / Информатика, №2, 2011 г. 12с..
2. Клыбик В.П. Применение физически неклонлируемой функции типа арбитр для решения задачи идентификации цифровых устройств / В.П Клыбик, А.А. Иванюк // Информатика. – 2015., - №3 – С 24-34
3. S.S Zalivaka, A.V. Puchkov, V.P. Klubik, A.A. Ivaniuk, C.H. Chang Multi-valued arbiters for Quality Enhancement of PUF Responses on FPGA Implementation.

АВТОМАТИЗАЦИЯ ОБРАБОТКИ ИНФОРМАЦИИ В СИСТЕМЕ МЕДИАПЛАНИРОВАНИЯ

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

Ковалева А. А.

Глухова Л. А. – к.т.н., доцент

В работе современных рекламных агентств все больше времени и средств уделяется повышению производительности и качества работы специалистов по медиапланированию. Одним из способов достижения данной цели является автоматизация процесса обработки информации и применение компьютерной логики к формированию выводов.

Автоматизация играет важную роль во всех отраслях производственной деятельности. Это так же относится к рекламной деятельности, а именно к той области работы рекламной компании, которая отвечает за медиапланирование. Автоматизация позволяет усовершенствовать и ускорить работы рекламных агентств, используя искусственный интеллект для выявления закономерностей и принятия некоторых решений. Также она уменьшает вероятность принятия ошибочных решений, зачастую приводящих к тому, что рекламный бюджет расходуется неэффективно.

Медиапланирование – это совокупность решений и действий, направленных на эффективное доведение рекламных сообщений до потребителей, подразумевающая выбор оптимальных каналов размещения рекламы, формирование рекламного графика размещения рекламных сообщений, продолжительность рекламной кампании. Разработка медиаплана является одной из самых важных частей медиапланирования. Медиаплан – это целевой программный документ, определенным образом структурированный, и представляющий собой систему расчетов, обоснований, описание мер и действий по работе со средствами массовой информации с учетом максимальной эффективности при определенном уровне затрат. В медиаплане рекламной кампании описываются выбранные типы рекламы, сроки размещения, стоимость, а также эффективность проводимой рекламной кампании. Эффективным медиапланированием

можно назвать такое медиапланирование, при котором рекламное сообщение будет размещено в максимально выгодное время, в наиболее подходящем для целевой аудитории носителе, в выигрышном месте, оптимальным размером, необходимое число раз при минимальном бюджете.

Причины необходимости внедрения автоматизации в процесс медиапланирования:

1. Сложность и, отчасти, невозможность обработки вручную огромных массивов данных, которые порождаются в результате сбора рекламной информации;

2. Низкая скорость обработки данных;

3. Вероятность допущения ошибки за счет невнимательности, недостаточно глубокого знания методик, неоднозначного понимания формул и др.;

4. Постоянный мониторинг динамически изменяющихся данных;

5. Высокая стоимость человеческих ресурсов в сравнении с использованием программных средств;

6. Сложность выявления всех допустимых путей развития рекламной компании продукта.

В настоящее время существует множество систем для автоматизации обработки информации в медиапланировании, например Galileo, TV Planet, Super Nova. Однако, их главным недостатком является анализ информации в рамках только одного типа медиа, они "привязаны" к конкретным базам данных маркетинговых исследований, которые из-за разной методики сбора данных формируются отдельно для каждого типа СМИ.

Для автоматизации обработки информации в медиапланировании предлагается система, которая состоит из модуля по загрузке данных и модуля по обработке данных. На вход модуля по загрузке данных подаются Excel файлы. На выходе данного модуля формируются данные, преобразованные к виду, необходимому для качественной работы модуля по обработке данных. Таким образом, выходные данные модуля по загрузке данных являются входными для модуля по обработке данных. На выходе модуля по обработке данных формируется отчет, который может в последствии быть дополнен и усовершенствован. На его основе специалисты по медиапланированию будут строить медиаплан.

Предлагаемая система позволит облегчить работу специалистам по медиапланированию. Медиаагентства смогут быстрее формировать рекламные стратегии, что в рамках жесткой конкуренции на рынке является одним из наиболее важных моментов. Благодаря системе можно будет извлечь максимум полезной информации из разнородных, неполных, неточных, противоречивых, косвенных знаний, выявить закономерности, которые затруднительно обнаружить при традиционном просмотре данных (по причине слабых связей или чрезмерного объема данных). В результате совместной работы специалиста по медиапланированию и данной системы значительно возрастет производительность ведения рекламной кампании продукта и будут учтены все возможные варианты ее развития.

Список использованных источников:

1. Кочеткова А. В. Медиапланирование : учебное пособие / А. В. Кочеткова – М.: РИП-холдинг, 2003. – 176 с.
2. Хромов Л. Рекламная деятельность: искусство, теория и практика / Л. Хромов – М.: ИНФРА-М, 2003. – 342 с.
3. Бузин, В. Н. Медиапланирование для практиков / В. Н. Бузин [и др.]. – М.: Вершина, 2006. – 448 с.

РАСШИРЯЕМОЕ ПРОГРАММИРОВАНИЕ ПРИЛОЖЕНИЙ MICROSOFT OFFICE

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

Коваль Д. А.

Куликов С. С. – к.т.н, доцент

В связи с компьютеризацией во всех отраслях человеческой деятельности возрастает спрос на средства электронного документооборота и расширение возможностей этих средств. Одно из самых популярных средств электронного документооборота – пакет приложений Microsoft Office, предоставляющий COM API для расширения своих возможностей. Однако, при наличии системы модулей расширения встает проблема организации межмодульного взаимодействия и совместного доступа к сервисам, а также проблема масштабируемости такой системы.

Компьютерная обработка данных в современных офисах стала основным видом обработки информации. Ввиду разнородности самих данных существует набор приложений, специализированных под конкретные типы данных. Однако целесообразно использовать не набор отдельных программ, а интегрированные пакеты офисного обслуживания, так как в них реализуется не просто объединение больших автономных программ в пакеты, а их интеграция в прикладные программные комплексы, означающая их полную унификацию – общий пользовательский интерфейс и единообразные подходы к решению типовых задач.

На рынке программного обеспечения в сфере прикладных офисных продуктов доминирует пакет приложений Microsoft Office. В состав пакета входит программное обеспечение для работы с различными типами документов – текстами, электронными таблицами, презентациями.

В последнее время наблюдается тенденция к компьютеризации во всех отраслях человеческой деятельности, вследствие чего необходимость в автоматизации механизмов работы с электронными документами возрастает.

Помимо возрастающего спроса на средства управления электронными документами, возникают потребности в расширении возможностей этих средств в соответствии со специализированными нуждами конкретных потребителей. Поэтому в большинство таких программных продуктов заложена возможность для динамического расширения своих возможностей за счёт сторонних модулей, динамически распознаваемых, загружаемых и выгружаемых во время работы главного приложения.

Приложения Microsoft Office предоставляют возможности для своего расширения, экспортируя набор COM-интерфейсов. Однако, при разработке системы модулей расширения независимыми коллективами программистов были обнаружены следующие проблемы:

- расширение неуправляемых приложений Microsoft Office (созданных на языке программирования C++) как неуправляемыми модулями, так и управляемыми (созданными на языке программирования C#) и их интеграция

- децентрализация интерфейсов прикладного программирования;
- отсутствие возможности взаимодействия модулей (в случае, если несколько модулей решают какую-то общую задачу)

- отсутствие возможности гибкого управления политикой изоляции управляемого модуля;
- отсутствие возможности совместного использования сервисов с общей для всех модулей функциональностью;
- проблемы с производительностью в случае увеличения количества модулей расширения (масштабируемость).

Поэтому задача по автоматизации разработки системы модулей расширения приложений Microsoft Office является актуальной в сфере разработке программного обеспечения.

Решением проблем автоматизации разработки модулей расширения приложений Microsoft Office на языках программирования C# и C++ является разработанное инструментальное программное средство CommonOffice.

На рисунке 1 показана диаграмма развёртывания программного средства CommonOffice. Фреймворк CommonOffice является неуправляемым COM-модулем расширения приложений Microsoft Office. Его задачи включают:

- хостинг CLR-среды в памяти хост-приложения;
- загрузка неуправляемых и управляемых модулей расширения в память хост-приложения;
- изоляция модулей расширения от хост-приложения (хост-приложение взаимодействует только с COM-модулем фреймворка CommonOffice) и от друг друга;
- обеспечение взаимодействия модулей расширения с хост-приложением, другими модулями (модули могут быть изолированы доменами приложений или размещены в другом процессе) и сервисами (размещёнными в скрытом процессе Daemon).

Модель предполагает наличие базовых сервисов для обеспечения консистентного состояния между процессами, обеспечения межпроцессного взаимодействия на основе событий, отображения пользовательских интерфейсов. Набор сервисов может быть расширен путём создания дополнительных сервисов (например, сервиса аутентификации), используя соответствующие интерфейсы.

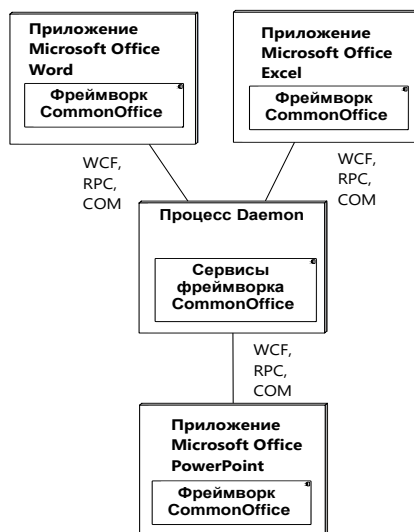
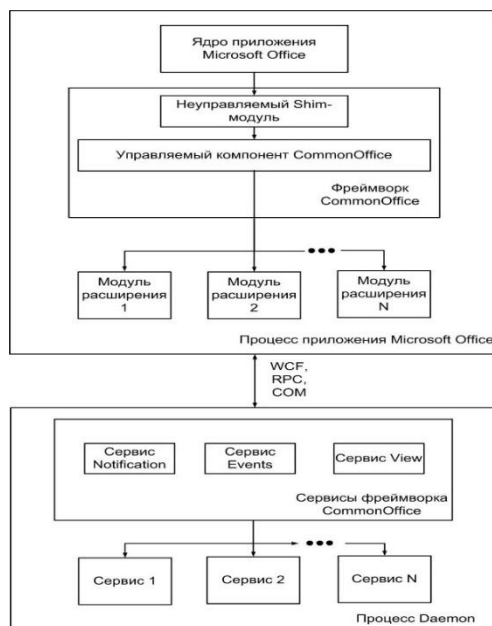


Рис. 1 – Диаграмма развёртывания программного средства CommonOffice

Таким образом, фреймворк CommonOffice, с одной стороны, взаимодействует с приложением Microsoft Office и является его модулем расширения, а с другой стороны – экспортирует интерфейсы прикладного программирования для создания модулей расширения и агрегирует эти модули (рисунок 2).

Решением проблемы падения производительности является замена модели последовательного

обслуживания модулей (модель Microsoft Office) на модель, при которой модули обслуживаются параллельно (параллельное обслуживание модулей берет на себя фреймворк CommonOffice).



Список использованных источников:

1. Слепцова, Л. Программирование на VBA в Microsoft Office 2010 / Л. Слепцова. – М. : Вильямс, 2010. – 432 с.
2. Thangaswamy, V. VSTO 3.0 for Office 2007 Programming / V. Thangaswamy – Packt Publishing, – 2009 – 260 p.

ПЕРСПЕКТИВЫ АВТОМАТИЗИРОВАНИЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ ПОЖАРНОЙ БЕЗОПАСНОСТИ

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

Корначева Т.А., старший инспектор НИИ ПБ и ЧС МЧС Беларуси

Ю.С.Иванов, к.т.н., НИИ ПБ и ЧС МЧС Беларуси

Технические нормативно-правовые акты (далее – ТНПА) в области пожарной безопасности направлены на предотвращение возникновения пожара, а также воздействия на людей его опасных факторов. В то же время ТНПА не должны противоречить экономическим критериям эффективности на всех стадиях жизненного цикла объекта (научная разработка, проектирование, строительство, эксплуатация) [1]. Одним из важных этапов при проектировании и строительстве объектов является обеспечение пожарной безопасности, в рамках которого необходимо осуществить организационно-технические и инженерные мероприятия. В зависимости от назначения объекта требования по пожарной безопасности аккумулируются из различных ТНПА.

В области противопожарного нормирования и стандартизации распространены два основных метода нормирования: нормативный – установление минимально-допустимых критериев согласно правилам и требованиям нормативов и функционально-ориентированный (расчетный) метод, который позволяет максимально уточнить требования пожарной безопасности для конкретного объекта, учитывая его экономическую эффективность и целесообразность [2].

Услуги по осуществлению экспертной деятельности, такие как разработка паспортов пожарной безопасности, проведение расчетов по обеспечению (оценке) пожарной безопасности (либо выборка из указанного перечня работ), относятся к лицензируемым видам деятельности согласно п. 10.7 [3]. Большое значение имеет наличие инженерно-технической квалификации и знаний в области предотвращения пожаров у специалистов, выполняющих данные расчеты, т.к. любые ошибки могут привести к неверным интерпретациям расчетных результатов и, как следствие, – к снижению уровня противопожарной защиты.

В настоящее время в Республике Беларусь автоматизировано несколько расчетных методик в области противопожарного нормирования и стандартизации, которые размещены на web-сайте МЧС [4]. Однако, эти программные средства (далее – ПС) имеют ряд недостатков: 1. разработаны для локального использования и исключают возможность работы с ними через web-доступ, что не гарантирует использование пользователем актуальной версии программы; 2. не являются кроссплатформенными и работают только на операционной системе (далее – ОС) MicrosoftWindows, пользователи других ОС (в том числе мобильных) не могут воспользоваться этими программами; 3. для решения комплексных задач потребуется использование сразу нескольких приложений, в то же время функционал некоторых из них может быть объединен в одном ПС; 4. отсутствует возможность сохранения истории выполненных расчетов или создания шаблонов расчетов для их применения при формировании отчета.

В целях создания более совершенного программного обеспечения ведется разработка информационно-вычислительного комплекса (далее – ИВК) на базе web-технологий с доступом через сеть Интернет. ИВК позволит централизовать и унифицировать проведение инженерных расчетов и существенно сократить трудозатраты на экспертизу проектной документации. Использование ИВК работниками инспекции государственного пожарного надзора предоставит возможность осуществить достоверную экспресс-проверку инженерных расчетов в области противопожарного нормирования и стандартизации в сжатые сроки при осуществлении надзорной деятельности.

Список использованных источников:

1. Система стандартов безопасности труда. Пожарная безопасность. Общие требования: ГОСТ 12.1.004-91. – Введ. с 01.07.92 / Государственный стандарт союза ССР. – М.: Издательство стандартов, 1992. – 91 с.
2. Татарников С. Расчетные методы обеспечения пожарной безопасности при проектировании зданий и сооружений / С.Татарников // Служба спасения 101. – 2010. – №7(151). – С.27-29
3. О лицензировании отдельных видов деятельности: Указ Президента Республики Беларусь 1 сентября 2010 г. № 450. – М.: Зар-но в Нац. реестре правовых актов Республики Беларусь 03.09.2010 г. № 1/11914
4. Расчетные программы для проектировщиков [Электронный ресурс] / Официальный сайт Министерства по чрезвычайным ситуациям Республики Беларусь – Режим доступа: <http://mchs.gov.by/rus/main/business/programs>. – Дата доступа: 21.03.2016.

ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ И ИХ ПРИМЕНЕНИЕ В КРИПТОГРАФИИ

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

Короткевич А. В.

Ярмолик В. Н. – д-р. техн. наук, профессор

Обеспечение безопасности информации является одной из важнейших проблем современного общества. Такая безопасность может быть достигнута с использованием различных криптографических методов, одними из самых передовых и перспективных среди которых являются криптосистемы, основанные на свойствах эллиптических кривых.

В общем случае уравнение эллиптической кривой имеет вид:

$$y^2 + axy + by = x^3 + cx^2 + dx + e,$$

где x, y – переменные; a, b, c, d, e – действительные числа.

Для определения операции сложения для точек на эллиптической кривой сделаем следующие предположения:

1. На плоскости существует бесконечно удаленная точка $0 \in E$, в которой сходятся все вертикальные прямые.

2. Будем считать, что касательная к кривой пересекает точку касания два раза.

3. Если три точки эллиптической кривой лежат на прямой линии, то их сумма есть 0.

Введем следующие правила сложения точек на эллиптической кривой [1]:

1. Точка 0 выступает в роли нулевого элемента. Так $0 = -0$ и для любой точки P на эллиптической кривой $P + 0 = P$.

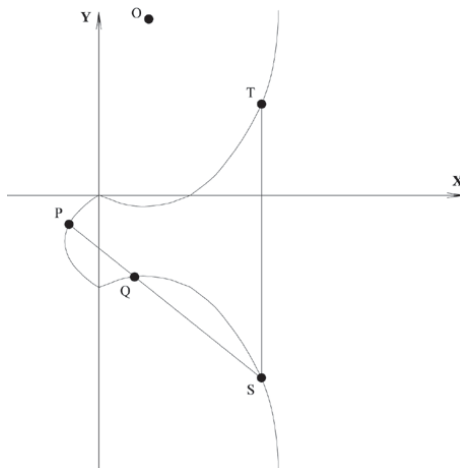
2. Вертикальная линия пересекает кривую в двух точках с одной и той же координатой x – скажем, $S = (x, y)$ и $T = (x, -y)$. Эта прямая пересекает кривую и в бесконечно удаленной точке. Поэтому $S + T + 0 = 0$ и $S = -T$.

3. Чтобы сложить две точки P и Q с разными координатами x , следует провести через эти точки прямую и найти точку пересечения ее с эллиптической кривой (рисунок 1). Если прямая не является касательной к кривой в точках P или Q , то существует только одна такая точка, обозначим ее S . Согласно нашему предположению $P + Q + S = 0$. Следовательно, $P + Q = -S$ или $P + Q = T$.

Чтобы удвоить точку Q следует провести касательную в точке Q и найти другую точку пересечения S с эллиптической кривой. Тогда $Q + Q = 2Q = -S$.

Введенная таким образом операция сложения подчиняется всем обычным правилам сложения, в частности коммутативному и ассоциативному законам. Умножение точки P эллиптической кривой на положительное число k определяется как сумма k точек P .

Как известно, стойкость алгоритмов асимметричной криптографии базируется на вычислительной невозможности эффективного решения некоторых математических задач. Например, стойкость криптосистемы



RSA базируется на сложности задачи факторизации больших чисел, а стойкость современных схем ЭЦП, большинство из которых являются вариациями обобщенной схемы Эль-Гамала, – на сложности задачи логарифмирования в конечных полях.

Рис. 1 – Сложение точек на эллиптической кривой

Для эллиптических криптосистем такой трудноразрешимой задачей является нахождение числа k , если известна точка P и точка, полученная в результате умножения точки P на k (задача дискретного логарифмирования на эллиптической кривой). Таким образом, операция умножения точки P на число k является ключевой в эллиптической криптографии, потому скорость ее выполнения напрямую влияет на производительность всей криптосистемы в целом. А именно высокая вычислительная сложность является основным недостатком асимметричных криптосистем по сравнению с симметричными.

Для оптимизации операции вычисления произведения точки эллиптической группы P на число k можно предложить следующие подходы:

1. Бинарное представление числа k в виде массива бит длиной t . На начальном шаге итоговому результату присваивается значение нулевой точки. Затем при проходе бинарного массива, начиная со старшего индекса, на каждом шаге результирующее значение удваивается, а после этого, если текущий бит равен 1, дополнительно увеличивается на P . В результате итоговое значение будет содержать сумму произведений точки P на числа, являющиеся степенью числа 2 и в сумме дающие k , что эквивалентно искомому произведению kP . Среднее число единиц в бинарном представлении числа k равно $t/2$, потому

сложность данного алгоритма можно представить как $\frac{t}{2}A + tD$, где A и D – сложности операций сложения и

умножения соответственно.

2. Использование представления числа k с помощью несмежных форм (NAF – non-adjacent form). Несмежная форма числа k обладает следующими свойствами [2]:

- k имеет уникальную несмежную форму, обозначаемую как $NAF(k)$;
- $NAF(k)$ содержит минимально возможное количество ненулевых чисел в своем представлении;
- длина $NAF(k)$ в худшем случае на 1 больше бинарного представления числа k ;
- если длину $NAF(k)$ обозначить как l , то $2^l/3 < k < 2^{l+1}/3$;

– среднее количество ненулевых цифр среди всех несмежных форм длины l составляет приблизительно $l/3$.

Как следует из свойств несмежной формы k , вычислительная сложность данного алгоритма составляет приблизительно $\frac{t}{3}A + tD$, где t – длина бинарного представления числа k , A и D – сложности операций

сложения и умножения соответственно. Следовательно, данный алгоритм требует на $t/6$ меньше операций сложения, чем алгоритм, основанный на бинарном представлении числа k , однако, требует предварительного вычисления $NAF(k)$. А раз алгоритм вычисления $NAF(k)$ обладает логарифмической сложностью относительно k , что незначительно по сравнению с операцией сложения точек эллиптической группы, то в целом алгоритм на основе несмежных форм оказывается эффективнее.

3. Предыдущий подход может быть улучшен путем обработки нескольких цифр числа k одновременно (так называемые оконные методы), т.е. с использованием несмежных форм ширины w . Пусть $w \geq 2$ – целое положительное число. Несмежная форма ширины w положительного числа k выражается как $k = \sum_{i=0}^{l-1} k_i 2^i$, где

каждое ненулевое значение k_i нечетно, $|k_i| < 2^{w-1}$, $k_{l-1} \neq 0$ и по крайней мере одно из подряд идущих k чисел является ненулевым [3].

Свойства несмежной формы ширины w (k – целое положительное число):

- k имеет уникальную несмежную форму ширины w , обозначаемую как $NAF_w(k)$;
- $NAF_2(k) = NAF(k)$;
- длина $NAF_w(k)$ по крайней мере на единицу больше длины бинарного представления k ;
- средняя плотность ненулевых чисел среди всех несмежных форм ширины w с длиной l составляет приблизительно $1/(w+1)$.

4. Если точка P является фиксированной для нескольких последовательных операций умножения, то можно скорость выполнения умножения может быть увеличена с использованием дополнительной памяти для хранения предварительно вычисленных данных, зависящих только от P . Например, если все точки $2P, 2^2P, \dots, 2^{t-1}P$ вычислены заранее, то ожидаемое время выполнения бинарного метода составит $(t/2)A$ (все удвоения устранены). Использование предварительных вычислений для фиксированной точки также может быть успешно скомбинировано и с другими предыдущими методами.

Таким образом, была исследована математическая модель эллиптических кривых и их применимость в криптографических системах. Выделена основная операция (умножение точки эллиптической группы на число), применяемая в криптографических системах на базе эллиптических кривых, и предложены пути увеличения скорости ее выполнения. В планы дальнейших исследований входит практическая проверка предложенных методов оптимизации алгоритма умножения точки эллиптической группы на число, а также разработка криптографической системы, основанной на свойствах эллиптических кривых, и анализ ее производительности.

Список использованных источников:

1. Henri, C. Elliptic and Hyperelliptic Curve Cryptography: Theory and Practice / C. Henri, F. Gerhard. – Chapman & Hall/CRC, 2006. – 808 с.
2. Аграновский, А.В. Практическая криптография: алгоритмы и их программирование / А. В. Аграновский, Р. А. Хади – М.: СОЛОН-Пресс, 2010. – 256 с.
3. Hankerson, D. Guide to elliptic curve cryptography / D. Hankerson, A. Menezes, S. Vanstone – Springer-Verlag New York, Inc, 2004. – с. 99.

ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ ТРЕНИНГА ПО КУРСУ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

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

Косик Д.Н.

Глухова Л.А. – к.т.н, доцент

В докладе рассмотрены достоинства и недостатки существующих обучающих систем. Сформулированы цели и особенности реализации программного средства организации тренинга по курсу «Основы алгоритмизации и программирования».

Для изучения нового материала, усовершенствования уже полученных знаний и приобретения практического опыта помимо выполнения лабораторных работ в университете необходимо заниматься самостоятельно. По этой причине всё большую популярность стали приобретать различные тренинговые системы, онлайн-курсы и вебинары. При анализе прототипов, таких как MyTestXPro и Quizful, был выявлен ряд недостатков: ограниченность в количестве проходимых тестов (Quizful), невозможность работы с программой в режиме online, платная лицензия для пользования программой по истечении пробного периода (MyTestXPro). Поэтому с целью устранения вышеназванных недостатков разработано программное средство организации тренинга по курсу «Основы алгоритмизации и программирования», предназначенное для автоматизации комплекса задач, связанных с усовершенствованием полученных знаний на лекциях и практических занятиях, изучением нового материала по курсу, получением практического опыта, а также проведением контроля знаний по данной дисциплине.

Основными целями создания программного средства являются:

- 1) Упростить процесс тестирования студентов – предоставить интуитивно понятный для пользователя интерфейс с возможностью проведения тестирования по курсу «Основы алгоритмизации и программирования».
- 2) Предоставить студентам быстрый доступ к теоретическим материалам и практическим примерам по заданной дисциплине.
- 3) Предоставить возможность для преподавателя (или администратора) создания новых тестов, редактирования уже существующих и удаления устаревших тестов.

4) Обеспечить для администратора возможность оперативного доступа в любое время к необходимой информации программного средства.

Программное средство организации тренинга по курсу «Основы алгоритмизации и программирования» предусматривает выполнение основных функций:

- 1) просмотр всех доступных тестов;
- 2) просмотр теоретических материалов по выбранному тесту – перед каждым тестом студенту предлагается изучить теоретический материал, разобрать примеры по теме теста;
- 3) прохождение теста студентом – после изучения теоретического материала и разбора примеров студент имеет возможность пройти выбранный тест;
- 4) просмотр результатов тестирования с подробным объяснением ответов на задания теста;
- 5) создание нового теста преподавателем (или администратором) – возможность добавления нового теста, который включает в себя: теоретический материал по теме создаваемого теста; примеры (если допускается); тест с вариантами ответов и указанием правильного ответа; пояснение правильного ответа к каждому заданию теста;
- 6) редактирование и удаление теста преподавателем (или администратором);
- 7) просмотр информации о пройденных тестах и полученной оценке за их прохождение.

Поддержка работоспособности программного средства под управлением 32- и 64-битных операционных систем Windows и MacOS достигается выбором платформы и языка написания программного средства. В данном случае программное средство реализовано на языке Java под управлением кроссплатформенной среды разработки Java Runtime Environment. В качестве сервера к программному средству выступает отдельно работающий веб-сервер Apache Tomcat, принимающий и посылающий запросы в формате HTTP.

Таким образом, разработанное программное средство организации тренинга по курсу «Основы алгоритмизации и программирования» упрощает процесс тестирования знаний студентов, позволяет студентам совершенствовать уже полученные знания и легко изучать новый материал. Программное средство может быть использовано студентами всех форм обучения, в том числе студентами дистанционной формы.

Список использованных источников:

1. Хорстманн, К. С., Корнелл, Г. Java 2. Библиотека профессионала, том 1. Основы / К.С. Хорстманн, Г. Корнелл. – М.: Вильямс, 2007. – 896 с.
2. Глухова, Л. А. Основы алгоритмизации и программирования : лаборатор. практикум для студентов специальности I-40 01 01 «Программное обеспечение информ. технологий» днев. формы обучения : в 4 ч. Ч. 2 / Л. А. Глухова, Е. П. Фадеева, Е.Е.. Фадеева. – Мн. : БГУИР, 2005. – 52 с.

ЯЗЫК ПРОГРАММИРОВАНИЯ ДЛЯ ОПИСАНИЯ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ ПРИНЦИПОВ DOMAIN- DRIVEN DESIGN

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

Костевич А.А.

Блинов И.Н. – кандидат физ.-мат. наук

Любое программное обеспечение связано с той или иной предметной областью. Построение модели предметной области, ее отображение в архитектуре системы и программная реализация являются одними из наиболее сложных и трудоемких задач проектирования. Создание языка программирования для описания модели предметной области на основе принципов Domain-Driven Design, позволило бы существенно упростить процесс моделирования.

Проектирование любой программной системы включает в себя создание модели предметной области, в которой будет работать пользователь. Возрастающая естественная сложность моделируемых предметных областей и, соответственно, моделей и архитектур систем обусловили появление нового подхода к проектированию – Domain-Driven Design (DDD).

DDD представляет собой набор приемов, подходов и принципов проектирования для ускорения разработки ПО для сложных областей деятельности, основывающихся на двух основных тезисах:

В большинстве программных проектах основное внимание должно быть сосредоточено на логической структуре предметной области и взаимосвязях с ней.

Архитектура сложного программного обеспечения должна основываться на модели предметной области.

Модель, согласно DDD, представляет собой “строго организованную выборку знаний”, представляющую собой приблизительное отображение реальности, удобную для ее программной

реализации. Модель позволяет выразить сложность процессов и богатство понятий той или иной предметной области в конструкциях языка программирования.

По словам Э. Эванса, автора DDD, “нужны четкие методы исследования моделей предметных областей и выражения их в работающих прикладных программах”. Современные высокоуровневые языки программирования общего назначения (C#, F#, Java, Scala, Swift) предоставляют богатый набор возможностей для выражения моделей различных предметных областей прикладной систем в коде программы, однако требуют для этого дополнительных трудозатрат и строгости на этапе проектирования.

Одним из способов увеличить производительность труда разработчиков и предоставить новые возможности по исследования предметных областей является создание нового языка программирования для описания модели предметной области, реализующего в себе основные принципы DDD: предметно-ориентированного языка программирования (domain-driven programming language).

Данное исследование нацелено на проектирование предметно-ориентированного языка программирования на основании объектно-ориентированного парадигмы и принципов Domain-Driven Design. На данном этапе, основными задачами проектирования предметно-ориентированного языка программирования являются:

Реализовать концепции объектно-ориентированной парадигмы программирования.

Предоставить типы для основных элементов модели предметной области: объектов-значений, сущностей и корней агрегации. Поведение и характеристики каждого элемента модели должны предоставляться соответствующим типом.

Предоставить специализированный синтаксис для описания служб и фабрик модели предметной области.

Предоставить возможность идентифицировать ограниченные контексты модели предметной области.

Предоставить возможность разбиения модели предметной области на модули.

Для интеграции с существующими инструментами проектирования и разработки исследуется компиляция исходного кода в Common Intermediate Language для платформы .NET или Java bytecode для платформы Java.

Список использованных источников:

1. V. Vernon Implementing Domain-Driven Design / V. Vernon – Addison-Wesley, 2013 – 612 p.
2. Эванс, Э. Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем / Э. Эванс – Вильямс, 2011. – 448с.
3. V. Vernon Effective Aggregate Design [Electronic resource] / V. Vernon – Mode of access: http://dddcommunity.org/library/vernon_2011

ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА СИГНАЛА СПЕКТРА, ПРИМЕНЯЕМАЯ В СИСТЕМЕ ПОДДЕРЖКИ ЛАЗЕРНОЙ ЭКСПРЕССНОЙ ЭКСПЕРТИЗЫ

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

Лазарь Д. В.

Рудикова Л. В. – д-р. техн. наук, профессор

В настоящее время немаловажное значение в научных исследованиях, связанных с экспрессной лазерной экспертизой, играет обработка полученных данных снятых спектров за короткое время, их накопление в базе данных и дальнейшее использование на различных этапах лазерной экспрессной экспертизы. Хранение информации в базе данных подразумевает определенный формат данных.

Предварительная обработка спектра необходима для выделения полезной информации из сигнала спектра для последующего приведения к определенному формату и сохранении в базу данных. Под полезной информацией подразумеваются пики, их интенсивность и площадь.

Алгоритм поиска пиков основан на методе деконволюции сигналов и осуществляется в несколько этапов.

1. Вычисление фона сигнала. Результатом является информация, содержащая пики и фон. Данный метод основан на SNIP-алгоритме (Sensitive Nonlinear Iterative Peak):

$$V_p(i) = \min \left\{ V_{p-1}(i), \frac{[V_{p-1}(i+p) + V_{p-1}(i-p)]}{2} \right\}$$

где P – номер итерации.

2. Вычисление сглаженного сигнала с помощью алгоритма, который базируется на дискретной цепи Маркова, которая имеет простое инвариантное распределение:

$$U_2 = \frac{P_{1,2}}{P_{2,1}} U_1, U_3 = \frac{P_{1,2}}{P_{2,1}} U_2 U_1, \dots, U_n = \frac{P_{n-1,n}}{P_{n,n-1}} U_{n-1} \dots U_2 U_1$$

где U_1 – определяется из нормализованного условия (1.12):

$$\sum_{i=1}^n U_1 = 1$$

3. Вероятность изменения позиции пика от канала i к каналу $i + 1$:

$$P_{i,i\pm 1} = A_i \sum_{k=1}^m \exp\left(\frac{y(i \pm k) - y(i)}{y(i \pm k) + y(i)}\right)$$

где A_i – такая константа, что выполняется равенство (1.14):

$$P_{i,i-1} + P_{i,i+1} = 1$$

4. Вычисляется вектор функции g , необходимой для решения уравнения свертки (1.15) (с учетом параметра функции поиска пиков):

$$f * g = h$$

где h – записанный сигнал, f – восстановленный сигнал, g имеет вид:

$$G(i) = 1000 * e^{-\frac{[i-3*s]^2}{2*s^2}}$$

где s – параметр функции поиска пиков.

Поиск пиков производится на восстановленном сигнале с учетом порогового значения, указанного в процентах. Пороговое значение высчитывается из значения наибольшего пика и означает, что пики, имеющие значение меньше порогового, не учитываются.

Для вычисления высоты и площади найденных пиков используется уровень фона, полученный ранее на первом этапе.

Для вычисления площади пика используется следующий алгоритм:

1. Вычисление крайних точек пересечения гистограммы и фона.
2. Разбиение площади пика на треугольники, как указано на рисунке 1.5 (зеленая линия – данные спектра, красная – вычисленный фон).
3. Вычисление площади треугольников.
4. Площадь пика считается сумма площадей треугольников.

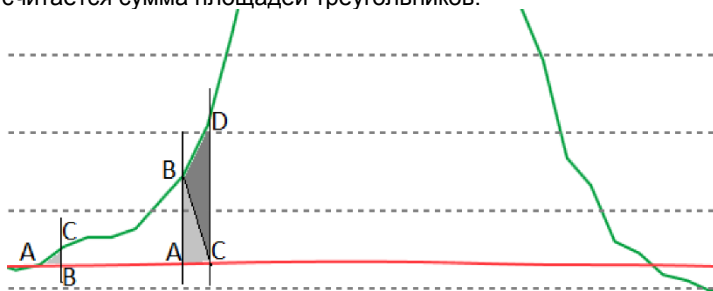


Рис. 1 – Разбиение площади пика на треугольники

Таким образом, был определен метод предварительной обработки спектра. Стоит отметить, что данный метод позволяет извлекать полезную информацию из секторов, а именно информацию о пиках,

Список использованных источников:

1. Ryan, C. G. and Clayton, E. and Griffin, W. L. and Sie, S. H. and Cousens, D. R., SNIP, a statistics-sensitive background treatment for the quantitative analysis of PIXE spectra in geoscience applications;

ОБ ОБЩИХ ПОДХОДАХ К РАЗРАБОТКЕ ГРАФИЧЕСКОГО ФРЕЙМВОРКА ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ

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

Ломакин Г.А.

Рудикова Л.В. – канд. физ.-мат. наук, доцент

Предлагается альтернативный вариант построения графического фреймворка для мобильной платформы на основе ОС Android. Охватываются основные аспекты реализации программного обеспечения, связанного визуализацией трехмерных объектов. Дается описание общей архитектуры разрабатываемого фреймворка.

В современном мире существуют сложившиеся и устоявшиеся подходы к разработке крупных графических приложений. Более крупные компании самостоятельно разрабатывают необходимые решения, а для всех остальных существует масса универсальных решений, например Unity 3D. В силу своей универсальности подобные платформы поддерживают несколько операционных систем или даже устройств. Кроме того, они предназначены для всевозможного применения, что сказывается, в первую очередь, на производительности. Такие решения также обычно поставляются с закрытым кодом и содержат ровно те функциональности, которые были в них заложены, т.е. без возможности не затратного расширения. Исходя из вышеизложенного, было решено разработать фреймворк на OpenGL с поддержкой одной платформы Android, который предоставляет открытый код и возможность легкого расширения. Кроме того, предлагаемый функционал должен быть в достаточной мере гибким и используемым. Абстрактная модель строится таким образом, чтобы все наиболее используемые возможности были легко доступны пользователю.

Одной из основных проблем при разработке подобных решений является реализация решения, отвечающего за позиционирование объектов в пространстве. В качестве такого решения были выбраны и доработаны кватернионы и системы кватернионов.

Положительные аспекты предлагаемого решения – математика кватернионов проще и более стабильная при реализации, чем углы Эйлера. Однако, это влечет немного большие затраты по производительности.

Наиболее интересное графическое решение было выбрано для генерации освещения в реальном времени. Алгоритм основан на базе классического Shadowmapping с некоторыми доработками для сглаживания теней. Данный алгоритм описывает отрисовку теней для сцены в реальном времени. Алгоритм состоит из двух этапов – генерация карты глубины из направленного источника света и отрисовки сцены от лица камеры с учетом карты затенения.

Первый этап. Вначале сгенерируется карта глубины из источника света. Ниже пойдет речь о направленном источнике света. Для начала необходимо понимать природу теней. Если говорить об одном источнике освещения, то в тени будет все то, что «не видно» с позиции источника света с учетом его направления. Отсюда следует необходимость сгенерировать карту глубины из источника освещения. Таким образом, получаем данные о затенении для последующего рендеринга. Для точечного источника света генерируется CubeMap из шести текстур глубины – верх-низ-право-влево-вперед-назад.

Второй этап. Зная карту глубин и расположение источника света, можно приступить к рендерингу. При отрисовке необходимо учитывать глубину, но, чтобы правильно ее извлечь, необходимо позицию конечного пикселя привести в систему координат источника освещения. Для этого необходимо определить и передать в шейдер матрицу MVP для источника освещения.

После применения матрицы к позиции пикселя получим 2D-координаты в системе освещения. Используя эти координаты, можно получить глубину затенения в этой точке и, если выбранный пиксель окажется с меньшей глубиной относительно источника света, чем сгенерированная ранее глубина в этой позиции, то пиксель будет освещен, иначе – затенен.

Главная идея предлагаемого фреймворка – расширенные возможности по сборке контента.

Изначально определяются несколько отдельных сущностей – Текстура, Шейдер, Меш, Логика обновления, Логика отрисовки, для которых разрабатывается механизм, позволяющий комбинировать все эти сущности между собой и получать на выходе требуемый результат.

Архитектура строится таким образом, что все эти сущности не связаны между собой. Основу разработки составляют несколько, так называемых, Store-хранилищ, в которых содержатся основные данные, заложенные в основу отрисовок. Итак, выделены три основных хранилища – Меши (3D-модели, Mesh), Текстуры и Шейдеры. При необходимости можно добавить также хранилище и для другого типа хранимых объектов. Все хранилища наследуются от базового хранилища, которое является абстрактным и типизируемым, которое также поддерживает целостность данных, защищено от OutOfMemory и других исключительных ситуаций. Рассмотрим указанные сущности.

Mesh. Включается в себя наборы данных о вершинах – текстурные координаты, вектора нормали для каждой вершины и индексы для отрисовки. Для Mesh реализован специальный класс MeshLoader, который загружает их из файла (формат MyGL, разработан специально для движка с целью минимизации данных в файлах описания Mesh), а также собирает Mesh из массивов текстурных координат нормалей и остальных данных. MeshLoader адаптирован под работу с нативными структурами данных, так как структуры данных в Java содержат избыточную реализацию и проблемы, связанные с производительностью при их использовании.

Texture. Загружается из форматов png, jpg, gif. Важной особенностью является то, что размеры текстуры должны быть кратны степеням числа 2, так как не все графические адаптеры работают с текстурами других размеров.

Shaders. Класс-обертка над программами, выполняемыми напрямую на GPU. Создаются с использованием ShaderLoader, который компилирует шейдеры из файловой системы и связывает с программой для GPU. Shaders также содержит всю логику по инициализации конкретных шейдеров и предоставляет гибкие возможности для использования и масштабирования.

Логика обновления. Перед тем как отрисовывать объект, необходимо учесть прошедшее время и рассчитать его положение в пространстве, а также другие изменения, которые могли произойти за это время. Для разных объектов реализуется совершенно разная логика поведения. Однако можно выделить общие абстрактные зависимости. Например, объекты должны реагировать на столкновения с другими объектами. Для таких целей разработан CollisionController, который обрабатывает такие события.

Логика отрисовки. Разные объекты необходимо отрисовывать по-разному, учитывая тени, альфа-канал, материал, постобработку, а также их комбинации. В силу этого логика отрисовки вынесена как отдельная сущность, общие черты которой можно выявить и составить эффективную иерархию наследования.

Итак, обобщая все изложенное выше, получаем следующую структуру. Разработанное ядро CoreX содержит в себе информацию о текущей загруженной сцене, а также хранилища для всего контента [1, 2]. Сцена представляется собой набор объектов (комбинаций сущностей), которые «знают», как себя вести за все время своего жизненного цикла. Предлагаемая конструкция позволяет достаточно просто генерировать многообразие объектов, используя абстрактные типы поведения и отрисовки, связывая их с текстурами, мешами и произвольным шейдером для материала. Кроме того, поддерживается этап постобработки, информация для которого может быть сгенерирована в процессе отрисовки конкретных объектов.

Список использованных источников:

1. Ломакин, Г.А. Об одном подходе к построению графических приложений / Г.А. Ломакин // Наука-2011 : сб науч. ст. В 2ч. Ч. 2 / БГУ ; рекол.: В.И. Малюгин (отв. ред.) [и др.] – Минск : БГУ, 2011. – С. 89-91.
2. Ломакин, Г.А. О разработке веб-среды, поддерживающей создание 3D-приложений / Г.А. Ломакин // Веб-программирование и Интернет-технологии WebConf2012: материалы 2-й Междунар. науч.-практ. конф., 5–7 июня 2012 г., С. 233-234. инск. – Минск Изд. центр БГУ, 2012. – С. 164-165.

ПОСТРОЕНИЕ ИНФРАСТРУКТУРЫ СИСТЕМЫ АВТОМАТИЗАЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

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

Мойса Д. С

Серебряная Л.В — к.т.н, доцент

В докладе рассматривается способ построения инфраструктуры поддержки жизненного цикла программного обеспечения, а также необходимые организационные изменения для внедрения такого способа работы с инфраструктурой.

В современных процессах разработки очень важное место занимает инфраструктура поддержки и автоматизации этих процессов, а также сама среда работы продукта. Наиболее часто используемые системы: контроля версий, сборки, непрерывной интеграции, развертывания, тестирования и т. д. Очень важно, чтобы такие системы работали бесперебойно, так как нарушение их работы может негативно повлиять на процесс разработки в целом, а иногда и полностью остановить его. Наиболее частые факторы риска: ошибки конфигурации, неработоспособность систем в течение некоторого времени или их полная неработоспособность.

Ошибки в таких системах часто связаны с человеческим фактором. Чаще всего их настройка и поддержка происходит в ручном режиме, что влечет за собой высокую вероятность внесения ошибки. Также нередки ситуации, когда при смене ключевого сотрудника не происходит передача необходимых знаний новому работнику, и информация о необходимой конфигурации остается потерянной.

При серьезных сбоях возможно полное разрушение системы. В этом случае единственным выходом является настройка этой системы с нуля. Все это время процесс разработки, поддерживаемый поврежденной системой, будет нарушен и могут возникать сложности вплоть до полной остановки разработки. Таким образом, надежность инфраструктуры поддержки процессов разработки является важной и актуальной задачей.

Наиболее часто применяемыми мерами для сохранения и быстрого восстановления систем являются резервные копии, однако восстановление из такой копии может также занять длительное время. Кроме того, копия может быть устаревшей, либо содержать только данные, не включающие информацию о конфигурации.

Представление инфраструктуры в виде кода — это процесс написания сценариев от создания виртуальной машины и установки операционной системы до установки и настройки программного обеспечения, связей между экземплярами и других параметров конфигурации. Среда, описанная сценариями, позволяет применять одну и ту же конфигурацию к одному узлу или нескольким узлам. Автоматизация инфраструктуры вносит гибкость как в процесс разработки, так и в эксплуатацию, потому что любой уполномоченный член команды может изменять сценарии с применением к инфраструктуре надлежащих методов разработки, таких как автоматизированное тестирование и управление версиями. Это значит, что к описанию инфраструктуры можно применять те же метрики эффективности и качества, что и к коду приложений.

Важным моментом в применении данного принципа является то, что необходимо обеспечить порядок применения изменений кода инфраструктуры к самой инфраструктуре. Для минимизации рисков необходимо организовать полный цикл работы с кодом инфраструктуры:

- версионирование кода инфраструктуры – исходный код хранится в системе контроля версий;
- ревизия и обсуждение кода ключевыми членами команды перед помещением в репозиторий (код-ревью);

- модульное тестирование (где возможно) – позволяет убедиться, что отдельные модули и подмодули после изменения кода возвращают ожидаемые результаты;

- автоматизированное функциональное тестирование кода инфраструктуры;

- автоматизированное развертывание – инфраструктура разворачивает и тестирует саму себя;

- мониторинг развернутой инфраструктуры и контроль за процессом извне;

Таким образом, организация вышеописанного процесса позволяет добиться следующих преимуществ:

1. Быстрое внесение и применение изменений. Любое изменение – это изменение кода, которое, проходя через все системы автоматически тестируется и применяется.

2. Возможность быстрого развертывания всей инфраструктуры или ее части с нуля.

3. Самодокументируемость конфигурации инфраструктуры.

4. Версионность инфраструктуры.

5. Повышение надежности.

Следовательно, предложенный подход позволяет существенно упростить настройку и поддержку инфраструктуры жизненного цикла программного обеспечения, а также заметно снизить риски, связанные с ней.

Список использованных источников:

1. Morris K. Infrastructure as Code — O'Reilly, 2016.

2. Taylor M. Learning Chef. A Guide to Configuration Management and Automation — O'Reilly, 2014

МЕТОД ОБНАРУЖЕНИЯ УЯЗВИМОСТЕЙ К SQL-ИНЪЕКЦИЯМ В WEB-ПРИЛОЖЕНИЯХ

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

Оношко Д.Е.

Бахтизин В.В. — к.т.н., профессор

Широкое использование web-приложений и повышение их сложности для решения новых задач поднимают значимость оценки их качества на новый уровень. Обнаружение уязвимостей к SQL-инъекциям, являющихся распространённым видом уязвимостей web-приложений, представляет собой важную часть такой оценки.

По состоянию на 2013 год по данным OWASP наиболее распространённым классом уязвимостей существующих приложений являются уязвимости-инъекции [1]. В web-приложениях наиболее частым видом инъекций являются SQL-инъекции. Ошибки, допущенные разработчиками при разработке алгоритмов обработки данных, позволяют злоумышленнику изменить поведение web-приложения путём формирования запросов, использующих эти ошибки, и являются причиной уязвимости web-приложений к SQL-инъекциям.

Большинство методов обнаружения уязвимостей к SQL-инъекциям основывается на анализе поведения web-приложения в условиях эксплуатации, т.е. на динамическом анализе. Однако такие методы по своей сути являются частным случаем функционального тестирования, и, следовательно, их эффективность определяется выбором тестовых данных, для которых производится анализ поведения, что в свою очередь означает невозможность обнаружения всех возможных ошибок данного класса. Статический анализ, т.е. анализ исходных кодов без запуска web-приложения, напротив, позволяет, последовательно анализируя отдельные части web-приложения, формально доказать корректность алгоритмов обработки данных либо выявить ошибки в них, а значит, получить ответ на вопрос о наличии в web-приложении эксплуатируемых и потенциальных уязвимостей к SQL-инъекциям. Вместе с тем, являясь рутинной процедурой, подобный анализ может быть автоматизирован посредством программного средства контроля качества кода, основанного на абстрактной интерпретации исходных кодов web-приложения.

Предлагаемый метод обнаружения уязвимостей основан на рассмотрении web-приложения как множества процедур $Q = \{P_1, P_2, \dots, P_N\}$, вызывающих друг друга с некоторыми параметрами. Обмен данными между процедурами может реализовываться в различных языках программирования по-разному, поэтому при использовании метода предлагается приводить все возможные взаимодействия к одному из двух видов формальных параметров: in-параметры (данные, передаваемые в процедуру) и out-параметры (данные, возвращаемые из процедуры). Формальным параметрам процедур назначаются оценки, в простейшем случае — бинарного характера: «опасный» (U) или «безопасный» (S), причём $U < S$.

Первоначально анализатору известны оценки только для стандартных процедур. Пусть на i -м шаге известны оценки параметров и возвращаемых значений для процедур $Q_i = \{P_1, P_2, \dots, P_{C(i)}\}$, $Q_i \subseteq Q$, где $C(i) < N$ — количество таких процедур на i -м шаге. Поскольку все процедуры принадлежат одному и тому же приложению, существует процедура $P_{C(i)+1}$, зависящая только от процедур из Q_i . Получение оценок для её параметров может быть сведено к последовательному анализу её операторов с целью назначения оценок для используемых в ней переменных. Вычисленные оценки параметров процедуры P_N (им соответствуют поступающие от пользователя данные) должны иметь значение U . Параметры, для которых это не выполняется, являются потенциально уязвимыми. Анализируя путь, по которому была получена оценка, можно выявить причину возникновения уязвимости и предложить способы её устранения.

В общем случае можно сформулировать 5 правил, описывающих процесс назначения оценок и проверки web-приложения на наличие уязвимостей к SQL-инъекциям.

1. Оценка для переменной, передаваемой в качестве in-параметра, должна быть лучше (выше) оценки данного параметра.

2. Оценка переменной, принимающей значение из out-параметра, выбирается как худшая (меньшая) из её собственной оценки в данной точке программы и оценки out-параметра.

3. Всем литералам назначаются оценки S .

4. При наличии у переменной или out-параметра нескольких различных оценок выбирается наихудшая (наименьшая).

5. При наличии у in-параметра нескольких различных оценок выбирается наилучшая (наибольшая).

Предлагаемый метод позволяет обнаруживать как реальные (эксплуатируемые при нынешнем состоянии кода web-приложения) уязвимости, так и потенциальные. Собранные с ходе анализа web-приложения с помощью данного метода сведения могут использоваться в оценке качества web-приложения.

Список использованных источников

1. OWASP Top 10-2013. The Ten Most Critical Web Application Security Risks [Электронный ресурс]. — Режим доступа: <http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf>. — Дата доступа: 01.03.2016.

СНИЖЕНИЕ РАЗМЕРНОСТИ ИССЛЕДУЕМЫХ ДАННЫХ НА ОСНОВЕ ЭВРИСТИЧЕСКОЙ ВОЗМОЖНОСТНОЙ КЛАСТЕРИЗАЦИИ

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

Петюкевич Н.С.

Вятченин Д. А. – канд. филос. наук, доцент

При необходимости обработки больших массивов данных на первой стадии возникает задача предварительного анализа имеющихся данных для:

- определения возможного числа классов, на которые «расслаивается» исследуемая совокупность;
- выделения в ней аномальных наблюдений;
- проецирования исследуемой совокупности на плоскость, к примеру, двух главных компонент или наиболее информативных признаков.

С этой целью может быть применен аппарат эвристической возможностной кластеризации.

Разведочный анализ данных традиционно используется, когда, с одной стороны, у аналитика имеется таблица многомерных данных, а с другой – информация о природе этих данных неполна или вовсе отсутствует. К задачам разведочного анализа данных традиционно [1] относят:

- проецирование данных в двумерное пространство;
- обнаружение аномальных наблюдений и очистка данных;
- обнаружение возможного числа кластеров в исследуемой совокупности.

При этом проецирование исследуемой совокупности в двумерное признаковое пространство предполагает вначале нахождение соответствующих признаков, на плоскость которых будет проецироваться исследуемая совокупность.

В последние 20-30 лет огромную популярность получили методы автоматической классификации, основанные на концепции теории нечетких множеств, предложенной выдающимся американским математиком Л.А. Заде [2], что обусловлено их высокой точностью и содержательной осмысленностью в сравнении с традиционными методами кластеризации. Одним из таких подходов стал эвристический метод возможностной кластеризации, предложенный в работе [3] и получивший дальнейшее развитие в задачах классификации и

управления. В частности, в работе [4] описан D-PAFC-TC-FS-алгоритм выбора наиболее информативных признаков, на плоскость которых можно проецировать данные исследуемой совокупности, показавший высокую эффективность. Следует указать, что предложенный в [4] D-PAFC-TC-FS-алгоритм был основан на транзитивном замыкании исходной нечеткой толерантности, описывающей матрицу близости признаков исследуемой совокупности.

При этом, в работе [5] был предложен ряд эвристических возможных кластер-процедур, основанных на TAGA-алгоритме [6], строящем наилучшую транзитивную аппроксимацию исследуемой совокупности. Рассмотренный в [5] подход, отыскивающий лучшую транзитивную аппроксимацию между всеми сгенерированными матрицами, может быть рекомендован для обнаружения множества информативных признаков. С этой целью в уже существующем D-PAFC-TC-FS-алгоритме [4] вместо операции построения транзитивного замыкания следует использовать предложенный в [6] TAGA-алгоритм, что существенно улучшит отбор информативных показателей в сравнении с D-PAFC-TC-FS-алгоритмом [4].

Построенная таким образом кластер-процедура получила название D-PAFC-TAGA-FS-алгоритма, и при проведении вычислительных экспериментов, осуществленных на широко известных данных Е. Андерсона по ирисам [7], показала хорошие результаты в сравнении с методом главных компонент [1], и оказалась более простой и удобной в сравнении с кластерным методом, предложенным в [3].

Список использованных источников:

1. Айвазян, С.А. Прикладная статистика: Классификация и снижение размерности: Справ. изд. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин; Под ред. С.А. Айвазяна. – М.: Финансы и статистика, 1989. – 607 с.
2. Zadeh, L.A. Fuzzy Sets / L.A. Zadeh // Information and Control. – 1965. – Vol. 8. – P. 338-353.
3. Viattchenin, D.A. A Heuristic Approach to Possibilistic Clustering: Algorithms and Applications / D.A. Viattchenin. – Springer-Verlag, Berlin, 2013. – 227 p.
4. Kacprzyk, J., Owsinski, J.W., Viattchenin, D.A. A New Heuristic Possibilistic Clustering Algorithm for Feature Selection / J. Kacprzyk, J.W. Owsinski, D.A. Viattchenin // Journal of Automation, Mobile Robotics and Intelligent Systems. – 2014. – Vol. 8. – P. 40-46.
5. Viattchenin, D.A., Damaratski, A. Direct Heuristic Algorithms of Possibilistic Clustering Based on Transitive Approximation of Fuzzy Tolerance. / D.A. Viattchenin, A. Damaratski // Informatica Economică. – 2013 – Vol. 17. - P. 5-15.
6. Dawyndt, P., De Meyer, H., De Baets, B. UPGMA Clustering Revisited: A Weight-Driven Approach to Transitive Approximation / P. Dawyndt, H. De Meyer, B. De Baets // International Journal of Approximate Reasoning. – 2006. – Vol. 42. – P. 174-191.
7. Anderson, E. The irises of the Gaspé Peninsula / E. Anderson // Bulletin of the American Iris Society. – 1935. – Vol. 59. – P. 2-

ПРОГРАММНОЕ СРЕДСТВО ИНТЕГРАЦИИ ERP-СИСТЕМЫ С СИСТЕМОЙ ПРОВЕДЕНИЯ ПЛАТЕЖЕЙ

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

Плотник В. Е.

Глухова Л. А. – канд. техн. наук, доцент

В докладе рассмотрены достоинства и недостатки существующих программных средств интеграции ERP-системы Microsoft Dynamics AX 2012 R3 и систем проведения платежей. Сформулированы цели и особенности реализации программного средства интеграции ERP-системы Microsoft Dynamics AX 2012 R3 и системы проведения платежей РХР.

Розничная торговля в международных масштабах не может существовать без грамотной организации приёма платежей, включающей в себя не только оплату наличными денежными средствами, но и оплату различными типами пластиковых карт (Visa, MasterCard, American Express, JCB). Для осуществления безопасной и удобной работы с таким разнообразным количеством пластиковых карт существует множество компаний, предоставляющих системы по обработке платежей, таких как РХР, BluePay, АСІ.

Для успешного функционирования предприятия необходимо не только принимать оплату различными способами, но и вести учёт платежей. Стандартом учёта и планирования в современном мире являются ERP-системы, позволяющие контролировать большинство аспектов предприятия, таких как розничная торговля, бухгалтерия, производство и др.

Для осуществления учёта платежей пластиковыми картами в ERP-системе Microsoft Dynamics AX 2012 R3 необходимо использовать программное средство интеграции ERP-системы и системы проведения платежей. На данный момент на рынке существует лишь одно распространённое программное средство интеграции ERP-системы Microsoft Dynamics AX 2012 R3 с системой проведения платежей: распространяемое с Microsoft Dynamics AX 2012 R3 программное средство Dynamics Payment Connector. Однако это программное средство имеет ряд важных недостатков: поддержка сильно ограниченного количества типов платёжных систем, невозможность использования программного средства интеграции для других типов платёжных систем.

Основными целями создания предлагаемого программного средства являются:

1) Обеспечить возможность приёма платежей пластиковыми картами в модуле розничной торговли ERP-системы.

2) Предоставить возможность сохранять информацию о совершённых платежах с помощью пластиковой карты в ERP-системе.

3) Обеспечить взаимодействие сервиса платёжной системы с устройством считывания информации с пластиковой карты.

4) Предоставить администраторам ERP-системы возможность настройки взаимодействия с платёжной системой.

Программное средство интеграции ERP-системы с системой проведения платежей предусматривает выполнение следующих основных функций:

1) Формирование запроса на проведение платёжной операции к системе проведения платежей РХР.

2) Обработка ответа от системы проведения платежей РХР о статусе проведённого платежа.

3) Формирование запроса на проверку актуальности программного обеспечения устройства считывания информации с пластиковой карты.

4) Обработка выполненного платежа в формате, понятном для ERP-системы Microsoft Dynamics AX 2012 R3.

Предлагаемое программное средство интеграции состоит из следующих компонентов:

1) Модуль обработки платёжных операций для Microsoft Dynamics AX 2012 R3 for Retail POS.

2) Модуль настроек интеграции с платёжной системой для Microsoft Dynamics AX 2012 R3.

3) Модуль взаимодействия с устройством считывания информации с пластиковой карты для Microsoft Dynamics AX 2012 R3 for Retail POS.

Таким образом, предлагаемое программное средство интеграции ERP-системы Microsoft Dynamics AX 2012 R3 и системы проведения платежей РХР позволяет осуществлять хранение и учёт проведенных платежей, что упрощает управление предприятием и контроль его финансов.

Список использованных источников:

1. ERP-системы: выбор, внедрение, эксплуатация. Современное планирование и управление ресурсами предприятия / Дэниел О'Лири — М.: Вершина, 2004

2. Платонов, Ю.Г. Методы обеспечения интеграции информационных систем / Ю.Г. Платонов. — Институт систем информатики им. А.П. Ершова, 2011.

СТРУКТУРИРОВАННОЕ ФОРМАТИРОВАНИЕ КОДА НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ В СИСТЕМАХ РАСПРЕДЕЛЕННОГО КОНТРОЛЯ ВЕРСИЙ

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

Подлозный В. А.

Куликов С.С. – к.т.н., доцент

Подход к форматированию исходного кода во многих языках высокого уровня является вопросом личных предпочтений разработчика. В коллективной разработке ПО нередки ситуации, когда правки в файлы с исходным кодом вносятся несколькими разработчиками, придерживающимися, зачастую, различных стилей форматирования. В работе представлена высокоуровневая архитектура системы, позволяющей использовать индивидуальный стиль форматирования исходного кода каждым из разработчиков, при поддержании единых стандартов форматирования в удалённых репозиториях систем распределённого контроля версий.

Форматирование кода для компиляторов многих высокоуровневых языков программирования является семантически несущественным. Тем не менее, исследования показывают, что систематическое следование определённым правилам форматирования улучшает читаемость кода и может способствовать повышению качества конечного программного продукта [1], [2].

Современные интегрированные среды разработки обладают необходимым функционалом для поддержания заданного стиля форматирования кода. Выбор конкретных стилевых опций, однако, определяется личными предпочтениями разработчика, что создаёт проблему при использовании систем контроля версий (в частности, распределённых, где на рабочей машине каждого из разработчиков размещается полнофункциональная локальная копия удалённого репозитория) в коллективной разработке ПО.

Несмотря на предпринятые попытки построения утилит семантического сопоставления ревизий файлов с исходным кодом [3], распространённым инструментом просмотра истории изменений всё ещё остаются diff-подобные утилиты посимвольного сравнения. Существенным ограничением данного класса утилит является неспособность разделить правки, вносящие изменения в логику работы исходного кода, от правок, вносящих изменения форматирования, для логики работы несущественных.

Распространённым подходом к решению данной проблемы при коллективной разработке ПО является

выработка единых для всех разработчиков стандартов форматирования. Несмотря на это, с высокой вероятностью различные проекты будут иметь отличные стандарты форматирования, что вынудит разработчика всякий раз перестраиваться при работе над очередным проектом.

Альтернативой вышеназванному подходу может служить подход, основанный на использовании ассоциированных с каждым репозиторием распределённой системы контроля версий (будь то локальным для каждого из разработчиков или удалённым) специфичных опций форматирования. В процессе копирования репозитория разработчик может задать предпочитаемый стиль форматирования, отличный от такового для исходного репозитория. Каждая ревизия полученных из исходного репозитория файлов форматируются в соответствии с заданными предпочтениями.

В процессе внесения правок в исходный код, в локальном репозитории разработчик руководствуется индивидуальным стилем форматирования. При публикации правок в удалённый репозиторий, выполняется форматирование файлов в соответствии с настройками удалённого репозитория, и сохранение в таком виде под версионным контролем. Таким образом, семантически эквивалентные файлы с исходным кодом имеют различное форматирование в локальном для каждого из разработчиков репозитории, но единое в централизованном, что позволяет эффективно применять diff-подобные утилиты для выявления изменений в логике работы исходного кода.

Ограничением предложенного подхода можно считать его несовместимость с системами контроля версий, основанными на хранении записей об изменении содержимого файлов (Mercurial), а не самих изменённых файлов (Git).

Список использованных источников:

1. Buse R. P. L., Weimer W. R. A metric for software readability // Proceedings of the 2008 international symposium on Software testing and analysis. – Seattle, WA, USA: ACM, 2008. – С. 121-130.
2. Miara R. J., Musselman J. A., Navarro J. A., Shneiderman B. Program indentation and comprehensibility // Commun. ACM. – 1983. – Т. 26, № 11. – С. 861-867.
3. Jackson D., Ladd D. A. Semantic Diff: a tool for summarizing the effects of modifications // Software Maintenance, 1994. Proceedings., International Conference on – 1994. – С. 243-252.

АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ ПРИ ПОМОЩИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

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

Попитич А. Я.

Серебряная Л. В. – к-т. техн. наук, доцент

В ходе решения задач, где необходимо использование нейронной сети, всегда присутствует проблема выбора конкретной архитектуры сети. Эта проблема в большинстве случаев решается простым подбором и подгоном параметров по причине разнообразия анализируемых данных и уникальности решаемых задач. В данном докладе будет предложено применение генетического алгоритма, который позволяет автоматизировать процесс и сгенерировать сеть, которая будет соответствовать анализируемым данным и решаемой задаче.

Основная задача была поставлена следующим образом – обеспечить построение топологии нейронной сети автоматическим образом. В качестве исходных данных была взята обучающая выборка для нейронной сети, которая в дальнейшем будет применяться для обучения каждого, генерируемого генетическим алгоритмом, экземпляра нейронной сети. Все эксперименты для простоты были организованы для решения небольшой задачи XOR, результаты которой могут быть впоследствии применены и для решения более масштабных задач.

Для применения генетического алгоритма в ходе решения поставленной задачи было использовано прямое кодирование связей нейронной сети. На рисунке 1 представлен пример небольшой нейронной сети и кодирования межнейронных связей.

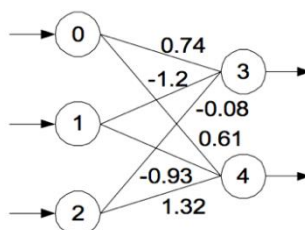


Рис. 1 – Пример топологии нейронной сети

Результат кодирования – геном, который представлен в таблице 1, где каждый ген в геноме содержит информацию об связующих нейронах и значения веса связи.

0	3	1	3	2	3	0	4	1	4	2	4
0.74		-1.2		-0.08		0.61		-0.93		1.32	

При этом были введены следующие правила при формировании новых и удалении имеющихся нейронов в ходе работы генетического алгоритма:

- все нейроны имеют определенный индекс. Удаление нейронов входного или выходного слоев не допускается;
- новые нейроны получают минимальный из возможных номеров;
- при удалении нейрона индексы остальных нейронов корректируются с учетом удаленного.

Оператор скрещивания для генетического алгоритма использует две особи (генома нейронных сетей) и в результате получает два потомка. Общие нейроны и связи наследуются между потомками, а значения связи в сетях потомков формируются при помощи двухточечного кроссовера. Различающиеся элементы разыгрываются между двумя потомками. На рисунке 2 представлен пример процесса скрещивания.

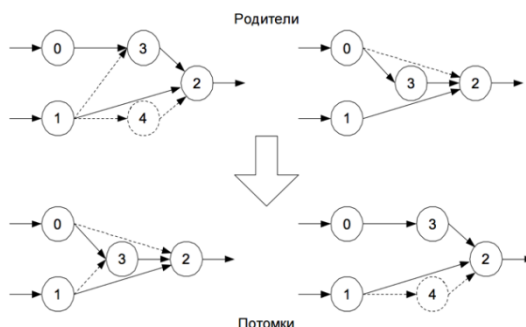


Рис. 2 – Пример процесса скрещивания двух топологий

Оператор мутации представляет собой композицию из нескольких операторов мутации: добавление нейрона в скрытый слой, удаление случайного нейрона вместе с его связями, добавление связей между нейронами, удаление связи между нейронами и изменение веса случайно выбранной связи.

Основной проблемой в случае применения мутации являлось бесконтрольное добавление и удаление связей, что в большинстве случаев приводило к нарастанию большого числа связей или наоборот – “обеднению” связей между нейронами. Для преодоления такого негативного эффекта были использованы специальные метрики, которые оценивали текущее состояние топологии и помогали алгоритму выбрать нужный оператор мутации.

Первый коэффициент – коэффициент связности нейронов. Чем больше данный коэффициент, тем более насыщенной связями является топология. Данный коэффициент рассчитывается формулой 1:

$$f_c = \frac{N_c}{2^{FB-1} [N_n(N_n - 1) - N_i(N_i - 1) - (1 - FB)N_o(N_o - 1)]} \quad (1)$$

где N_c – количество связей в сети, N_i, N_o, N_n – количество входных, выходных и общее количество нейронов в сети

Второй коэффициент f_n вычисляется на предположении, чем больше нейронов присутствует на входных и выходных слоях в сумме, тем более сложная сеть (с большим количеством нейронов) необходима сеть (формула 2):

$$f_n = (N_i + N_o) / N_n \quad (2)$$

Чем больше нейронов в сети, тем меньше будет коэффициент и тем меньше вероятность выбора мутации по добавлению нового нейрона.

Полученные в результате скрещивания и мутации потомки оцениваются при помощи фитнес-функции. В качестве фитнес функции взята формула вычисления среднеквадратичной ошибки в ходе обучения сети на обучаемых данных.

По результатом данной функции можно определить именно те нейронные сети, которые наиболее приспособлены к данным для обучения.

Для тестирования разработанного алгоритма была решена задача XOR, которая является частным случаем n-битной задачи четности, когда для последовательности из n-бит необходимо определить четность единичных разрядов (1 – нечетно, 0 - четно). Изначально все особи представляли собой одинаковые сети с входным и выходным слоем, без нейронов промежуточного слоя.

В результате применения генетического алгоритма была получена следующая топология сети (рисунок 3).

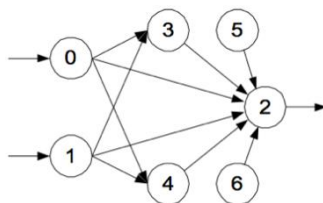


Рис. 3 – Пример автоматически сгенерированной топологии сети

Минимальная сеть, которая способна решать поставленную задачу, состоит из 4-х нейронов и 5-и связей, однако стоит отметить, что полученная сеть была сгенерирована автоматическим образом, что является неплохим результатом

Для дальнейшего улучшения результатов автоматического построения можно провести дополнительные эксперименты по выбору способа кодирования топологии сети. Для этого можно применить косвенный способ кодирования, где строятся определенные грамматики топологических структур или кодируются отдельные блоки.

Список использованных источников:

1. N.J. Radcliffe, Genetic Neural Networks on MIMD computers / Radcliffe N.J., Davis L // University of Edinburgh, Edinburgh, England, 1990 – 65p.
2. K. O. Stanley, Evolving Neural Topologies through Augmenting Topologies / Stanley K. O., Miikkulainen R. // Evolutionary Computation, The MIT Press, 2002. – № 10(2). – p. 99-127.
3. J. R. Koza, Genetic generation of both the weight and architecture for a neural network / Koza J. R., Rice J. P. // International Joint Conference on Neural Networks. 1991. – Vol. 2, P. 397–404.

ЭВОЛЮЦИОННЫЙ АЛГОРИТМ ГЕНЕРАЦИИ ТЕСТОВЫХ СЦЕНАРИЕВ JSON ВЕБ-СЕРВИСОВ

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

Проведенцев Е. С.

Скобцов В. Ю. – кандидат технических наук

В практике современных информационных технологий и программной инженерии широкое распространение получили веб-сервисы, принимающие в качестве параметров данные в JSON формате. Запросы и ответы подобных сервисов могут быть описаны с помощью JSON-schema.

Для функционального тестирования JSON-сервисов зачастую применяется ручная разработка тестовых сценариев. Автоматизация разработки тестового сценария для JSON-сервиса осложнена, большим количеством входных и выходных параметров, а также возможными зависимостями логики срабатывания одного параметра от другого [1]. Для решения трудно формализуемых задач большой размерности широкое применение получили эволюционные алгоритмы, являющиеся одним из перспективных направлений исследований в области интеллектуальных систем, задач поиска, интеллектуальной обработки данных [2]. В частности, эволюционные алгоритмы эффективно используются в области автоматизации тестирования программного обеспечения [3]. Поэтому в данном исследовании предлагается эволюционный алгоритм генерации тестовых сценариев для подобных веб-сервисов.

В данной работе предлагается алгоритм, использующий JSON-schema для автоматизированного создания тестового сценария JSON-сервиса. Целью данного алгоритма является генерация тестового сценария, обнаруживающего максимальное количество ошибок и, в то же время, содержащего минимальное количество тестовых случаев (test case). В основе данного алгоритма лежит представление тестового случая в качестве экземпляра, каждое поле запроса, которого представляет собой отдельный ген.

Алгоритм работает следующим образом:

1. Генерируется множество-популяция случайных запросов, заполненных в соответствии с предоставленной JSON-схемой. Наполнение запроса должно быть максимальным, то есть каждому полю, по возможности, должно быть присвоено значение. В зависимости от типа поля выбирается стратегия заполнения. Для числовых типов могут выбираться граничные значения, для строковых – значения из предопределенного списка, который составляется в зависимости от специфики программного кода сервиса. Цель данного запроса – задействовать как можно больше ветвей кода, повысив вероятность выявления ошибки. В случае, если данный пункт выполняется не первый раз, то необходимо убедиться, что запрос не

совпадает с уже выполненными на предыдущих итерациях.

2. Сгенерированные запросы отправляются сервису на выполнение.

3. Ответ от сервиса проверяется на соответствие JSON-схеме.

4. Если ответ соответствует JSON-схеме, значит ошибка не была выявлена, и необходимо произвести мутацию запроса. Мутация может выполняться как внутри поля, так и над структурой всего запроса целиком. Возврат к пункту 2.

5. Если ответ не соответствует JSON-схеме, считается, что была обнаружена ошибка. Ошибочный ответ запоминается для дальнейшей обработки.

6. Запускается алгоритм поиска гена, который вызывает ошибку. Для этого из запроса по очереди удаляются все гены (поля) и отправляются на выполнение. Если после удаления очередного гена запрос выполнен без ошибки, то удаленный ген либо сам, либо в комплексе с другими генами является причиной ошибки, производится мутация до тех пор, пока не будет выявлен набор генов, являющийся причиной ошибки. Данный набор добавляется в массив для дальнейшей обработки. Возврат к пункту 1.

В качестве критерия останова цикла 1 – 6 могут выступать временные рамки либо достижение определенного количества мутаций запроса, не приведших к ошибке. После цикла 1 – 6 из наборов генов, полученных в 6 пункте, генерируется минимальное число запросов, соответствующих JSON-схеме. Полученный набор тестовых случаев является результатом работы данного алгоритма.

Подобный алгоритм может быть полезен при генерации smoke-тестов, так как при поиске ошибок задействуется максимально возможное число параметров, следовательно, покрытие функционала тестами стремится к 100%. Значительную пользу может принести набор генов, вычисленный в 6 пункте, являющийся причиной ошибки. Программисту будет значительно проще локализовать ошибку в коде, имея запрос состоящий только из тех полей, которые необходимы для воспроизведения ошибки.

Данный алгоритм может быть использован для регрессионного тестирования: тестирование можно считать успешным при безошибочном выполнении изначального набора запросов и последующих, мутировавших определенное число раз.

Алгоритм можно рассматривать в контексте стратегии мутации при генерации более сложных тестовых сценариев (например, CRUD сценарий) с помощью эволюционных алгоритмов.

Список использованных источников:

1. Тестирование программного обеспечения / Сэм Канер, Джек Фолк, Енг Кек Нгуен / ДиаСофт – Москва, 2001 – 544 с.
2. Ю.А.Скобцов, Д.В.Сперанский. Эволюционные вычисления. Москва: ИНТУИТ, 2014.
3. T. Mantere, J.T. Alander Evolutionary software engineering, a review // Applied Soft Computing 5 (2005) pp.315–331.

АНАЛИЗ ЭФФЕКТИВНОСТИ РАБОТЫ СИСТЕМЫ РАСПРЕДЕЛЕННОЙ ПЕРЕДАЧИ ДАННЫХ

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

Разумов Е.В.

Иванюк А.А. – д-р. техн. наук

Одним из возможных методов, по средствам которого можно достичь большую устойчивость к MITM-атакам, является метод построения системы распределенной передачи данных по средствам peer-to-peer сети. Основная идея, заложенная в данную систему, заключается в разбиении передаваемой информации на блоки, шифровании каждого из этих блоков и последующей передаче их через отдельный промежуточный узел.

Оценивать эффективность работы такой системы целесообразно по нескольким позициям: скорости работы системы в целом, криптостойкости такого способа пересылки данных, зависимости от внешних факторов, преимуществах перед существующими решениями.

С точки зрения скорости, передача данных между адресатами будет значительно медленнее, чем в системах, которые передают данные от отправителя к получателю напрямую. Зависимость времени передачи данных между адресатами от количества пройденных промежуточных узлов не является линейной. Это объясняется в первую очередь тем, что в реальной системе кроме затрат времени, которые необходимы непосредственно для операций шифрования/расшифровывания, необходимо еще время на передачу блоков зашифрованных данных от одного узла к другому. И это время не является постоянной величиной и зависит от качества сети.

Следующей позицией, по которой можно оценить эффективность разработанной системы, является скорость передачи данных между адресатами в зависимости от числа разбиений исходных данных при условии, что данные передаются через один промежуточный узел. Анализируя результаты, полученные практическим путем, можно сделать вывод, что с увеличением числа разбиений скорость работы системы в целом уменьшается. Причиной тому являются следующие факторы:

- скорость работы сети на каждом из узлов;

- производительность промежуточного узла;
- загруженность промежуточного узла;
- дополнительные затраты времени отправителем на разбиение исходных данных на блоки, шифрование каждого из блоков, установление сессий с каждым из промежуточных узлов;
- затраты времени конечным адресатом на склейку пришедших блоков.

Следовательно в случаях, когда необходимо увеличить криптостойкость системы к MITM-атакам, целесообразно делить исходные данные на большее число пусть и небольших блоков. В противном случае число разбиений исходных данных должно быть в пределе 2-4, что позволит избежать сильного замедления работы системы.

Еще одной немаловажной характеристикой данной системы является производительность каждого из узлов в зависимости от размера передаваемого через него блока. Эта зависимость также не является линейной. Узел способен обработать большее количество байтов информации в единицу времени, если она передается большими блоками. Это связано с тем, что для обработки каждого из блоков информации необходимо устанавливать защищенную сессию с другим узлом сети и инициализировать криптосервисы для обработки полученного блока.

Дополнительным параметром при оценке эффективности разработанной системы является также величина затраченных ресурсов для передачи данных по средствам peer-to-peer сети. Узлы данной сети могут быть использованы не только для ретрансляции зашифрованных блоков данных между адресатами, но и для дополнительных задач, которые решаются при помощи построения peer-to-peer сети. Таким образом нельзя сказать, что зависимость затрат электроэнергии (или интернет-трафика) прямопропорциональна числу одновременно работающих узлов сети. Узел такой сети изредка будет использоваться для ретрансляции (или же передачи/получения) блоков данных, а в оставшееся время будет вести себя как будто это приложение и не было установлено.

На основании проведенных исследований установлено, что разработанную систему целесообразно применять в случаях, когда ключевым параметром является безопасность передачи данных, а скорость передачи при этом не имеет такого весомого значения, так как она всегда будет заметно ниже, чем у классических методов.

Список использованных источников:

1. Trappe, Wade (2005). Introduction to Cryptography with Coding Theory. New York: Pearson. p. 257.
2. Network Forensic Analysis of SSL MITM Attacks – NETRESEC Network Security Blog. Retrieved March 27, 2011.

МОДЕЛИ КАЧЕСТВА КЛИЕНТСКОЙ И СЕРВЕРНОЙ ЧАСТЕЙ ВЕБ-ПРИЛОЖЕНИЙ

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

Резванов А. В.

Бахтизин В. В. – к-т. техн. наук, профессор

В настоящее время вопрос оценки качества веб-приложений является актуальным, т.к. в современной экономике веб-приложение может выступать в качестве дополнительного канала связи между промышленными предприятиями, организациями и их клиентами (B2C) или для связи между предприятиями (B2B).

При оценке качества веб-приложения его можно разделить на клиентскую часть и серверную часть. Клиентская часть включает файлы гипертекстовой разметки HTML, стили CSS и код JavaScript. Серверная часть включает код на таких языках программирования, как Java, C#, PHP, Python и других.

Оценка качества веб-приложений может быть основана на международных стандартах качества, таких как стандарт ISO/IEC 25010:2011 [1]. В данном стандарте 8 характеристик качества. При этом некоторое подмножество этих характеристик является более важным для клиентской части, в то время как другое подмножество более характерно для серверной части.

Для клиентской части важными характеристиками являются функциональная пригодность, совместимость, удобство использования, надежность и сопровождаемость (рис. 1).

Для серверной части важными являются функциональная пригодность, эффективность функционирования, совместимость, надежность, защищенность, сопровождаемость и мобильность (рис. 2).

Жирным шрифтом выделены наиболее важные характеристики.

Надежность клиентской части включает подхарактеристику завершенности, в то время как для серверной части кроме завершенности имеют значение такие подхарактеристики как готовность, отказоустойчивость и восстанавливаемость.

Характеристика удобства использования клиентской части включает такие подхарактеристики, как

обучаемость (learnability), простота использования (operability) и эстетичность пользовательского интерфейса (user interface aesthetics). Кроме этого могут быть предложены дополнительные подхарактеристики удобства использования клиентской части, такие как обратная связь (feedback), сходство с аналогами (familiarity) и интерактивность (interactivity), которые учитывают специфику веб-приложений. Модель клиентской части не включает подхарактеристику доступность (accessibility) характеристики удобство использования, так как она определяется контекстом использования веб-приложения.

При анализе качества веб-приложений, помимо клиентской и серверной части, можно также оценивать качество информационного наполнения (контента) и информационной архитектуры веб-приложения, для которых также возможно создание собственных моделей качества.

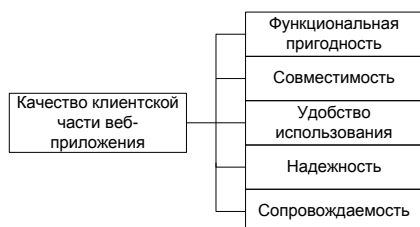


Рис. 1 – Модель качества клиентской части веб-приложения



Рис. 2 – Модель качества серверной части веб-приложения

Два рассмотренных подмножества характеристик качества позволяют более полно учитывать специфику как клиентской, так и серверной части веб-приложений.

Список использованных источников:

1. ISO/IEC 25010:2011: System and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models, ISO Copyright Office, Geneva, March 2011
2. В. В. Бахтизин, Л. А. Глухова, С. Н. Неборский. Метрология, стандартизация и сертификация в информационных технологиях. Мн.: БГУИР, 2013.

ОБРАБОТКА ДАННЫХ СЕНСОРНЫХ УСТРОЙСТВ МОБИЛЬНЫХ ПЛАТФОРМ

Белорусский государственный университет информатики и радиоэлектроники

Селиванов И. А.

Бранцевич П. Ю. – к-т техн. наук, доцент

В докладе рассмотрены подходы к разработке методов и алгоритмов обработки данных, полученных с сенсорных устройств. Отмечается необходимость учитывать следующие параметры таких устройств: пропускная способность канала передачи данных, тактовая частота микроконтроллера, объём оперативной памяти. Данные параметры накладывают определённые ограничения на разрабатываемые алгоритмы. Также в алгоритмах нужно реализовать фильтрацию данных.

Сенсорные данные и получаемая с их помощью информация активно используются в различных направлениях жизнедеятельности человека: от промышленности и научных приборов до устройств повседневного использования. Широкое распространение на данный момент получили мобильные устройства с набором датчиков, предназначенных для получения информации об активности и жизнедеятельности человека - так называемые фитнес-трекеры и им подобные устройства.

Основными функциями таких устройств является: вычисление количества сделанного пользователем шагов; определение пройденного расстояния, подсчёт пульса. Более сложные устройства специализируются на вычислениях конкретных параметров человеческой активности в контексте его деятельности. Такими параметрами могут быть, например, фазы сна или скорость во время бега, и т.д. Некоторые устройства предназначены для спортсменов, они вычисляют параметры конкретных активностей спортсмена во время игр или тренировок – это могут быть удары в боксе или теннисе, прыжки и повороты в горнолыжном спорте.

Одним из таких устройств является сенсор PIQ[1], который позиционируется как спортивный трекер. Данное устройство содержит в себе два акселерометра, гироскоп, магнитометр, датчики температуры и давления. Все данные, получаемые с этих датчиков, могут передаваться на другие мобильные устройства

через протокол Bluetooth Low Energy с частотой дискретизации до 500 Гц, однако наиболее часто используемыми являются 100 и 250 Гц. На устройства PIQ наложены определённые ограничения по производительности, размеру аккумуляторов, низкому энергопотреблению, габаритам.

Так как пропускная способность протокола BLE довольно низкая, потеря данных при частоте даже 100 Гц во время их передачи на другие, более мощные устройства, достаточно вероятно, соответственно, качество обработки из-за потерь данных снижается. Поэтому необходимы алгоритмы, способные выполняться на самих сенсорных устройствах. Как следствие, устройство должно передавать уже обработанные данные, объём которых значительно меньше. Все алгоритмы разрабатываются с учётом данного ограничения.

Разработка методов и алгоритмов обработки данных инерциальных сенсоров проводится для устройства PIQ. Эти методы и алгоритмы должны преобразовывать и интерпретировать сенсорные данные в корректную и полезную для пользователей информацию. Разрабатываемые алгоритмы обработки данных включают в себя: определение промежутков данных, в период которых происходила какая-либо активность; вычисление кинематики движения и других параметров, интересных для пользователя-спортсмена; машинное обучение и распознавание движений на основе сенсорных данных.

Режим обработки данных при непосредственном их получении от датчиков необходим в связи с малым объёмом оперативной памяти и невозможностью хранить в ней большие массивы данных. Алгоритмы, обрабатывающие большие массивы данных и способные повторно использовать для вычислений уже полученные ранее сенсорные данные, проще и эффективнее. Однако эти алгоритмы не смогут быть применены на исследуемых сенсорных устройствах. Проблема потоковой обработки данных решается с помощью использования буфера данных минимальных размеров (например, для алгоритмов сглаживания достаточно буфера данных размером 5-11 точек), а также путём применения потоковых формул для подсчёта статистических данных (к примеру, таких, как среднее арифметическое, среднее квадратичное отклонение и дисперсия). Выявление активностей во время получения данных производится с использованием конечного автомата с состояниями, которые определяют начало, конец активности и некоторые другие состояния.

Тактовая частота микроконтроллеров сенсорных устройств довольно низкая. Поэтому необходимо разрабатывать достаточно простые по вычислениям методы и алгоритмы обработки данных, чтобы скорость работы алгоритма не уменьшалась из-за меньшей тактовой частоты микроконтроллера по сравнению с более мощными компьютерами. Однако при этом необходимо учитывать и качество обработки.

Фильтрация сенсорных данных, избавление от постоянной составляющей и внешних искажающих факторов в данных остаётся открытым вопросом и одним из важных пунктов в исследовании. На данный момент один этап фильтрации производится на сенсорах – применяется аппаратный фильтр низких частот. Также осуществляется калибровка датчиков, она основана на постоянной проверке состояния сенсора и в момент, когда сенсор не двигается, вычисляются постоянные составляющие данных и удаляются. Однако, как показывают результаты тестов, этого недостаточно для корректных вычислений кинематики движения и других пользовательских параметров.

Таким образом, при разработке алгоритмов обработки сенсорных данных необходимо предложить их простую реализацию, способную выполняться на сенсорных устройствах, с обеспечением режима потоковой обработки данных и достаточной точности предоставляемых потребителю результатов.

Список использованных источников:

1. PIQ – Homepage [Электронный ресурс]. – 2016. – Режим доступа: www.piq.com

АРХИТЕКТУРА ДЕЦЕНТРАЛИЗОВАННОЙ РАСПРЕДЕЛЁННОЙ ПОДСИСТЕМЫ ХРАНЕНИЯ ДАННЫХ В СИСТЕМАХ ПРИНЯТИЯ РЕШЕНИЙ

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

Скачихин А.А.

Бранцевич П.Ю. – к.т.н., доцент

Увеличение объёмов данных и сложности алгоритмов в системах принятия решений неизбежно ведут к распределённому хранению и обработке, в силу того, что ограниченные возможности одного компьютера не позволяют решать поставленные задачи за приемлемое время. Децентрализованная распределённая система хранения информации, учитывающая особенности предметной области, может дать ряд преимуществ по сравнению с реляционными базами данных, а также не специализированными распределёнными системами хранения, архитектурные решения которых направлены на максимальную универсальность.

Рассмотрим сценарий использования системы принятия решений на примере системы оценки технического состояния механизмов на основе вибрационных параметров. Данные с виброизмерительных датчиков падают в подсистему хранения и сбора информации, где, в зависимости от поставленных задач,

могут использоваться для принятия решений как в реальном времени, так и для обработки и статистического анализа в дальнейшем. Длительное хранение представляет наибольший интерес в рамках этой работы. На основе собранных данных подсистемы обработки данных выводят новые знания, которые в дальнейшем используются для интеллектуального анализа. При этом стоит отметить, что обработка данных возможна без их изменения. Второй важный аспект работы таких систем – отсутствие конкуренция на запись, ведь каждый датчик является отдельным источником данных. Третье, данные хранятся в неструктурированном виде, как последовательность измерений, что даёт возможность эффективно хранить их в бинарном виде.

Современные универсальные системы хранения информации, оказываются неоправданно сложными для задач хранения вибрационных сигналов. Неиспользуемыми остаются такие возможности как транзакционность, возможность задать строгую схему, механизмы блокировок при конкурентной записи. В тоже время универсальность существующих решений не позволяет применять эффективные методы сжатия, такие как дельта компрессия, а методы доступа к данным не могут быть оптимизированными учитывая особенности алгоритмов, которые в последствии будут обрабатывать полученные данные.

Вышеизложенные факты лежат в основе гипотезы, что, применив специализированную децентрализованную распределённую подсистему хранения данных, можно добиться повышения эффективности работы системы принятия решений. А именно, обеспечить:

- 1) уменьшение времени доступа благодаря эффективному двоичному хранению на диске блоками, соответствующими размеру окна алгоритмов;
- 2) уменьшение времени доступа благодаря отказу от транзакций и блокировок на операции чтения;
- 3) уменьшение объемов занимаемого пространства на диске за счет применения методов сжатия однородных двоичных данных;
- 4) отказоустойчивость системы, гарантируемую путем репликации данных, на основе конфигурируемого признака (например, времени поступления данных);
- 5) децентрализацию, которая ведёт к тому, что запись в систему хранения может происходить полностью параллельно без участия управляющей стороны [1].

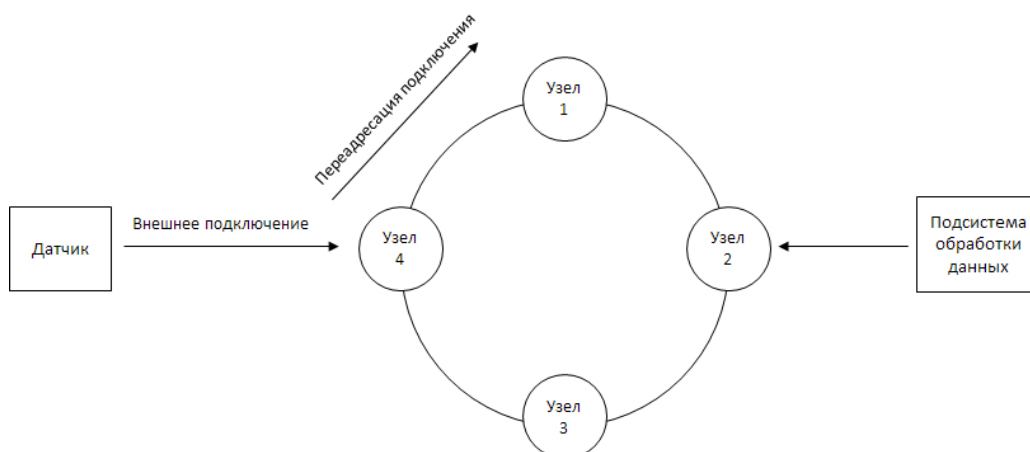


Рис. 1 – Упрощенная архитектура децентрализованной распределённой подсистемы хранения

Выделим следующие ключевые особенности архитектуры (рис.1):

- 1) количество узлов в подсистеме не является фиксированным и может быть изменено в соответствии с решаемой задачей;
- 2) так как подсистема является децентрализованной, датчики и подсистема обработки данных могут подключаться к любому узлу системы. Если необходимые данные не найдены на узле, то узел связывается с узлом, на котором данные располагаются. В случае с датчиками узел может принять решение переадресовать подключение на другой узел чтобы устранить необходимость передачи данных по сети. Подключение подсистемы обработки данных никогда не переадресовывается, так как в большинстве сценариев обращения на чтение требуют доступа к данным на нескольких узлах;
- 3) при старте подсистемы узлы обмениваются информацией о том, по какому принципу происходит репликация и сегментирование данных;
- 4) узлы, по возможности, кэшируют данные в оперативной памяти для ускорения доступа;
- 5) произвольный доступ к данным не возможен, это позволяет использовать дельта компрессию и существенно экономить дисковое пространство.

У такой реализации также есть ряд недостатков по сравнению с более универсальными аналогами:

- 1) узкая специализация может привести к тому, что придётся комбинировать данные из различных источников в промежуточное представление, что окажет существенное влияние на время обработки данных [2];
- 2) нет возможности вносить изменения в существующие данные в результате их обработки, что в ряде случаев будет предпочтительнее, чем хранение результатов отдельно, потребляя дополнительное дисковое пространство. Однако, для большинства сценариев использования изменение исходных данных представляется менее предпочтительным.

Таким образом, предлагается архитектура подсистемы хранения в системах принятия решений, позволяющая улучшить количественные характеристики системы путём проектирования с учётом особенностей предметной области.

Список использованных источников:

1. Zaharia, M. An Architecture for Fast and General Data Processing on Large Clusters / M. Zaharia // PHD Dissertation. – 3 February 2014 / Berkeley, USA. – 2014. P. 3 – 35
2. Pu, Q. FairRide: Near-Optimal, Fair Cache Sharing / Q. Pu, H. Li, M. Zaharia, A. Ghodsi, I. Stoica // NSDI. – 16 March 2016 / Santa Clara, USA. – 2016. P. 2 - 3

ПОСТРОЕНИЕ АНАЛИТИЧЕСКОГО СЕРВИСА ДЛЯ АНАЛИЗА СПЕКТРОВ

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

Усиков А. В.

Рудикова Л.В. – доцент, канд. физ-мат. наук

На сегодняшний момент накоплено довольно много материалов, которые представлены в определенных таблицах и атласах спектральных элементов. Непосредственная автоматизация процесса обработки спектрограмм, получаемых с помощью мобильного лазерного спектрометра, представляет собой программные инструменты, которые включают в себя достаточно узкий комплект возможностей. Недоступность инструмента, который дает возможность хранить и анализировать спектры в режиме реального времени с интеллектуальным поиском является актуальным направлением.

Установим необходимые требования к разрабатываемому сервису. В первую очередь в сервисе должны присутствовать возможности аналитической обработки данных, интеллектуального поиска данных (алгоритмы добычи данных), использования нейронных сетей для прогнозирования, корреляций, типовых образцов и исключений в больших объемах данных спектров. Необходимо отметить, что каждая из этих возможностей представляет собой самостоятельные части с точки зрения архитектуры.

Сервис должен равным образом предоставлять удаленный доступ к проанализированным и результирующим данным, поэтому необходимо предоставить публичную конечную точку пользователям для доступа в любое время к сервису и получения данных анализа.

Многоуровневая архитектура обеспечивает группировку связанной функциональности приложения в различных слоях, выстраиваемых вертикально, поверх друг друга. Слои слабо связаны, и между ними осуществляется явный обмен данными. Точное разбиение приложения на слои помогает поддерживать строгое распределение функциональности, что в свою очередь, обеспечивает гибкость, а также практичность и несложность сопровождения.

Функциональные области приложения разделяются на многослойные группы (уровни). Сервис состоит из шести взаимодействующих друг с другом слоев: уровень хранения данных, уровень надстроек, уровень доступа к данным, уровень бизнес-логики, уровень сервисов, уровень клиентов. Исходя из предъявляемых требований, для анализа спектров, полученных в результате лазерной экспрессной экспертизы будет приемлемым разрабатывать с применением многоуровневой архитектуры.

Для построения сервиса представлена многоуровневая архитектура. Подобная архитектура обеспечивает группировку связанной функциональности приложения в различных слоях, выстраиваемых вертикально, поверх друг друга. Слои слабо связаны, и между ними осуществляется явный обмен данными. Точное разбиение приложения на слои помогает поддерживать строгое распределение функциональности, что в свою очередь, обеспечивает гибкость, а также практичность и несложность сопровождения.

Каждый слой агрегирует ответственности и абстракции уровня, расположенного непосредственно под ним. При строгом разделении на слои составляющие одного слоя могут взаимодействовать только с составляющими такого же слоя или составляющими слоя, расположенного прямо под данным слоем. Более свободное разделение на слои позволяет составляющим взаимодействовать с составляющими того же и всех нижележащих слоев.

Внедрение подобного рода возможностей для обработки, анализа и хранения спектров, полученных в результате лазерной экспрессной экспертизы, дадут возможность внушительно усовершенствовать скорость аналитической обработки данных с интеллектуальным поиском, использованием нейронных сетей для прогнозирования сходств, корреляций, типовых образцов и исключений в больших объемах данных спектров. Следует отметить, что благодаря версионному хранению спектров, данные снимков каждый раз можно реконструировать до конкретного состояния в зависимости от требований..

Список использованных источников:

1. Кудрявцев, Ю. А. OLAP технологии: обзор решаемых задач и исследований / Ю.А. Кудрявцев // Бизнес-информатика. - 2008. - № 1. - С. 66-70.
2. Рудикова, Л. В. Разработка программного визуализатора спектров для поддержки лазерной экспрессной экспертизы //

АНАЛИЗ ЗАЩИЩЕННОСТИ WEB-ПРИЛОЖЕНИЙ

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

Халецкий С.Д.

Глухова Л. А. – канд. техн. наук, доцент

В докладе обоснована значимость анализа защищенности web-приложений, а также рассмотрены методы повышения защищенности путем своевременного обнаружения и устранения недостаточно защищенных web-приложений.

Защищенность – степень защиты программным продуктом или системой информации и данных так, чтобы люди, другие продукты или системы имели степень доступа к данным, соответствующую типам и уровням их авторизации.[1]

К подхарактеристикам защищенности относят: конфиденциальность, целостность, непроверяемость, идентифицируемость, аутентичность.

Зачастую современные web-приложения имеют дело с конфиденциальной информацией, которая в свою очередь доступна посредством Web. При этом обмен информацией между браузером и сервером происходит по открытым каналам с использованием открытых протоколов. В связи с этим контролировать передаваемые данные сложно. Поэтому важное значение имеют вопросы обеспечения защищенности web-приложений.

Open Web Application Security Project (OWASP) – это библиотека, содержащая исчерпывающее руководство по поиску различного рода уязвимостей, а также содержит рекомендации к процессу проведения анализа защищенности web-приложений.

Наиболее простым и, как следствие, более распространенным способом анализа защищенности web-приложений является инструментальное обследование. Данный метод предусматривает использование сканеров безопасности, а также дополнительных инструментов, автоматизирующих некоторые сценарии эксплуатации и выявления уязвимостей. Одним из основных недостатков данного подхода является то, что необходимо постоянно поддерживать сигнатуры в актуальном состоянии, а также для получения корректной оценки работы необходимо осуществлять проверку транзакционно, то есть на ряду с проверкой результатов последнего вызова, проверять результаты предыдущих, что является весьма непросто.

Использование инструментального анализа не всегда является возможным, например в банковской сфере. Важность выявления уязвимостей приводит к тому, что их поиск необходимо осуществлять также и вручную, хотя это и требует больших временных затрат, а также не исключает проявление “человеческого фактора”.

Если исходный код используемых приложений, сервисов, библиотек является открытым, достаточно действенным методом по обнаружению уязвимостей будет анализ исходного кода. Если не требуется проверка воспроизведения найденных уязвимостей, то анализ можно будет проводить вообще не затрагивая работу самого web-приложения. Наибольшее распространение среди методов по анализу исходного кода получил метод статического анализа, основанный на использовании сигнатур, базисом которых являются регулярные выражения. Так как не все сигнатуры могут присутствовать, то некоторые из уязвимостей будут не выявлены. Поэтому совместно со статическим анализатором кода следует применять динамические анализаторы, которые на низком уровне разбирают синтаксис языка программирования web-приложения, после проверок которого выявляются грубые ошибки, допущенные разработчиками. Стоит учитывать, что, наряду со сканерами, анализаторам присущи те же недостатки.

Организацию процесса анализа защищенности следует начинать прежде всего с постановки цели самого анализа, определения области исследования, после чего сформировать перечень производимых проверок. В зависимости от цели анализа выбирается стратегия. Если необходимо выявить возможности проникновения, нарушение штатного режима, то приложение следует рассматривать как “черный ящик”, проведение работ будет осуществляться без предварительного получения какой-либо информации о нем. Если достаточных средств для проведения анализа нет, а цель – повышение уровня защищенности web-приложения, то его стоит рассматривать как “серый ящик” (например, когда злоумышленнику известна структура каталогов, исходный код некоторых файлов, функций) с использованием инструментального подхода к его анализу, а также использовать ручные проверки в наиболее критических местах.

Так как web-приложения ориентированы на массовое использование, то сбои в работе, вызванные действиями злоумышленника, оказывают сугубо негативные последствия. Этому также способствует возможность использования однотипных сценариев эксплуатации уязвимостей, так как зачастую используются и шаблонные решения, а обновление до актуальной безопасной версии не осуществляется. Поэтому анализ защищенности web-приложения должен быть частью общей стратегии обеспечения защищенности.

Список использованных источников:

1. ISO/IEC 25010:2011. Проектирование систем и разработка программного обеспечения. Требования к качеству систем и

программного обеспечения и их оценка (SQuaRE). Модели качества систем и программного обеспечения. – Введ. 2011-03-03. – Женева, 2011.

СИСТЕМА КРИТЕРИЕВ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПРОГРАММНЫХ СРЕДСТВ ОРГАНИЗАЦИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

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

Царевич Д.Ю.

Бахтизин В.В. – к.т.н., доцент

В связи с интенсивным развитием сфер человеческой деятельности, в том числе научной, экономической, медицинской, растут потребности общества в вычислительных ресурсах, необходимых для обработки информации. Одним из возможных способов удовлетворения данных потребностей является использование распределенных вычислений (РВ).

Для оценки и сравнения параметров программных средств организации РВ в работе формулируется система критериев оценки эффективности ПС организации РВ, включающая следующие критерии [1]:

1. Производительность – ключевой критерий, характеризуется отношением показателей производительности ПС организации РВ: максимального из полученных практически к максимально возможному теоретически.

2. Масштабируемость – способность ПС организации РВ выполнять работу, пропорциональную числу вычислительных узлов (ВУ) без потери производительности при росте числа ВУ.

3. Безопасность – описывает эффективность механизмов защиты информации, вычислительных узлов и процессов ПС организации РВ от несанкционированного доступа.

4. Отказоустойчивость – характеризуется соотношением промежутка времени, в течение которого ПС организации РВ находилось в состоянии работы к общему промежутку времени, в течение которого производилось измерение.

5. Прозрачность – характеризует способность ПС скрывать свою распределенную природу, т.е. распределение процессов и ресурсов по множеству ВУ, и способность представляться для пользователей и разработчиков приложений в виде единой централизованной компьютерной системы.

В качестве критерия оценки *производительности* ПС организации выбрано отношение значений двух используемых вариантов измерения производительности – максимальной и пиковой [2].

Пиковая производительность (P_{peak} , TFLOPS) – теоретический предел производительности (выражаемый через операции с плавающей точкой) для оцениваемого ПС, находится с помощью формулы (1.1):

$$P_{peak} = \sum_{i=1}^N (F_i * P_i * I_{tick} / C), \quad (1.1)$$

где F_i – тактовая частота процессора i ВУ, МГц.

P_i – число процессоров в i ВУ.

I_{tick} – количество инструкций с плавающей запятой на такт (4 – для процессоров Core2; 8 – для процессоров Intel с AVX).

C – константа, $C = 10^6$.

N – количество ВУ, используемых ПС.

Максимальная производительность – максимальная производительность ПС, достигаемая при решении практических задач.

Критерий оценки производительности P ПС организации РВ определяется как соотношение значений максимальной и пиковой производительностей по формуле (1.2):

$$P = P_{max} / P_{peak} * 100\%, \quad (1.2)$$

где P_{peak} – пиковая производительность, TFLOPS.

P_{max} – максимальная производительность, TFLOPS.

Рассматривая *масштабируемость* как способность оцениваемого ПС выполнять работу, пропорциональную числу вычислительных узлов (ВУ), критерий масштабируемости определяется с помощью критерия оценки параллельной эффективности [2].

Ускорение (англ. «speedup»), получаемое при использовании распределенного алгоритма (РА) для P ВУ, по сравнению с последовательным вариантом выполнения вычислений, определяется величиной K по формуле (1.3):

$$K = T_S / T_N, \quad (1.3)$$

где T_S – время, затраченное на решение задачи последовательным алгоритмом на одном ВУ

T_N – время, затраченное на решение той же задачи при использовании распределенного алгоритма, выполненного на N ВУ.

N – количество ВУ, используемых ПС.

Параллельная эффективность E определяется как отношение ускорения к количеству используемых ПС вычислительных узлов по формуле (1.4):

$$E = K / N * 100\%, \quad (1.4)$$

где K – ускорение, получаемое при использовании распределенного алгоритма

N – количество ВУ, используемых ПС.

В качестве основы для разработки критерия оценки *безопасности* ПС организации РВ выбран стандарт Department of Defense Trusted Computer System Evaluation Criteria, TCSEC, DoD 5200.28-STD [3].

В соответствии с вышеуказанным стандартом, критерии определения безопасности ПС делятся на 4 раздела: D, C, B и A, из которых наивысшей безопасностью обладает раздел A. Каждый раздел представляет собой значительные отличия в доверии индивидуальным пользователям или организациям. Разделы C, B и A иерархически разбиты на серии подразделов, называемые классами: C1, C2, B1, B2, B3 и A1. Каждый раздел и класс расширяет или дополняет требования, указанные в предшествующем разделе или классе.

Таким образом, критерий оценки безопасности S определяется как принадлежность ПС организации РВ к определенному классу вышеуказанного стандарта в соответствии с формулой (1.5):

$$S = S_{class}, \quad (1.5)$$

где S_{class} – класс безопасности по стандарту Department of Defense Trusted Computer System Evaluation Criteria, TCSEC, DoD 5200.28-STD [3].

Значение критерия оценки безопасности S постоянно для ПС, т.е. не зависит от конкретной решаемой вычислительной проблемы.

Критерий оценки *отказоустойчивости* R определяется как соотношение промежутка времени, в течение которого ПС организации РВ находилось в состоянии работы к общему промежутку времени, в течение которого производилось измерение и рассчитывается по формуле (1.6):

$$R = T_{ff} / T_{total} * 100\%, \quad (1.6)$$

где T_{ff} – промежуток времени, в течение которого ПС находилось в состоянии работы.

T_{total} – общий промежуток времени, в течение которого производилось измерение.

Для определения критерия оценки *прозрачности* ПС организации РВ из стандартов эталонной модели для распределенной обработки в открытых системах Reference Model for Open Distributed Processing (RM-ODP) выбраны семь наиболее важных типов прозрачности [4]: прозрачность доступа (англ. access transparency), прозрачность местоположения (англ. location transparency), прозрачность перемещения (англ. migration transparency), прозрачность смены местоположения (англ. relocation transparency), прозрачность репликации (англ. replication transparency), прозрачность одновременного доступа (англ. concurrency transparency), прозрачность отказов (англ. failure transparency).

Критерий прозрачности O определяется как соотношение количества поддерживаемых ПС организации РВ типов прозрачности к общему количеству наиболее важных типов прозрачности по формуле (1.7):

$$O = O_{supp} / O_{total} * 100\%, \quad (1.7)$$

где O_{supp} – количество поддерживаемых ПС типов прозрачности.

O_{total} – общее количество наиболее важных типов прозрачности, $O_{total} = 7$.

Значение критерия оценки прозрачности O постоянно для ПС, т.е. не зависит от конкретной решаемой вычислительной проблемы.

Предлагаемая система критериев оценки эффективности ПС организации РВ может применяться для анализа эффективности различных ПС организации РВ, а также выбора ПС с более высокими значениями приоритетных критериев для решаемой распределенной вычислительной задачи.

Список использованных источников:

1. Coulouris, G. Distributed Systems: Concepts and Design (5th Edition). Coulouris G. [and others]. – Boston: Addison-Wesley, 2011. – 1008 p.
2. Черемсинов, Д.И. Проектирование и анализ параллелизма в процессах и программах / Д.И. Черемсинов. – Минск: Беларус. навука, 2011. – 300 с.
3. 5200.28-STD Trusted Computer System Evaluation Criteria // National Security Institute [Электронный ресурс]. – 2015. – Mode of access: <http://csrc.nist.gov/publications/history/dod85.pdf>. – Date of access: 18.02.2016.
4. Farooqui, K. ISO Reference Model of Open Distributed Processing (RM-ODP) // Institute For Enterprise Architecture Developments [Электронный ресурс]. – 2015. – Режим доступа: <http://www.enterprise-architecture.info/Images/Document-s/RM-ODP2.pdf>. – Дата доступа: 15.01.2016.

СЕМАНТИЧЕСКАЯ НАВИГАЦИЯ В БОЛЬШИХ ИНФОРМАЦИОННЫХ МАССИВАХ

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

Чеушев К. В.

Глухова Л. А. – канд. техн. наук, доцент

Предложено системное решение для актуальной задачи навигации в больших информационных массивах, основанное на применении комплекса аналитических функций, создающего возможность целенаправленного движения пользователя в пространстве информационного ресурса, направляемого не априори установленными ссылками, а его текущими потребностями и интересами. Выделена обеспечивающая семантическую навигацию операциональная триада «структурирование – экстрагирование – поиск подобного», элементы которой собираются в цепи и каскады, реализующие все необходимые аналитические функции, такие как сложные высокоточные формы поиска, классификация, кластеризация, автоматическое выделение ключевых термов, построение сниппетов, аннотаций, дайджестов и т.д., позволяющих пользователю удовлетворить сложные информационные потребности, итерационно двигаясь с помощью аналитической системы к искомому результату.

Экспоненциальный рост доступных в сетях информационных ресурсов (ИР) создаёт заманчивую перспективу их использования в профессиональных, образовательных и других целях. Гигантские ИР сами по себе ещё не достаточное условие доступности и качества знаний, с ростом масштабов ИР увеличивается и усложняется проблема нахождения элементов ИР, удовлетворяющих конкретной информационной потребности пользователя (ИПП), критериям полноты, точности и качества. Острая ещё пару десятилетий назад проблема информационного голода трансформировалась в информационное перенасыщение, когда мы практически уверены в том, что в пределах нашей доступности нужная информация есть, но слабо представляем, как её получить, отобрать, сократить объём подлежащего «ручному» анализу до приемлемого. Раньше основные надежды возлагались на то, что информация перед помещением в состав ИР будет переработана человеком, должным образом структурирована и снабжена семантическими и другими метками, но сегодня всё чаще на это рассчитывать уже нельзя – объёмы новых ИР велики настолько, что качественно их обработать и всесторонне структурировать люди уже не в состоянии, но и в любом случае структурирование по одним критериям может не коррелировать с критериями конкретной ИПП. В текстовых ресурсах разметка для обеспечения гиперссылочной навигации уже перестала достаточно полно обеспечивать возможности доступа к искомой информации, а полнотекстовый поиск зачастую порождает огромный «шум», что вынуждает либо проводить очень ёмкую последующую обработку найденного, либо ограничиться использованием малого количества найденных документов, сомневаясь в их качестве и полноте удовлетворения ИПП. Велика и возрастает актуальность и востребованность так называемой семантической навигации – развитого способа ориентирования в огромных информационных ресурсах с опорой не на априори расставленные гиперссылки, а на интеллектуальные автоматические инструменты.

Под семантической навигацией понимается процесс движения аналитика или просто пользователя в информационном пространстве ИР, при котором он имеет возможность итерационно корректировать свои цели и интересы и переходить в соответствии с ними к другим подобластям ИР, стартуя с какой-то позиции, возможно семантически даже отдалённой от искомого результата (т.е. допускается нечёткое начальное представление цели). Отличие от навигации в классическом понимании (по априори установленным ссылкам) состоит в том, что направление движение в пространстве ИР не определено заранее, оно обеспечивается функциональными аналитическими механизмами непосредственно в момент взаимодействия пользователя с информационно-аналитической системой.

Исследования и эксперименты показывают, что для осуществления всех главных поисковых и аналитических функций необходимо и достаточно три функциональных (операциональных) механизма:

- структурирование дисперсной среды ИР;
- экстрагирование семантических детерминантов;
- собственно семантическая навигация как переход от одних элементов ИР к другим.

Эта тройка, будучи собранной в последовательности и в иерархию, обеспечивает все нужные смысловые операции разных уровней. При этом в каждую из них на более высоком уровне рассмотрения обычно входят две другие с предыдущего иерархического уровня. Так, операцию кластеризации как структурирование довольно высокого уровня можно выразить цепью операций экстрагирования и семантической навигации более низкого уровня.

Структурирование здесь – это выделение из дисперсной среды ресурса групп (кластеров) элементов, “откликающихся” на одну или несколько информационных потребностей. Это может быть поиск, и тогда результатом станет один кластер найденных текстов (а потребность сформулирована запросом), или параллельная группа поисков по разным основаниям, и тогда результат – группа кластеров. Если основания (так называемые профили классов) заданы априори, то это задача классификации, а если априори смысловые основания структурирования не заданы и их нужно определять в процессе, то это кластеризация. В отличие от часто понимаемых классификации и кластеризации как “раскладывания по полочкам” обязательно всех элементов ресурса здесь такая цель не ставится – в этом нет нужды, а нужно выделить из ИР только те группы элементов, которые потенциально удовлетворяют ИПП и только им. И второе – элемент ИР будет помещаться в такое количество кластеров, на какое у него будет достаточный “отклик”. Кластеризация в этой модели не только самостоятельная аналитическая задача, но и нередко решается как промежуточная в другом процессе. Например, построение кластеризованного дайджеста – по сути экстрагирование – содержит в себе кластеризацию как служебную задачу.

Экстрагирование семантических детерминантов – самый интеллектуально ёмкий элемент триады, во многом определяющий успех аналитических цепочек. Прежде всего это ранжирование элементов ресурса и выделение верхней части ранжированного списка, селекция тех элементов – семантических детерминантов, – которые наиболее полезны для получения планируемого результата, удовлетворения заданной информационной потребности. В более сложных случаях за подбором значимых элементов следует ещё и придание им структуры. Роль экстрагирования, как и других операций рассматриваемой триады, двойственна: оно может иметь самостоятельное значение (выделение ключевых термов, построение контекстных (так называемые сниппеты) и бесконтекстных аннотаций, дайджестов), а может входить в состав других операций. Так, кластеризация существенно основана на экстрагировании групп ключевых термов, которые могут составлять “смысловое ядро” кластера.

Семантическая навигация как элемент операциональной триады предполагает переход от одних элементов ИР к другим. И если структурирование и экстрагирование сужают рассматриваемое пространство ИР при решении аналитической задачи, то переход расширяет его. Ключевой конструктив этой операции – поиск (или отбор) подобного. Так, если взять текст, то смысловой переход это поиск текстов, подобных на заданный. Операция семантической навигации на уровне поиска подобных включает в себя пару других операциональных элементов: чтобы найти подобные тексты, система экстрагирует из “эталона” ключевые термы и выполняет по ним операцию структурирования (поиск, а затем, возможно, классификацию или кластеризацию найденного).

Навигация поиском подобных становится точнее, если взять не один текст, а кластер похожих текстов: тогда проще отобрать подходящие ключевые термы, отражающее как раз то, что в текстах кластера общее. Так формируется “экстракт кластера” в форме ранжированного списка ключевых термов. Полезна ещё одна операция – получение своего рода “экстракта кластера” уже в другой форме – в форме той части текстов кластера, которые сильнее других “откликаются” на выделенное множество ключевых термов, т.е. текстов, максимально релевантных тематике кластера и концентрированно отражающих эту тематику (тогда как у пользователя нигде не возникла необходимость сформулировать её в явном виде, и это одно из наиболее ярких достоинств подхода).

Определённая таким образом семантическая навигация может быть сколь угодно долгой, поскольку комплекс функциональных механизмов обеспечивает как сужение пространства анализа, так и его расширение. Сужение пространства анализа достигается применением того или иного способа структурирования (классификации, кластеризации) с последующим переходом к одному из кластеров. В таком кластере также возможно сужение (повторная кластеризация или поиск по словарю, запросу, фрагменту текста и т.д.) или выделение “экстракта” в виде наиболее выразительных текстов, а возможно и расширение путем поиска похожих текстов или поиска по ключевым словам или дайджестам, которые система предложит для этого кластера.

Семантическая навигация должна стать действенным механизмом решения поисково-аналитических задач, в том числе априори неизвестных. Действительно, если создать необходимый набор интеллектуальных инструментов – функций для всех элементов рассмотренной триады, собрать их в более высокоуровневые цепочки, связать их в единое целое с интерфейсными решениями, то можно получить очень мощный комплекс, работающий по принципу взаимодействия пользователя и аналитической системы достижению полезного результата, с положительной обратной связью, реализующий цепь “синергических транзакций” взаимодействия человек-машина. Начав с посылки даже самого малого смыслового сигнала, хоть как-то отражающего информационную потребность, пользователь получит отклик системы (например, кластер текстов со сниппетами, подсветкой автоматически выявленных ключевых термов, дайджестом), с которым он может провести действия, уточняющие потребность (отобрать наиболее релевантные и представительные

тексты или предложения, поднять веса некоторым термам и т.п.) и запустить следующую аналитическую цепь, которая даст уже более полный и точный результат – и так далее до достижения нужного качества и достаточной полноты решения поставленной задачи.

На основе предложенного и проверенного эксперимента комплекса решений планируется создание небольшой информационно-аналитической системы, на которой будут отработаны все предлагаемые решения. В качестве ИР по ряду мотиваций планируется использовать документальный ресурс Википедии. С одной стороны, этот ресурс доступен и понятен, с другой – реализованная система будет полезна на практике, реализуя ряд важных востребованных аналитических функций. И, наконец, разметка данного ресурса позволит в некоторых случаях применить автоматизированные приёмы анализа результатов работы компонентов системы, что даст возможность быстрее и качественнее подобрать параметры алгоритмов и метрик.

ХАРАКТЕРИСТИКИ КАЧЕСТВА СПЕЦИФИКАЦИЙ ТРЕБОВАНИЙ

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

Чиркова А. Ю.

Бахтизин В. В. – к. техн. наук, доцент

Качество программного обеспечения можно определить как соответствие явно установленным функциональным и эксплуатационным требованиям и явно указанным стандартам разработки. Такое определение качества программного обеспечения подчеркивает, что требования к программному обеспечению – основа, относительно которой определяется качество ПО.

Обеспечение качества программного обеспечения – непрерывный процесс в течение всего жизненного цикла ПО, который охватывает: методы и средства анализа, проектирования и кодирования; методику многоуровневого тестирования; контроль программной документации и внесенных в нее изменений; процедуры обеспечения соответствия стандартам в области разработки ПО, соответствие которым определено в задании на разработку данного ПО.

Разработка любого программного средства (ПС) начинается с определения требований, которые предъявляются к нему. Качество конечного продукта напрямую зависит от того, насколько качественно сформулированы требования [1]. По оценкам Д. Леффингуэлла и Д. Уидринга, стоимость исправления ошибок на финальных этапах разработки ПС возрастает на порядок [2]. Именно поэтому важно обеспечивать качество разрабатываемого продукта еще на этапе создания спецификации требований.

Обеспечение качества спецификации невозможно без его оценки, что в свою очередь не возможно без соответствующих методов оценки качества, основанных на модели качества. Современные стандарты, такие как, например, IEEE 830 [3], ISO/IEC/IEEE 29148 [4], ISO/IEC 25030 [5] содержат только рекомендации и методики составления спецификаций требований, а также характеристики качества требований. Однако они не описывают ни модели качества, ни метрики(меры), по которым можно было бы оценить качество требований.

Модель качества ПС в соответствии с ISO/IEC 25010 [6] представлена в иерархическом виде, где на первом уровне отображены характеристики (показатели) качества, а на втором уровне – подхарактеристики. Всего модель содержит восемь характеристик качества программного продукта: функциональное соответствие, эффективность функционирования, совместимость, практичность, надежность, защищенность, сопровождаемость, мобильность. Однако данная модель качества может быть применена только к программному продукту, или к компьютерной системе, которая включает в себя программное обеспечение, так как большинство подхарактеристик имеют отношение к ПС и системам. Поэтому модель качества спецификаций требований должна содержать характеристики и подхарактеристики качества, оценивающие не только на качество требований, но и влияющие на качество ПС в конечном итоге.

Качество данных является ключевым компонентом качества и полезности информации, полученной из этих данных, и большинство бизнес-процессов зависит от качества получаемых данных. Таким образом, спецификация требований также должна отвечать критериям качества данных.

Согласно стандарту ISO/IEC 25012 [7] модель качества данных содержит следующие характеристики: правильность, полнота, согласованность, достоверность, актуальность, доступность, соответствие, конфиденциальность, эффективность, точность, трассируемость, понятность, пригодность, переносимость, восстанавливаемость.

Основываясь на моделях качества, предложенных стандартами ISO/IEC 25010 и ISO/IEC 25012, а также на требованиях и рекомендациях стандарта ISO/IEC 25030, были выделены характеристики качества требований, а также подхарактеристики.

Спецификацию требований к ПО следует рассматривать как множество отдельно взятых требований, которые должны отвечать характеристикам качества, предъявляемым требованиям, а также как систему требований, к которой должна соответствовать своему набору характеристик качества.

Каждое требование к ПО должно обладать следующими свойствами. Каждое требование определяет важность, характеристики ПС и ограничения.

Требования должны быть актуальными и не устаревать с течением времени. Они не должны зависеть

от реализации: требование указывает, что должен делать продукт, а не как требование должно быть выполнено. Требование должно быть однозначными, т.е. указано таким образом, чтобы оно могло быть интерпретировано только одним способом как для тех, кто составляет его, так и для тех, кто его использует.

Описанное требование должно быть завершенным, т.е. оно не нуждается в дополнительных уточнениях, потому что оно измеримо и достаточно описывает возможности и характеристики, чтобы удовлетворить потребность заинтересованного лица. Требование должно быть единичным, или атомарным. Требование должно быть выполнимым, т.е. оно технически реализуемо, вписывается в системные ограничения. Требования должны быть отслеживаемыми (трассируемыми). Требования являются проверяемыми, если и только, если существует некий конечный эффективный процесс, используя который пользователь или машина могут убедиться, что программное средство удовлетворяет этому требованию.

Существуют определенные свойства, которые должны быть рассмотрены для совокупности требований, а не для отдельного требования.

Спецификация требований (СТ) является корректной, если, и только, если каждое требование, изложенное в ней, является требованием, которому должно удовлетворять программное обеспечение. СТ является полной (завершенной), если и только, если она включает следующие элементы.

Все существенные требования, независимо от того, относятся ли они к функциональным возможностям, рабочим характеристикам, проектным ограничениям, атрибутам или внешним интерфейсам. В частности, должны быть подтверждены и обработаны любые внешние требования, налагаемые спецификацией системы. Определение откликов программного обеспечения на все классы входных данных, которые могут быть реализованы, во всех возможных ситуациях.

СТ не должна включать требования, которые противоречат друг другу. Требования не должны дублироваться. СТ является внутренне непротиворечивой, если и только, если никакой набор отдельных требований, описанных в ней, не находится в противоречии с ней.

Использование стандартной терминологии и определений поддерживает непротиворечивость. СТ должна быть ограниченной, т.е. описывать определенный рамки для предполагаемого решения без описания того, что выходит за рамки потребностей пользователей. СТ должна быть модифицируемой.

При создании СТ должно учитываться упорядочивание по значимости и/или устойчивости.

Опираясь на вышеперечисленные свойства, предъявляемые к требованиям к ПО, можно описать модель качества требований к ПО, включающую характеристики и подхарактеристики качества.

1. Характеристика качества «Функциональное соответствие» представляет собой совокупность свойств спецификации требований, определяющую пригодность, необходимость, приоритетность требований, а также соответствие всем нуждам заинтересованных лиц и требованиям бизнеса. Данная характеристика включает в себя следующие подхарактеристики: функциональная пригодность, функциональная правильность, эффективность и обязательность.

2. Характеристика качества «Полнота» представляет собой совокупность свойств требований, характеризующую их завершенность, согласованность между собой, непротиворечивость друг другу, а также атомарность и целостность.

3. Характеристика «Практичность» – это совокупность свойств требований, характеризующая конкретность, однозначность, достоверность, актуальность, корректность и понятность каждого отдельного требования в спецификации, а также простоту использования всей спецификации в целом.

4. Характеристика «Сопровождаемость» включает в себя следующие подхарактеристики: измеримость требования, трассируемость, выполнимость, достижимость, проверяемость, модифицируемость, анализируемость.

Таким образом, были предложены и описаны характеристики и подхарактеристики качественных требований, предъявляемых к программным средствам. Оценка качества требований является важной частью обеспечения качества ПО.

Список использованных источников:

1. Леффингуэлл Д., Уидринг Д. Принципы работы с требованиями к программному обеспечению. – М., 2002. – С. 5–23.
2. Вигерс К. Разработка требований к программному обеспечению /К. Вигерс. – М.: Русская Редакция, 2004.
3. 830-1998 — IEEE Recommended Practice for Software Requirements Specifications. 1998.
4. ISO/IEC/IEEE 29148:2011 Systems and software engineering -- Life cycle processes -- Requirements engineering
5. ISO/IEC 25030:2007 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements
6. ISO/IEC 25010:2011 : Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models.
7. ISO/IEC 25012:2008 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Data quality model

СЕКЦИЯ «ФИЗИКА»

РАСПРОСТРАНЕНИЕ ЭЛЕКТРОМАГНИТНОГО ИЗЛУЧЕНИЯ

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

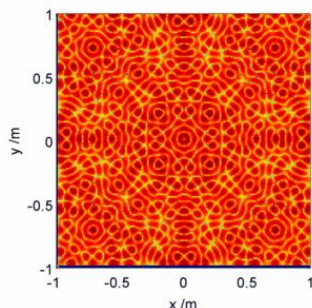
Фисько Д. В., Каранчук Р. С.

Смирнова Г. Ф. - к. ф. - м. н., доцент

Мы использовали уравнения Гельмгольца для моделирования распространения электромагнитных волн. Более точно, если временную зависимость электромагнитной волны можно представить формой $\sin(\omega t)$ и рассеивание дано отношением $w = ck / n(x)$ для некоторого индекса преломленного распространения $n(x)$, тогда электрическое поле E находится через: $\nabla^2 E + \frac{k^2}{n^2} E = f(x)$, где $f(x)$ некоторая функция источника. Дан источник излучения и геометрия распространения, в принципе уравнение Гельмгольца может быть решено для полного излучения поля $E(x, y, z)$. Но на практике это трудно, достижимо, в виду вычислительных сложностей. В работе мы выбрали двумерную модель и установили размер вычислительной сетки $N \times M$ с ячейками (i, j) для которых выполняется $1 < i < N, 1 < j < M$. С учетом такой дискретности, получаем уравнение ниже:

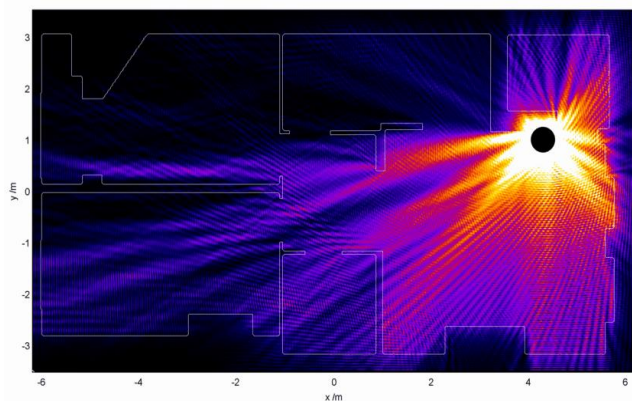
$$\frac{E(i+1, j) + E(i-1, j) - 2E(i, j)}{\Delta x^2} + \frac{E(i, j+1) + E(i, j-1) - 2E(i, j)}{\Delta y^2} + \frac{k^2}{n(i, j)^2} E(i, j) = f(i, j)$$

Это линейное уравнение в текущей ячейки $E(i, j)$, как функция от 4 соседних ячеек. Каждая ячейка представляет уравнение, которое характеризует влияние на соседей, это $N \times M$ уравнений с $N \times M$ неизвестными. Направивается применение линейной алгебры для решения задачи — если все уравнения могут быть представлены как одна большая матрица уравнений, эта матрица может быть инвертирована и мы получим точное решение для E . В частности мы будем иметь $M^*E = f$ для некоторой матрицы M , т.е. мы можем найти $E = M^{-1} * f$, где есть M пустых ячеек между $1/\Delta x^2$ и $1/\Delta y^2$ ячейками. На самом деле, очевидно, что подавляющее большинство матрицы равно нулю, что может помочь при рассмотрении огромных размеров матрицы. Для этой задачи помещение составляет около 5 метров в диаметре, а длина волны составляет около 5 см. Нам потребуется $N > 500$, что означает количество элементов около 10^{10} ! Таким образом, хранение матрицы с единичной точностью потребует около 60 GB RAM. И наши вычислительные средства не смогут инвертировать такую матрицу, даже если сможем разместить в памяти. Для решения этой проблемы, будем использовать понятие разреженной матрицы или матрицы заполненной в основном нулями.



Как же всё-таки выглядит решение уравнение Гельмгольца, на единичном квадрате для больших K ? Установим граничное условие $E=0$, на границы квадрата и предположим $f(x, y) = \delta(x, y)$. Для визуализации, ниже выбраны нескольких полученных картинок, где мы взяли логарифмическую цветовую гамму, чтобы подчеркнуть позиции с нулевым электрическим полем - «узлы». По мере того как анимация продолжается, K начинается с высоких значений и постепенно снижается, соответствующие быстрые колебания пластины постепенно становятся все медленнее.

Для демонстрации, мы взяли модель помещения, в которой мы хотим смоделировать распространения электромагнитных волн, для анализирования оптимальности точки расположение для Wi-Fi.



Мы используем этот план помещения в качестве карты показателя преломления — стены имеют очень высокий показатель преломления, а пустое пространство имеет показатель преломления равный 1. Затем, мы создали WiFi антенну в качестве небольшого источника излучения расположенного в углу. Начиная с длиной волны излучения 10 см. Также мы попытались добавить в мнимую часть показателя преломления для стен. Это позволило учитывать некоторое поглощение в бетоне, таким образом, мы уже не симулируем только идеальный процесс.

Список использованных источников:

1. Электромагнитные колебания и волны. Автор. Хайкин Семен Эммануилович г.1959
2. Helmholtz equation (jasmscole.com/2014/08/25/helmhurts)

МЕТОД РЕЗЕРФОРДОВСКОГО ОБРАТНОГО РАССЕЯНИЯ КАК СПОСОБ ИЗУЧЕНИЯ СТРУКТУРЫ ВЕЩЕСТВА: ОСНОВЫ И ТЕХНОЛОГИИ

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

Момотова Ю.О., Оверченко А.С., Солодкий Д.М.

Ташлыкова-Бушкевич И.И., доцент

В статье рассматриваются основы метода резерфордского обратного рассеяния (POP) ионами гелия и водорода. Обоснован и представлен программный комплекс iRB.Space, разрабатываемый для автоматизации анализа элементного состава твердых тел методом POP.

Исследование поверхностного состава вещества может выполняться разрушающими и неразрушающими методами. Метод резерфордского обратного рассеяния (POP) позволяет производить неразрушающий качественный и количественный анализ элементного состава и структуры образца в приповерхностной области. Это делает метод незаменимым при решении ряда практических задач, связанных с изучением свойств твердых тел [1]. Он применяется при исследовании профилей распределения примесей в полупроводниковых материалах, состава и геометрии тонких пленок, а также для изучения модификации структуры кристаллических материалов, включая кинетику накопления радиационных дефектов при ионном облучении [2].

Ключевой особенностью метода POP является использование высокоэнергетических ионов гелия, проникающих глубоко внутрь твердого тела и рассеивающихся обратно. При столкновении часть энергии передается от движущейся частицы неподвижному атому мишени; уменьшение энергии рассеянной частицы зависит от ее массы и массы атома мишени. Кинематика упругого столкновения двух изолированных частиц описывается с помощью законов сохранения энергии и импульса, что позволяет идентифицировать атом мишени. Затем происходит регистрация энергии ионов, рассеянных на некоторый угол относительно своего первоначального направления распространения [3]. Пример энергетического спектра, построенного на основе полученных данных приведен на рис. 1.

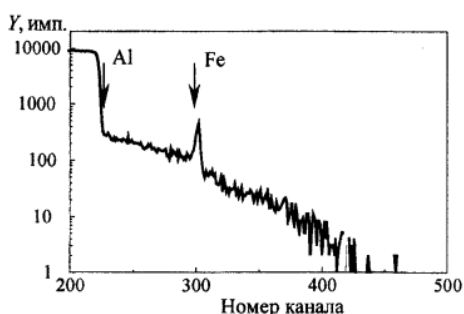


Рис. 1. Энергетический спектр ОР поверхности фольги сплава Al-Fe, полученный при $E_0=2$ МэВ

является единственной массовой коммерческой (платной) ОС, а RUMP вовсе является консольной утилитой с текстовым интерфейсом времен MS-DOS. В то же время мощность современных вычислительных устройств позволяет производить расчеты и на смартфонах, на общем рынке вычислительных устройств доминируют бесплатные модификации *NIX-подобных ОС (OS X, iOS, Android, Debian, RHEL, Arch, *BSD). Также лицензионная политика указанных программ не в пользу исследователей: если RUMP распространяется бесплатно, то лицензия Simnra составляет от 250 €.

Вышеизложенная ситуация на рынке побудила авторов статьи приступить к разработке собственного решения под названием iRB.Space. Программный комплекс iRB.Space разрабатывается не только для анализа и хранения результатов исследований методом POP, но и моделирования состава с возможностью совместной работы. Архитектурное решение по технологии SaaS (“программное обеспечение как услуга”) позволяет использовать решение независимо от операционной системы и типа вычислительного устройства, а также предоставляет конкурентные преимущества в виде работы нескольких исследователей одновременно. С помощью Web-интерфейса можно анализировать, моделировать и просматривать результаты без установки дополнительного ПО.

Список использованных источников:

1. Ташлыкова-Бушкевич, И.И. Элементный послойный анализ распределения компонентов в объеме быстротвердевших низколегированных сплавов алюминия / И.И. Ташлыкова-Бушкевич, В.Г. Шепелевич // Физика и химия обработки материалов. – 2000. – № 4. – С. 99-106.
2. Афанасьев, М.С. Особенности POP-спектроскопии тонких пленок перовскитов / М.С. Афанасьев, А.В. Буров, В.К. Егоров, П.А. Лучников, Г.В. Чучева // Вестник науки Сибири. – 2012. – №1. – С. 127-128.

3. Швей, И.В. Исследование дефектообразования в слоях дисилицида кобальта, синтезированных ионно-лучевым методом в кремнии: дипломная работа / И.В. Швей. – Москва: МГУ, 2001. – 35 с.
4. Солодкий, Д.М. Композиционный анализ состава твердых тел методом резерфордовского обратного рассеяния: программные комплексы / Д.М. Солодкий, И.И. Ташлыкова-Бушкевич // Актуальные направления научных исследований XXI века – сб. докладов. – Воронеж: ВГЛУ, 2015. – № 8, ч. 1. – С. 275-278.
5. Computer Graphic Service [Электронный ресурс]. – Режим доступа: <http://www.genplot.com/>.
6. Simnra [Электронный ресурс]. – Режим доступа: <http://home.rzg.mpg.de/~mam/>.

МЕТОДЫ СИНТЕЗА ЗНАНИЙ, ОПЫТА И РЕСУРСОВ ПОЛЬЗОВАТЕЛЕЙ В КОНТЕКСТЕ «ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СОВМЕСТНОЙ РАБОТЫ»

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

Краснов А.Ю.

Воронов А. А. – к-т. техн. наук, доцент

Тема доклада раскрывается в рамках магистерской диссертации: «Алгоритмы объединения и визуализации ресурсов пользователей на примере музыкальной социальной сети». Цель магистерской работы - разработка метода объединения ресурсов пользователей и интеграция данного метода в веб-приложение «социальная музыкальная сеть».

Актуальность исследования обусловлена развитием проектных форм организации работ (проектного менеджмента), которые являются ответом на такие вызовы времени как ускорение темпов научно-технического прогресса, и высокая скорость социально-экономических изменений [1]. Таким образом, возникает необходимость оперативного группового обсуждения ситуации и проблем, в условиях отсутствия экспертов извне, а также недостатка времени. Это обуславливает использование знаний, опыта и ресурсов ограниченного числа людей в рамках решения совместных задач. Исходя из этого можно сделать вывод, что разработка технологии совместной работы и объединения ресурсов пользователей является актуальной.

Наиболее близким аналогом нашего программного продукта будет класс «программного обеспечения для совместной работы» (groupware) – тип программного обеспечения (далее – ПО), созданного с целью поддержки взаимодействия между людьми, коллективно работающими над решением общих задач [2]. К данному типу программного обеспечения применим закон Меткалфа, который гласит, что полезность сети пропорциональна квадрату численности пользователей этой сети. Таким образом, чем больше людей используют что-либо, тем более ценным оно становится.

Ниже приведена классификация «программного обеспечения для совместной работы» (далее – ПОСР):

- совместная работа над документами (Google Docs, Office Online);
- вики технологии [3] (Wikipedia);
- системы управления проектами (Jira, eGroupWare, Citadel, Microsoft Project);
- аудио, видео конференции (Skype, Skype For Business, Google Hangouts.);
- облачные хранилища данных с возможностями совместной работы над файлами (Dropbox, OneDrive, Яндекс-диск);

- корпоративные социальные сети (Битрикс24, DaOffice, Jive, Yammer);
- базы знаний[4] (в особенности экспертные системы) (CLIPS, WolframAlpha, MYCIN);
- интернет-опросы и голосования (Examinare, Surveymonkey, Testograf);
- менеджеры задач (LeaderTask, WonderList, Google Tasks).

Нами был разработан собственный метод синтеза знаний, опыта и ресурсов, использующий технологии, присутствующие вышеприведённым ПОСР:

- создания сообществ разными пользователями;
- разрушения иерархических границ между специалистами предприятия;
- организации контента: тэги, закладки;
- коллективного обсуждения (форумы, чаты);
- голосований, опросов, рейтингов;
- совместного редактирования контента в реальном режиме времени;
- контроля и мониторинга состояния системы (возможность подписки на обновления в системе, слежение за новостями в системе);
- публикации различного типа контента (ссылки, документы, изображения, аудио и видео);
- просмотра истории всех изменений в элементах системы.

Цель метода: синтезировать ресурсы, знания и опыт пользователей.

Задачи метода:

- разделить этапы структурирования области сбора информации и этапы заполнения этой области ресурсами;
- иерархически структурировать область сбора информационных ресурсов (построить «конфигуратор» - древовидную иерархию, структурирующую данную область);
- создавать условия для быстрого, удобного и последовательного редактирования любого элемента системы заинтересованным сообществом пользователей;
- осуществить отбор информации, опираясь на мнение большинства участников группы;
- произвести слияние ресурсов пользователей, по критериям значимости для данной области знания;
- предоставить результат слияния в удобной визуальной форме.

Последовательность этапов работы сообщества по синтезу знаний:

1. Определение области, по которой будет осуществляться сбор информации, синтез знания и опыта;
2. Определение участников;
3. Построение конфигуратора:

3.1. Выдвижение каждым участником структур раздела конфигуратора первого уровня. Голосование и определение победителя;

3.2. Дополнение и редактирование победившего предложения путем внесения предложений и голосований по данному разделу;

3.3. Обсуждение процесса редактирования каждого из разделов;

3.4. Структурирование разделов путем определения подразделов и редактирования их выше описанными методами (глубина иерархии определяется участниками, нижние уровни иерархии предназначены для их непосредственного заполнения содержимым);

4. Наполнение содержим нижних ячеек конфигуратора (файлами, текстами, картинками документами, ссылками):

4.1. Внесение участниками содержимого разного типа;

4.2. Голосование и определение лучшего «связывающего текста» (взятие его за основу) и прилагаемых к нему «файлов-ресурсов» (под «связывающим» подразумевается текст, содержащий ссылки на «файлы-ресурсы» и таким образом объединяющий весь контент);

4.3. Редактирование «связывающего текста» и прилагаемых к нему «файлов-ресурсов»;

4.4. Повторение вышеприведенной процедуры для пункта «важные дополнительные материалы», т.е. конструирование «связывающего текста» и сборка прилагаемых к нему «файлов-ресурсов».

Пример работы метода синтеза знаний, опыта, ресурсов в контексте встраивания его в функционирование «музыкальной социальной сети».

Один из участников сети создает новую сессию и определяет тему, по которой будет производиться синтез ресурсов (предположим, что темой сессии будет сбор информации о музыкальной группе). После этого он добавляет необходимых пользователей сети (например, друзей) в эту сессию, наделяя их правом сделать аналогичное предложение, например, своим друзьям.

Следующим этапом является построение конфигуратора, для этого участники должны вынести на рассмотрение сообщества структуры разделов первого уровня. После внесения предложений производится голосование и определяется победитель, группа переходит к рассмотрению и редактированию победившей структуры. Во время редактирования каждый участник может предложить изменить существующий раздел, удалить его или добавить новый раздел. После предложений запускается голосование и по его итогам определяется судьба раздела. В процессе голосования по каждому разделу идет обсуждение в чате. Так как чаты разделены по разделам, общий информационный поток обсуждения не перегружается, чаты разделены и структурированы, удобны для работы. Каждый отредактированный раздел содержит историю изменений с набором всех предложений, итогами голосований за них, историей обсуждений (чаты) по каждому предложению.

Предположим, что разделами первого уровня выбраны: история группы, дискография, состав группы, концерты, статьи о группе, интересные факты. Далее по каждому из разделов запускается аналогичная процедура которая позволяет уточнить из каких подразделов он состоит и разделить его на более мелкие структурные единицы.

Когда иерархия готова участники начинают заполнять нижние уровни (листья дерева). К этому моменту раздел дискография может состоять из подразделов с конкретными названиями альбомов группы, а раздел интересные факты может быть разделен по временным отрезкам (70-ые гг., 80-ые гг. и. т. д).

В процессе наполнения ресурсами участники вносят свои материалы в виде «связывающего текста» и «файлов ресурсов», прилагающихся к нему. Таким образом, в итоге у них получается текст по данному разделу со ссылками либо на сторонние ресурсы, либо на загружаемые ими (файлы, картинки, аудио или видео). После того как группа выбрала лучший текст из предложенных по данному тексту, строится итоговое представление, которое содержит результирующий «связывающий текст» для данного раздела. Далее участники могут работать над ним делая правки, удаления или добавления, прикрепляя новые ресурсы и так далее.

Выводы. Разработанный нами метод синтеза знаний, опыта и ресурсов позволяет заинтересованным сообществам, члены которых не могут общаться напрямую, использовать компьютерные технологии для наиболее эффективного общения, целью которого может быть обмен, анализ, слияние и синтез информационных ресурсов участников. В результате работы они получают структурированную и заполненную иерархию разделов по выбранной теме, которая может в перспективе дорабатываться и наполняться новым содержимым. Участники могут подписываться на обновления в системе и просматривать историю модификации разделов. Созданный участниками конфигуратор может использоваться в качестве основы для последующих сессий сбора информации по аналогичным темам.

Список использованных источников:

1. М. К. Румизен Управление знаниями. Полное руководство. – М: «АСТ, Астрель», 2004. – 317 с.
2. Groupware [Электронный ресурс]. – Режим доступа: <http://nexus.awakentech.com:8080/at/awaken1.nsf/UNIDs/CFB70C1957A686E98825654000699E1B?OpenDocument>
3. Иванов Д. Технология Вики / Д. Иванов, П. Смирнов [Электронный ресурс]. – Режим доступа: https://dulanov.wordpress.com/2004/06/01/tehnologiya_viki/
4. Гаврилова и др. Базы знаний интеллектуальных систем // Учебник для вузов. — СПб. Питер, 2000.

ОПТИЧЕСКИЕ ИЛЛЮЗИИ

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

Дацук А.А., Клименков В.В.

Смирнова Г.Ф., кандидат физ.-мат. наук, доцент

Несмотря на научный прогресс область оптических иллюзий все также остается на донаучной стадии и зачастую объяснения обманов зрения носят недостоверный характер. Таким образом предмет нашей работы нуждается в научных обоснованиях, а также в разработке основных методов и подходов к изучению той или иной иллюзии.

Рассмотрим причины возникновения оптических иллюзий. Зрительный аппарат человека устроен очень сложно и оптические свойства нашего глаза несовершенны. Отметим, что на сетчатке глаза картинка, которую мы видим перевернута, также изображение может быть расфокусировано. Более того, даже сфокусировавшись на изображении, глаз совершает слабые колебания. Ко второй причине, вызывающей иллюзии можно отнести то, что у всех людей разные представления о мире. В большинстве случаев человек не хочет видеть то, что он не ожидает увидеть. И вообще можно сказать, что наше зрение — это одна сплошная иллюзия, ведь всё что мы видим — это результат химических реакций в сетчатке глаза, подкреплённый работой мозга.

Существует большое количество разновидностей оптических иллюзий. Некоторые из них связаны с определением образа в зависимости от его пространственного положения, другие основаны на искажении восприятия цветов, третьи — это так называемые двусмысленные картинки и т.д.

Возможно, мы этого и не замечаем, но сегодня в нашей жизни оптические иллюзии встречаются достаточно часто. И природа как источник иллюзий тоже не исключение. Например, голубое небо или существование радуги — это явления, возникающие из-за особенностей рассеивания света.

То же касается и искусственных иллюзий. Зная особенности человеческого зрения, люди научились создавать невероятно красивые иллюзии в искусстве, а также полезные в быту. На самом деле, первые иллюзии начали появляться очень давно, причём в архитектуре. Самый впечатляющий пример — древнегреческий храм Парфенон. Его колонны были спроектированы таким образом, что храм казался больше чем он есть на самом деле. Отличный пример иллюзий в современной архитектуре — это здание таможни в Мельбурне (рис.1), фасад которого выполнен в виде параллельных линий. Однако, из-за иллюзии искривления вряд ли можно сказать, что они параллельны. Также иллюзии широко применяются в цирке, изобразительном искусстве, дизайне, рекламе и т. д.



Рис.1 Здание таможни в Мельбурне



Рис.2 Иллюзия 3-D

Рассмотрим одну из интереснейших оптических иллюзий — эффект голограммы. Конечно, принцип работы устройства в нашем опыте — это не совсем голография. Также мы не получаем настоящий 3-D объект, который сделать на самом-то деле не так просто, т.к. нужен источник света, обладающий достаточной степенью когерентности. В данном опыте мы получили вполне убедительную оптическую иллюзию 3-D изображения (рис.2), созданную с помощью 2-D картинки.

Таким образом, мы встречаемся с оптическими иллюзиями практически ежедневно, знание их основных видов и причин это не только интересно, но и может избавить нас от неприятных ситуаций, которые могут возникнуть из-за зрительных обманов.

Список использованных источников:

1. Перельман Я.И. Занимательная физика. Москва. 1981, с.73.
2. А. А. Вадимов, М.А. Тривас. Иллюзии зрения. Москва: Наука, 1971.
3. Рассел К., Картер Ф. Улучши свой интеллект. Минск, «Попурри» 1996, с.45.
4. Толанский С. Оптические иллюзии. Москва, Мир, 1967

5. Сечель А. Удивительные трюки зрения: как работают оптические иллюзии. Москва, Эксмо, 2011

ФИЗИКО-МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА ТАЯНИЯ ЛЕДНИКОВ И ЕЁ ПРОЕКЦИЯ НА ГЕОФИЗИЧЕСКУЮ ПРОБЛЕМУ ПОВЫШЕНИЯ УРОВНЯ МИРОВОГО ОКЕАНА

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

Шарибченко А. С., Бирюков В. С.

Смирнова Г. Ф. – канд. физ.-мат. наук, доцент

В современном мире человечество на пути развития постоянно сталкивается с глобальными проблемами, которые влияют абсолютно на все стороны жизни людей и стран в целом. Одной из таких проблем является проблема повышения уровня мирового океана, непосредственной причиной чего является всемирное таяние ледников.

Оценка последствий повышения уровня мирового океана необходима для примерного расчёта количества мировых территорий, которые могут быть затоплены. Для решения поставленной задачи будем считать, что таянию больше всего подвержен Гренландский ледяной щит, который является вторым по размеру ледяным щитом (первым – Антарктический).

Таяние льда – простой процесс встречающийся в повседневной жизни каждого человека. Но у больших массивов льда данный процесс проходит гораздо сложнее и связан с перетеканием льда под действием гравитации с одного места на другое.

Так как это физико-математическая модель процесса, то мы вольны делать некоторые допущения:

1. В реальной жизни территория Гренландии – неправильная математическая фигура. Мы возьмём в качестве приближения остров прямоугольной формы со сторонами равными: $h_a = 2L$, $h_b = 5L$, где $L = 410$ км – коэффициент размерности расстояния.

2. Профиль высоты ледяного щита $H(x)$ не зависит от координаты y , и растёт от 0 при $x = \pm L$, до H_{max} при $x = 0$.

3. Несмотря на большую толщину ледяного щита, лёд в его основании несжимаем, то есть считаем, что лёд – несжимаемое вещество.

На рисунках 1, 2, 3 наглядно изображены допущенные нами упрощения модели:



Рис. 1 – Территория Гренландии, покрытая льдом.

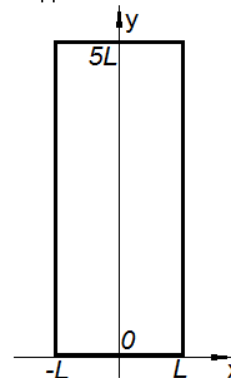


Рис. 2 – Модель территории Гренландии, используемая в работе.

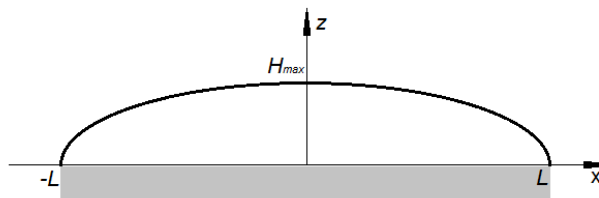


Рис.3 – Вертикальное сечение в плоскости xz ледяного щита, показывающее профиль высоты $H(x)$.

В малые промежутки времени профиль высоты щита постоянен, и давление внутри него описывается формулой:

$$p(x, z) = \rho g(H(x) - z),$$

где z – высота, на которой нас интересует давление, а x – расстояние от линии раздела льда.

Если рассмотреть произвольную тонкую вертикальную пластину льда, опирающуюся на Δx и Δy , которая находится в состоянии равновесия, то возникает горизонтальная составляющая силы давления, которая

действует на каждую вертикальную сторону:

$$F = \int_0^{H(x)} p(x, z) \Delta y dz = \rho g * \frac{H^2(x)}{2} * \Delta y$$

и результирующую ΔF :

$$\Delta F = F(x) - F(x + \Delta x) = -\rho g H(x) \frac{dH}{dx} \Delta x \Delta y,$$

которая уравновешивается силой трения льда о землю, которая в этом случае рассчитывается по формуле: $F_{\text{тр}} = \beta * \Delta x \Delta y$, где коэффициент напряжения сдвига $\beta = 100 \text{ кПа}$. Из этого получаем

$$H(x) dH = -\frac{\beta}{\rho g} dx$$

откуда, с учётом $H(L) = 0$:

$$H(x) = \sqrt{\frac{2\beta L}{\rho g}} * \sqrt{1 - x/L} \rightarrow H_{\text{max}} = \sqrt{\frac{2\beta L}{\rho g}}.$$

Теперь мы можем найти объём льда, который находится на модели Гренландии:

$$V_{\text{лёд}} = \int_0^{5L} \int_{-L}^L H(x) dx dy = \frac{20}{3} L^2 \sqrt{\frac{2\beta L}{\rho g}}.$$

Максимальное повышение уровня мирового океана будет при полной расплавке ледяного щита Гренландии. Будем считать, что

процесс поднятия уровня океана происходит только на территории океана, то есть затопления не происходит.

Так как масса растаявшего льда и образовавшейся воды равна, то $V_{\text{вода}} = \frac{\rho}{\rho_{\text{вода}}} V_{\text{лёд}}$

Тогда уровень, на который поднимется океан, будет равен:

$$\Delta h = \frac{V_{\text{вода}}}{S_0} = \frac{20}{3} \frac{\rho}{\rho_{\text{вода}}} \frac{L^2}{S_0} \sqrt{\frac{2\beta L}{\rho g}}$$

где площадь территории океана $S_0 = 3,6 * 10^{14} \text{ м}^2$.

Подставив числа, находим, что уровень океана поднимется на $\Delta h = 8,5 \text{ м}$.

На территориях, где уровень поверхности меньше полученного результата проживает более 10% населения земли, а значит – они будут вынуждены искать новый дом в другом месте.

Полученная цифра близка к оценкам учёных-геофизиков, которые рассчитали, что при полном превращении льдов Гренландии в воду уровень мирового океана поднимется более чем на 7м.

Таким образом была разработана модель таяния ледника, для которой был использован метод физико-математического моделирования. Этот метод позволил довольно точно оценить искомую величину, с учётом допустимых приближений. Результат работы показывает необходимость поиска человечеством способов по предотвращению повышения температуры Земли, иначе появление и обострение других глобальных проблем, таких как чрезмерное повышение плотности населения и голод, неизбежно.

Список использованных источников:

1. Гренландский ледяной щит. – Электрон. текстовые дан.– Режим доступа: https://ru.wikipedia.org/wiki/Гренландский_ледяной_щит, свободный.
2. Математическая модель.–Электрон. текстовые дан.–Режим доступа: https://ru.wikipedia.org/wiki/Математическая_модель, свободный.

КОЛЕБАНИЯ УПРУГОЙ СТРУНЫ

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

Бублей Д.В., Атарик К.А.

Смирнова Г.Ф. – кандидат физ.-мат. наук, доцент

Математика служит языком физики. Результаты математических изысканий находят применение в описании различных физических явлений и процессов. Однако случается и так, что решение совершенно практической физической задачи служит мощнейшим толчком в развитии математики. Одной из таких задач является задача колебания струны.

Рассмотрим малые колебания струны. Основная характеристика состояния струны – её отклонение от стационарного положения – $U(t, x)$ (Рис.1).

Рассмотрим силы, действующие на бесконечно малый участок струны (Рис.2).

Согласно второму закону Ньютона:

Рис.1 Форма струны в определённый момент.

$$\vec{F} = m\vec{a}.$$

$|\vec{F}_1| = |\vec{F}_2| = T$, где T – сила натяжения струны.

$a = \ddot{U}(t, x)$ – по определению ускорения.

Таким образом, второй закон Ньютона для рассматриваемого участка струны запишется в виде:

$$\Delta m \ddot{U}(t, x) = T \sin \theta_2 - T \sin \theta_1 - \gamma \Delta x U(t, x + 1/2 \Delta x) - \Delta mg \quad (1),$$

где γ – модуль Юнга среды, g – ускорение свободного падения.

Вычислим значения синусов, входящих в уравнение (1). Т.к. колебания малые то и углы θ_1 и θ_2 малы, тогда:

$$\sin \theta \approx \tan \theta = U'(t, x)$$

$$\sin \theta_1 = U'(t, x)$$

$$\sin \theta_2 = U'(t, x + \Delta x)$$

Рис.2 Силы, действующие на участок струны.

В итоге уравнение (1) преобразуется к виду:

$$\Delta m \ddot{U}(t, x) = T(U'(t, x + \Delta x) - U'(t, x)) - \gamma \Delta x U(t, x + 1/2 \Delta x) - \Delta mg.$$

Разделив на Δx и обозначив линейную плотность струны как λ имеем:

$$\lambda \ddot{U}(t, x) = TU''(t, x) - \gamma U(t, x) - \lambda g \quad (2) \quad (\text{полное уравнение колебания струны}),$$

$$\lambda \ddot{U}(t, x) = TU''(t, x) - \lambda g \quad (3) \quad (\text{без учёта воздействия упругой среды}),$$

$$\lambda \ddot{U}(t, x) = TU''(t, x) \quad (4) \quad (\text{без учёта воздействия упругой среды и силы тяжести}).$$

Получаем уравнения в частных производных, для их решения необходимо определить начальные условия.

Рассмотрим решение уравнения (4) в общем виде. Не ограничивая общности, определим начальные условия:

$$\begin{cases} U(0, x) = \varphi(x), \\ \dot{U}(0, x) = \omega(x). \end{cases}$$

Где

$\varphi(x)$ - описывает изначальную форму струны,

$\omega(x)$ - описывает изначальную скорость струны в каждой её точке.

Перепишем уравнение (4) в виде $\ddot{U}(t, x) = a^2 U''(t, x)$, где $a^2 = \frac{T}{\lambda}$, $[a] = \frac{m}{c}$. Решив это дифференциальное уравнение получим

$$U(t, x) = \frac{\varphi(x - at) + \varphi(x + at)}{2} + \frac{1}{2a} \int_{x-at}^{x+at} \omega(\tau) d\tau$$

Подобные решения были получены Даламбером и Эйлером.

В этом решении $\varphi(x)$ и $\omega(x)$ – произвольные функции, это и вызвало

серьёзные споры среди математиков XVIII века. Проблема заключалась в различном трактовании понятия произвольной функции. До решения этой задачи все рассматриваемые математиками функции составлялись путём комбинирования элементарных функций. Тогда существовали два определения функции: 1) функция – это закон, позволяющий по определённым правилам вычислять её значение в зависимости от величины её аргумента (подход Бернулли), 2) функция – переменная величина, изменяющаяся во времени (подход Ньютона). Возникшие разногласия послужили толчком к развитию дифференциального исчисления и функционального анализа.

Позже Фурье, решая эту же задачу, предположил, что форму струны в любой момент времени можно представить как сумму бесконечного ряда некоторых функций, например тригонометрических. То есть, если начальные условия представимы в виде

$$\begin{cases} U(0, x) = \varphi(x) = \sum_{n=1}^{\infty} A_n \sin \frac{\pi n x}{l} \\ U'(0, x) = \omega(x) = \sum_{n=1}^{\infty} B_n \frac{\pi n}{l} \sin \frac{\pi n x}{l} \end{cases} \quad \text{где} \quad \begin{cases} A_n = \frac{2}{l} \int_0^l \varphi(x) \sin \frac{\pi n x}{l} dx \\ B_n = \frac{2}{\pi n a} \int_0^l \omega(x) \sin \frac{\pi n x}{l} dx \end{cases}$$

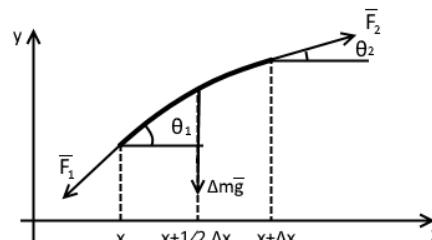
то искомое решение уравнения (4) имеет вид:

$$U(t, x) = \sum_{n=1}^{\infty} U_n(t, x) = \sum_{n=1}^{\infty} (A_n \cos \frac{\pi n}{l} at + B_n \sin \frac{\pi n}{l} at) \sin \frac{\pi n}{l} x$$

Ещё одна проблема, поднимаемая данной задачей, — проблема дифференцируемости функций $\varphi(x)$ и $\omega(x)$. Так как $\varphi(x)$ и $\omega(x)$ произвольные функции, они могут претерпевать излом в некоторых точках, т.е. не являются дифференцируемыми в них, однако, получить решение и при подобных начальных условиях оказывается возможным. Тогда возникает вопрос: какой смысл имеет дифференциальное уравнение, если его решением является не дифференцируемая функция. Эта проблема разрешилась лишь в 1935 году, с введением С.Л. Соболевым понятия обобщённой функции, что также позволило дать строгое математическое описание таким физическим моделям как материальная точка, точечный диполь и др.

Список использованных источников:

1. Christensen, T. Eighteenth-Century Science and the Corps Sonore: the Scientific Background to Rameau's Principle of Harmony // Journal of Music Theory. — 1987.
2. Стиллвелл, Дж. Математика и её история. — Ижевск: Институт компьютерных исследований/РХД, 2004. — 530 с.



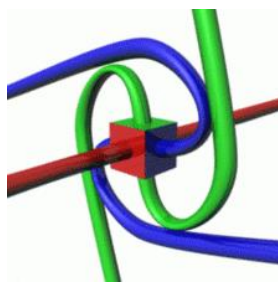
СПИТРОНИКА: БУДУЧЫНЯ ЭЛЕКТРОННЫХ ПРЫЛАД

Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі
г. Мінск, Рэспубліка Беларусь

Былинович У.М., Дзікі А.О.

Смірнова Г.Ф. - к. ф. - м. н., дацэнт

Традыцыйныя электронныя прылады пакладаюцца на транспарціроўку электрычных носьбітаў зарадаў (электронаў) у паўправядніках, напрыклад, у крэмніі. Аднак, зараз фізікі спрабуюць паставіць на службу чалавецтва спін электрона, а не яго зарад, каб стварыць новае пакаленне прылад, якія будуць менш па памеры, больш універсальныя, і больш надзейныя, у параўнанні з тымі, што выкарыстоўваюць крэмніевыя чыпы і інтэгральныя схемы.



Мал. 1 - Аб'ект са спінам $\frac{1}{2}$

Спітроніка (спінавая электроніка) - раздзел квантавай электронікі, які займаецца вывучэннем спінавага токапереноса (спін-палярызаванага транспарту) у цвёрдацельных рэчывах, у прыватнасці ў гетэраструктурах ферромагнетык-парамагнетык або ферромагнетык-звышправяднік.

Спін - уласны момант імпульсу элементарных часціц, які мае квантавую прыроду і не злучаны з перамяшчэннем часціцы як цэлага. Спін вымяраецца ў адзінках \hbar (прыведзенай пастаяннай Планка, альбо пастаяннай Дзірака) і роўны $\hbar J$, дзе J - характэрнае для кожнага гатунку часціц цэлы (у тым ліку нулявы) альбо паўцэлы станоўчы лік - так званы спінавы квантавы лік, які звычайна называюць проста спінам. Згодна прынцыпу квантавання праекцыі спіна на выбраную вось, электроны падзяляюць на два тыпа носьбітаў току: электроны са спінам-уверх і электроны са спінам-уніз ($\frac{1}{2}$ або $-\frac{1}{2}$).

У канцы 1988 года было выяўлена, што паток спін-палярызаваных электронаў у наслоенай канструкцыі пакрыцця (два тонкіх пласта ферромагнетыка, падзеленыя пластам немагнітнага металу) можна значна змяняць, перакідаючы на супрацьлеглую палярнасць знешняга магнітнага поля. Гэты эфект, які атрымаў назву GMR, альбо гіганцкае магнітасупраціўленне, дазволіў ствараць нашмат больш адчувальныя магнітныя галоўкі і, адпаведна, памяншаць памеры магнітных даменаў, кадавальныя двайковыя дадзеныя на пласцінах. Інакш кажучы, была значна павялічана інфармацыйная ёмістасць назапашвальнікаў на цвёрдых магнітных дысках.

Маніпуліраванне спінавымі характарыстыкамі - перанос электронных спінаў паміж двума металамі – лёг ў аснову MRAM, магнітарэзістыўнай памяці адвольнага доступу - новага тыпу камп'ютэрных запамінальных прылад, якія захоўваюць інфармацыю без электрасілкавання.

Фізіка работы MRAM заснавана на эфекце, вядомым як тунэльнае магнітасупраціўленне (TMR), які нагадвае GMR. Тут два пласты ферромагнітнага металу, падзеленыя тонкім пластом ізалявальнага матэрыялу, такога як аксід алюмінія або аксід магнію. Калі ў GMR адбываецца павольнае, за кошт класічнай дыфузіі, перасоўванне спін-палярызаваных электронаў з аднаго ферромагнітнага пласта на іншы, то ў канструкцыі TMR мае месца асабліва квантавы тунэльны пераход праз пласт, які іх падзяляе (класічны забаронены працэс, пры якім часціца праходзіць праз патэнцыйны бар'ер, які перавышае яе кінэтычную энергію). Такого роду прылады называюцца магнітнымі тунэльных пераходамі, або MTJ (magnetic tunnel junctions).

Іншай сферай ужывання тэорыі спіна з'яўляецца спінавы дыёд. Ідэя спінавага двухкантактнага дыёда ўпершыню была прапанавана Мэцьюсам. Дыёд складаецца з пяціпластовай магнітнай сістэмы, у якой тры ферромагнітных пласты падзелены пластамі парамагнетыка. У будучыні плануецца выкарыстоўваць спінавыя дыёды як элементарныя ячэйкі MRAM-памяці.

Даследчыкі прадказваюць шырокае выкарыстанне распрацовак спітронік ў сферы квантавага камп'ютынга. Лічыцца, што наступным сур'ёзным этапам развіцця спітронікі стануць прылады, у якіх інфармацыя будзе перадавацца не пасродкам спінаў электронаў, а з выкарыстаннем складаных кубітавых пар. Напрыклад, шматкантактныя спінавыя прылады, якія могуць быць заснаваны на патоках забытых кубітаў. Практычна рэалізаваць падобную прыладу можна на базе спін-электронных транзістараў.

У артыкуле прадстаўлены агляд цяперашняга становішча і перспектывы спітронікі. Камерцыйны патэнцыял, закладзены ў спінавай электроніцы, стымулюе актыўнасць даследаванняў у гэтай галіне фізікі ва ўсім свеце, і поспехі, дасягнутыя ў спітроніке, магчыма, будуць вызначаць тэхналогіі XXI ст.

Спіс выкарыстаных крыніц:

1. А.В.Огнев, А.С.Самардак Спитроника: физические принципы, устройства, // Вестник ДВО РАН. 2006.№4.
2. Данилов Ю.А., Демидов Е.С., Ежевский А.А. Основы спитроники. Учебное пособие. Нижний Новгород, 2009, 173 с.

УСТРОЙСТВО РОТОРНОГО ВЕТРОГЕНЕРАТОРА. ЗАВИСИМОСТЬ ПРОИЗВОДИТЕЛЬНОСТИ ОТ ФОРМЫ И ПЛОЩАДИ ЛОПАСТЕЙ

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

Клыбик П.М., Королёва П.Б.

Смирнова Г.Ф. – к-т. ф-м наук, доцент

Рост потребления человеком электрической энергии и загрязнение окружающей среды призывает людей переходить на альтернативные источники энергии. В последнее время растёт спрос на ветряные электростанции. В связи с этим необходимо разработать модель, которая обеспечивала бы наивысшую эффективность.

Мощность ветра прямо пропорциональна массе воздуха и квадрату его скорости, значит эффективность работы устройства зависит от площади контакта воздушных масс с рабочей частью устройства. Поэтому считаем, что моделирование ветряной электростанции должно начинаться с тщательного продумывания будущего ветродвигателя (лопастей).

Ветроэлектростанции с горизонтальной осью вращения производят электроэнергию только тогда, когда дует ветер со скоростью 4-5 м/с - если их мощность более 200 кВт, или 2-3 м/с, если их мощность менее 100 кВт. Они состоят из башни, на вершине которой располагается кабина с электрогенератором и редуктором, к оси которого прикреплены лопасти ветровой турбины. Преимущество ветроэлектростанции с вертикальной осью вращения – расположение электрогенератора на земле, отсутствие необходимости ориентации на ветер.

В своей работе мы рассматриваем роторный (вертикальный) ветрогенератор, так как для него не требуется поднятие на большую высоту, отсутствует необходимость в ориентации на ветер, он имеет хорошие показатели при любой скорости ветра и не теряет своей эффективности при любых осадках, что является плюсом для внедрения роторных генераторов в нашей стране.

Роторный ветрогенератор мы смонтировали на сварной подставке, на стальном стержне укрепили разрезанные по диаметральной плоскости полые цилиндры со смещёнными друг относительно друга частями, на этом же стержне укрепили шкив, вращение с которого передаётся на автомобильный генератор. Такой ветродвигатель можно использовать на даче или на ферме, поднимая воду из скважины или колодца. При скорости ветра до 15 км/ч его мощность составит около 700 Вт. Использовали следующие схемы:

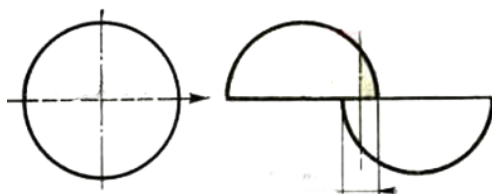


Схема №1



Схема №2

Первая модель полностью соответствует схеме №2. Вторая модель состоит из двух частей, которые отображены на схеме №1, расположенных одна над другой. Третья модель опять же состоит из элементов схемы №2, совместимых друг с другом и расположенных на определённой высоте на одной оси.

Работа всех трёх моделей была рассматривала в один день, а значит погрешности в сравнительном анализе быть не может. Была выведена площадь и формула лопастей, которые обеспечивают самую высокую скорость вращения ротора. В качестве базовой формулы использовали мощность:

$$P = \frac{1}{2} \cdot \rho \cdot \pi \cdot r^2 \cdot V^3,$$

где P – мощность ветрогенератора (Вт),

ρ – плотность воздуха (кг/м³),

$\pi = 3,14$,

r – радиус площади лопастей (м²),

V – скорость ветра (м/с).

Таким образом, были разработаны модели роторных генераторов, проанализированы эффективности их работы, также была выбрана схема, показавшая лучшие результаты. Установлено, что её применение покажет хорошие результаты как на местности, где наблюдается постоянная высокая скорость ветра, так и на местности, где скорость довольно прерывна. Данная модель проявляет себя с хорошей стороны при любой погоде и на любой высоте.

Список использованных источников:

1. Богданов, К. Ю. Ветер: энергия из ничего / Приложение к журналу «Квант» №6 2006 г.
2. Большая Российская энциклопедия. – Москва, 2006. – 1456с.

СЕКЦИЯ «ВЫСШАЯ МАТЕМАТИКА»

МЕТОД МОДЕЛИРУЕМОГО ОТЖИГА И ЕГО ПРИМЕНЕНИЕ ДЛЯ РЕШЕНИЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЯ УЧАСТИЯ АКТЕРОВ В КИНОСЪЕМКЕ

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

Рыбак Ю. А.

Луцакова И. Н. – канд. физ.-мат. наук, доцент

Алгоритм имитации отжига (Simulated annealing) — общий алгоритмический метод решения задачи глобальной оптимизации, особенно дискретной и комбинаторной оптимизации.

Алгоритм основывается на имитации физического процесса, который происходит при кристаллизации вещества, в том числе при отжиге металлов [1]. Предполагается, что атомы уже выстроились в кристаллическую решётку, но ещё допустимы переходы отдельных атомов из одной ячейки в другую. Предполагается, что процесс протекает при постепенно понижающейся температуре. Переход атома из одной ячейки в другую происходит с некоторой вероятностью, причём вероятность уменьшается с понижением температуры.

При помощи моделирования такого процесса ищется точка множества X , на которой достигается минимум некоторой числовой функции $F(\bar{x})$, где $\bar{x} = (x_1, x_2, \dots, x_m) \in X$.

Решение ищется последовательным вычислением точек $\bar{x}_0, \bar{x}_1, \dots$ пространства X ; каждая точка, начиная с \bar{x}_1 , «претендует» на то, чтобы лучше предыдущих приближать решение.

Алгоритм принимает точку \bar{x}_0 как исходную. На каждом шаге находится новая точка и понижается значение величины, понимаемой как «температура». Алгоритм останавливается, если не происходит улучшения значения целевой функции на протяжении определённого числа итераций, или по достижении точки, которая оказывается при температуре близкой к нулевой.

Точка \bar{x}_{i+1} по алгоритму получается на основе текущей точки \bar{x}_i следующим образом. К точке \bar{x}_i применяется оператор A , который случайным образом модифицирует ее, в результате чего получается новая точка \bar{x}^* .

Точка \bar{x}^* становится точкой \bar{x}_{i+1} с вероятностью $P(\bar{x}^* \rightarrow \bar{x}_{i+1} | \bar{x}_i)$, которая вычисляется в соответствии с распределением Гиббса:

$$P(\bar{x}^* \rightarrow \bar{x}_{i+1} | \bar{x}_i) = \begin{cases} 1, & F(\bar{x}^*) - F(\bar{x}_i) < 0, \\ \exp\left(-\frac{F(\bar{x}^*) - F(\bar{x}_i)}{Q_i}\right), & F(\bar{x}^*) - F(\bar{x}_i) \geq 0. \end{cases}$$

Здесь $Q_i > 0$ — элементы произвольной убывающей, сходящейся к нулю положительной последовательности, которая задаёт аналог падающей температуры в кристалле.

Алгоритм имитации отжига не гарантирует нахождения минимума функции, однако при правильной стратегии генерации случайных точек $\bar{x}_0, \bar{x}_1, \dots$ в пространстве X , как правило, происходит улучшение начального приближения.

Рассмотрим задачу составления расписания участия актёров в киносъёмке [2,3] и применим для ее решения данный алгоритм.

Крупная кинокомпания планирует съёмку фильма. Известно, что в съёмке фильма будут участвовать n актёров. Фильм состоит из m эпизодов. Каждый эпизод s_j , $1 \leq j \leq m$, снимается известное количество дней. Для каждого эпизода известны актёры, которые в нем задействованы. Актёр a_i , $1 \leq i \leq n$, может сниматься в нескольких эпизодах. Он приезжает на съёмочную площадку в тот день, когда снимается первый эпизод с его участием, и покидает площадку в конце последнего дня съёмок всех эпизодов с его участием. На рисунке 1 приведён пример подобного расписания.

За каждый день присутствия на съёмочной площадке кинокомпания выплачивает актёру a_i персональный гонорар C_i (независимо от того, занят ли актёр в съёмке в этот день или нет).

Необходимо определить, в каком порядке должны сниматься эпизоды, чтобы сумма выплаченных гонораров была минимальной.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	$c(a)$
a_1	X	-	X	-	-	X	-	X	X	X	X	X	2
a_2	X	X	X	X	X	-	X	-	X	-	X	-	
a_3	-	X	-	-	-	-	X	X	-	-	-	-	
a_4	X	X	-	-	X	X	-	-	-	-	-	-	1
a_5	-	-	-	X	-	-	-	X	X	-	-	-	
a_6	-	-	-	-	-	-	-	-	-	X	-	-	
cost	35	39	78	43	129	43	33	66	29	64	25	20	60
extra	0	20	28	34	84	13	24	10	0	10	0	0	22

Рис. 1 – Исходное расписание и его стоимость

	s_5	s_2	s_7	s_1	s_6	s_8	s_4	s_9	s_3	s_{11}	s_{10}	s_{12}	$c(a)$
a_1	-	-	-	X	X	X	-	X	X	X	X	X	20
a_2	X	X	X	X	-	-	X	X	X	X	-	-	5
a_3	-	X	X	-	-	X	-	-	-	-	-	-	4
a_4	X	X	-	X	X	-	-	-	-	-	-	-	10
a_5	-	-	-	-	X	X	X	X	-	-	-	-	4
a_6	-	-	-	-	-	-	-	-	-	-	X	-	7
cost	45	19	19	39	39	66	29	29	50	25	54	20	434
extra	0	0	10	4	9	10	20	0	0	0	0	0	53

Рис. 2 – Преобразованное расписание и его стоимость

Постановка задачи: необходимо найти такую перестановку множества $S = \{s_1, s_2, \dots, s_m\}$, т.е. последовательность съемки эпизодов, чтобы сумма $\sum_{i=1}^n L_i * C_i$ была минимальной. Здесь L_i - длина

оплачиваемого отрезка времени, C_i - стоимость одного дня актёра a_i . Пример исходного и преобразованного расписаний и их стоимости приведены на рис. 1 и рис. 2.

Как можно заметить, точное решение возможно при полном переборе вариантов, то есть при времени выполнения, зависящем от $m!$, что в большинстве случаев неприемлемо.

Однако применяя эвристические алгоритмы, одним из которых является метод моделируемого отжига, можно получать неплохие приближенные решения за небольшое время выполнения.

Тремя важнейшими составляющими алгоритма являются выбор оператора A , выбор начальной температуры и контроль изменения температуры. Эти три составляющие подбираются экспериментально и зависят от конкретной задачи.

Для написания программы алгоритма был выбран язык программирования Delphi. Разработанная схема экспериментов аналогична схеме из работы [4]. Начальная температура – 15, коэффициент, на который умножается текущая температура на каждой итерации равен 0,999. Это способствует тому, что в 25% случаев будет выбрано худшее расписание в качестве новой точки. Ограничение на количество итераций, при которых не происходит улучшения значения целевой функции – 2000. При воздействии оператора A выбиралось лучшее из расписаний-соседей исходной точки, которое могло получиться каким-либо из следующих способов: перестановка двух случайно выбранных элементов (эпизодов) множества S , перестановка двух случайно выбранных непересекающихся блоков элементов, поворот случайно выбранного блока. Наибольшую эффективность продемонстрировали первый и второй способы.

Компьютерные эксперименты проводились на тестовых примерах, взятых из работ [2,3]. В таблице 1 приводятся результаты, которые получились на тестовых примерах, в сравнении с результатами, полученными другими методами [2,3].

Табл. 1 – Результаты экспериментов

Тестовые примеры	Число эпизодов, m	Число актеров, n	Smith (2005) Время(s)	de la Banda et al [2] Время (s)	Qin et al (2016) [3] Время (s)	Общая стоимость [3]	Simulated annealing Время(s)	Simulated annealing Общая стоимость
MobStory	8	28	64,71	0,11	0,05	871	0,01	983
Film 103	8	19	76,69	0,06	0,02	1031	0,04	1056
Film 105	8	18	16,07	0,02	0,02	849	0,03	854
Film 114	8	19	127	0,08	0,03	867	0,04	869
Film 116	8	19	125,8	0,16	0,03	541	0,01	652
Film 117	8	19	76,86	0,10	0,03	913	>0,01	1019
Film 118	8	19	93,1	0,04	0,02	853	0,01	926
Film 119	8	18	70,8	0,08	0,02	790	0,04	808

Как видно из результатов, алгоритм показал себя неплохо в сравнении с другими методами. Главными достоинствами метода моделируемого отжига являются его сравнительная простота в реализации, небольшое потребление памяти и высокая скорость работы.

Список использованных источников:

1. Алгоритм имитации отжига // Онлайн ресурс ru.wikipedia.org.
2. de la Banda, M. G. Solving Talent Scheduling with Dynamic Programming /M.G. de la Banda, P.J. Stuckey, G. Chu // INFORMS Journal on Computing.-2011.-Vol.23, Issue 1.- P. 120-137.
3. Qin, H. An Enhanced Branch-and-bound Algorithm for the Talent Scheduling Problem /H. Qin, Z. Zhang, A. Lim, X. Liang // European Journal of Operational Research. – 2016. - Vol. 250.- P. 412-426.
4. Вернер, Ф. Минимизация суммарного времени обслуживания для системы с двумя приборами и одним сервером /Ф. Вернер, С.А. Кравченко, К. Хасани // Информатика - Минск, 2014.-№ 1(41).-С.15-24.

**СЕКЦИЯ «МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
ТЕХНИЧЕСКИХ СИСТЕМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»**

УДАЛЕННОЕ УПРАВЛЕНИЕ МНОГОКООРДИНАТНОЙ СИСТЕМОЙ ПЕРЕМЕЩЕНИЙ ЧЕРЕЗ ИНТЕРНЕТ

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

Бегун Д. Г. - аспирант

Дайняк И. В. – канд. техн. наук, доцент

Для управления удаленной системой прецизионных координатных перемещений в составе оборудования микроэлектроники необходимо организовать канал передачи данных, который обеспечит надежный обмен сообщениями между двумя сторонами с заданной скоростью. Если стороны расположены на значительном удалении друг от друга, тогда наиболее предпочтительным способом будет использование сети Интернет и стека протоколов TCP/IP.

В случае управления многокоординатной системой такое управление так же имеет важное прикладное значение, востребованным являются удаленное конфигурирование и наладка оборудования, диагностика и мониторинг, непосредственное управление движением. Например, при возникновении проблем с наладкой оборудования либо управлением системой перемещений у пользователя, ему необходимо обращаться к производителю за поддержкой. Так как производитель зачастую не желает раскрывать детали реализации системы, которые являются коммерческой тайной, то может потребоваться командировка специалиста от предприятия производителя. В некоторых отдельных случаях система может сообщить пользователю полезную информацию, например, коды ошибок, информационные сообщения, журнал событий и т.д., на основании которых можно устранить проблему самостоятельно. Однако это не всегда возможно.

При удаленном управлении посредством протокола TCP/IP есть определенные ограничения. Так как это не детерминированный протокол, который может использоваться, в том числе, и в открытых сетях, то достаточно сложно обеспечить стабильную скорость обмена данными между удаленными сторонами без потерь и задержек пакетов, могут появляться обрывы соединения в силу различных объективных причин. Так же в целях безопасности необходимо использовать средства защиты данных, для предотвращения несанкционированного доступа и различного рода атак.

В современном оборудовании и станках ЧПУ для обеспечения управления в реальном времени системы перемещений необходимо выдерживать время управляющего цикла не менее 1мс, а зачастую и больше, так как от этого напрямую зависит производительность и количество выпускаемой продукции. Классические протоколы последовательной передачи данных и сопряжения с управляющим компьютером не дают достаточной скорости для управления системами перемещениями с несколькими степенями свободы (независимых координат), так как они обладают малой пропускной способностью. Поэтому, все контуры управления размещаются в специализированном контроллере системы управления, обеспечивающим непосредственное управление приводами прямого действия [1].

Системы управления шаговыми двигателями прямого действия на базе EtherCAT позволяют обеспечить время управляющего цикла менее 100мкс для 1000 устройств ввода-вывода, с максимальным дрожанием сигнала (jitter) менее 1мкс. Благодаря этому контуры управления положением, скоростью и током могут быть вынесены из контроллера, и управляющие команды на каждом цикле будут отправляться непосредственно с управляющего компьютера. Такой компьютер должен иметь в своем составе процессор с двумя и более ядрами, оперативной памятью не менее 512 мегабайт, и операционную систему реального времени (Unix, Linux, Windows RTX и др.) [2].

В общем случае, если рассматривать удаленное управление такой системой через Интернет либо локальную сеть, то возможны различные варианты конфигурации:

1. Все контуры управления располагаются на локальной машине. Удаленный клиент может лишь загружать программу и инициировать запуск движения, а также отправлять другие команды высокого уровня. При этом непосредственное управление движением будет производиться локальным компьютером.

2. Контур управления по положению располагается на удаленной машине, которая может отправлять команды на перемещение в заданную точку на локальный компьютер. На локальном компьютере производится расчет скорости и значения управляющих токов, а затем полученные параметры передаются в контроллер управления. Каждый цикл инициируется удаленной машиной.

3. Контуры управления скоростью и положением располагаются на удаленной машине. На каждом управляющем цикле параметры отправляются на локальную машину, которая перенаправляет их непосредственно контроллеру системы перемещений.

Наиболее быстрым и предпочтительным является первый из вариантов управления. Так как непосредственное управление осуществляется с локального компьютера, и управляющий цикл всегда занимает один и тот же интервал времени. Часть функций системы, которые не связаны с непосредственным управлением движением могут быть организованы удаленно, через веб-приложение в интернет браузере либо программу-клиент. Например, команды для запуска и остановки двигателей, наладки системы, тестирования и др., которые не требуется выполнять в реальном времени, могут отправляться также удаленно.

Второй способ управления подразумевает, что положение двигателей напрямую зависит от удаленно отправляемых команд. Каждая команда указывает в какую точку должен выйти шаговый двигатель на этом цикле. После получения команды локальный компьютер дополнительно производит расчет скорости и значения управляющих токов, только после этого все параметры отправляются на выполнение приводам. В

случае многокоординатной системы, команда содержит координаты всех шаговых двигателей, на которые они должны выйти на заданном цикле. Третий способ очень похож на второй, отличие лишь в том, что параметры скорости и координаты на каждом цикле вычисляются на удаленном ПК. В такой конфигурации производительность системы перемещений будет напрямую зависеть от качества связи сети, включающего в себя следующие параметры:

- Полоса пропускания, описывающая номинальную пропускную способность среды передачи информации и определяющая ширину канала (измеряется в бит/с, кбит/с, мбит/с, гбит/с);
- Задержка при передаче пакета (в миллисекундах);
- Колебания (дрожание) задержки при передаче пакетов или джиттер;
- Потеря пакетов. Определяет количество пакетов, потерянных в сети во время передачи.

В сети Интернет применяются две основных модели качества обслуживания QoS (Quality of Service). Это интегрированный сервис InetServ и дифференцированное обслуживание DiffServ. Эти механизмы, как правило, используются рядом сетевых приложений, таких как потоковые-мультимедиа, VoIP и видеоконференции, а также приложения, требующие гарантированного качества надежности. Первый основан на резервировании ресурсов маршрутизаторов на основе протокола RSVP (Resource reSerVation Protocol). Второй механизм использует в составе IP-пакета специального байта ToS (Type of Service), который содержит набор критериев, определяющих тип обслуживания. Он позволяет приоритизировать IP-трафик на сетевых маршрутизаторах, с целью обеспечения высокого качества передачи данных [3].

Согласно RFC 1633, модель интегрированного обслуживания обеспечивает сквозное (End-to-End) качество обслуживания, гарантируя необходимую пропускную способность. IntServ использует для своих целей протокол резервирования сетевых ресурсов RSVP, который обеспечивает выполнение требований ко всем промежуточным узлам. В отношении IntServ часто используется термин «резервирование ресурсов». Узел-источник до передачи данных, требующих определённого нестандартного качества обслуживания (например, постоянной полосы пропускания для передачи видеoinформации), посылает по сети специальное сообщение в формате протокола RSVP. Это сообщение о пути содержит данные о типе передаваемой информации и требуемой пропускной способности. Оно передаётся между маршрутизаторами по всему пути от узла-отправителя до адреса назначения резервируется необходимая пропускная способность с целью обеспечения необходимо зарезервировать определённую полосу пропускания.

Маршрутизатор, получив такое сообщение, проверяет свои ресурсы с целью определения возможности выделения требуемой пропускной способности. При её отсутствии маршрутизатор запрос отвергает. Если требуемая пропускная способность достижима, то маршрутизатор настраивает алгоритм обработки пакетов таким образом, чтобы указанному потоку всегда предоставлялась требуемая пропускная способность, а затем передаёт сообщение следующему маршрутизатору вдоль пути. В результате, по всему пути от узла-отправителя до адреса назначения резервируется необходимая пропускная способность с целью обеспечения запрашиваемого качества обслуживания.

Дифференцированное обслуживание описано в RFC 2474 и RFC 2475. Обеспечивает QoS на основе распределения ресурсов в ядре сети и определенных классификаторов, и ограничений на границе сети, комбинируемых с целью предоставления требуемых услуг. В этой модели вводится разделение трафика по классам, для каждого из которых определяется свой уровень QoS. DiffServ состоит из управления формированием трафика (классификация пакетов, маркировка, управление интенсивностью) и управления политикой (распределение ресурсов, политика отбрасывания пакетов). DiffServ является наиболее подходящим примером «умного» управления приоритетом трафика. В протоколе IPv6 вместо байта ToS используется байт Traffic Class [3].

Таким образом, были рассмотрены возможные варианты реализации удаленного управления многокоординатной системой перемещений через локальную сеть и сеть Интернет. Дана их сравнительная характеристика. В случае управления через Интернет, важным является качество связи сети QoS. Совершенствование сетевого оборудования и технологий, обеспечивающих высокую надежность передачи данных с заданными параметрами, создает условия для применения сети Интернет для управления многокоординатными системами перемещений.

Список использованных источников:

1. Жарский В.В. Системы многокоординатных перемещений и исполнительные механизмы для прецизионного технологического оборудования : моногр. / В. В. Жарский, С. Е. Карпович, И. В. Дайняк [и др.] ; под. ред. д-ра техн. наук, проф. С. Е. Карповича. - Минск : Бестпринт, 2013. - 208 с.
2. EtherCAT Technology Group: EtherCAT – шина Ethernet. [Электронный ресурс]. Режим доступа <http://www.ethercat.org/en/technology.html>. – Дата доступа: 20.12.2016.
3. Sziget T. End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks, 2nd Edition // Sziget T., Hattingh C., Barton R., Briley K. - Cisco Press, 2013. – 1090с.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РЕШЕНИЯ ПРЯМОЙ ЗАДАЧИ КИНЕМАТИКИ МАНИПУЛЯТОРА НА ГИБРИДНОМ ЛИНЕЙНОМ ПРИВОДЕ

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

Войтов А. Ю.

Дайняк И. В. – кандидат техн. наук, доцент

Реализация необходимого перемещения и ориентации исполнительного элемента, связанного с инструментом или заготовкой автоматизированного оборудования, характеризуемого подвижностью до шести степеней свободы включительно, наиболее эффективно осуществляется при использовании реконфигурируемых механизмов параллельной кинематики на гибридных многокоординатных приводах прямого действия. В сравнении с традиционными промышленными роботами и манипуляторами, имеющими разомкнутую структуру, они отличаются высокой структурной жёсткостью, модульностью конструкции и позволяют добиваться высокой динамической точности за счёт устойчивости к вибрациям.

В настоящей работе рассматривается параллельный манипулятор с шестью степенями свободы на гибридном приводе прямого действия, содержащем жёсткое треугольное основание с тремя направляющими для шести линейных подвижных координатных электромагнитных модулей. Структурно-кинематическая схема параллельного манипулятора приведена на рис. 1. Особенностью схемы является предложенная конфигурация гибридного привода со спаренными координатными модулями на каждой из трёх направляющих, линейные перемещения которых $s_i (i=1,2,\dots,6)$, как задаваемые функции положения ведущих звеньев исполнительного механизма параллельной кинематики, преобразуются в шесть независимых между собой координатных функций положения подвижного исполнительного элемента, треугольной платформы ABC , включая три линейных $x_{O_1}, y_{O_1}, z_{O_1}$ и три угловых φ, θ, ψ [1]. Структурная схема шестикоординатного линейного привода на треугольной направляющей представлена на рис. 2.

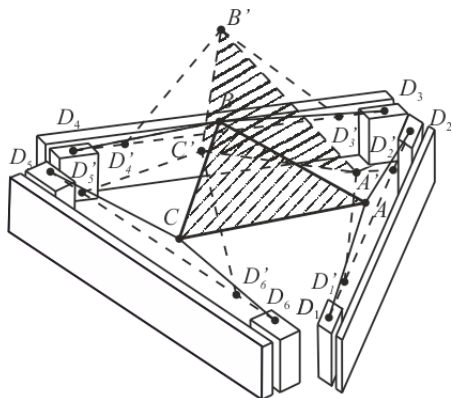


Рис. 1 - Структурно-кинематическая схема параллельного манипулятора

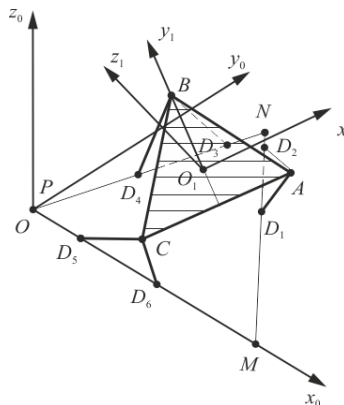


Рис. 2 - Структурная схема линейного привода на треугольной направляющей

Прямая задача кинематики для рассматриваемого манипулятора состоит в определении координат положения и ориентации подвижной платформы ABC (рис. 1), представляемых тремя линейными $x_{O_1}, y_{O_1}, z_{O_1}$ и тремя угловыми φ, θ, ψ функциями положения относительно неподвижной системы координат. Формальная постановка прямой задачи состоит в том, что предполагаются заданными фиксированные положения точек M, N и P вершин треугольного гибридного привода, определяющих направляющие MN, NP и PM для подвижных точек $D_i (i=1,2,\dots,6)$ управляемых модулей, положение которых задаётся управляющими функциями положения. Необходимо определить функции положения и ориентации платформы ABC в зависимости от управляющих функций и топологии механизма параллельной кинематики.

С учётом того, что плоскость треугольника MNP совмещена с координатной плоскостью $x_0 O y_0$ (рис. 2) получим координатное представление этих точек в виде: $M(x_M, y_M, 0); N(x_N, y_N, 0); P(x_P, y_P, 0)$. Не нарушая общности решения прямой задачи примем линейные размеры сторон треугольника ABC равными между собой и равными a . Длины промежуточных звеньев, шатунов механизма параллельной кинематики, примем тоже равными между собой и равными l , то есть $D_1A=D_2A=D_3B=D_4B=D_5C=D_6C=l$. По этим исходным данным будем решать позиционную прямую задачу кинематики, заключающуюся в нашем случае в нахождении положения точек A, B и C подвижной треугольной платформы ABC , выраженных в неподвижной системе координат $S_0(x_0, y_0, z_0)$.

Для решения поставленной задачи вначале необходимо получить аналитическое параметрическое

задание положения шести подвижных линейных модулей, характеризуемых положением точек $D_i(i=1,2,\dots,6)$ в системе координат S_0 . Для этого в системе координат S_0 зададим направляющие векторы $\vec{q}_1, \vec{q}_2, \vec{q}_3$ перемещений спаренных подвижных линейных модулей по направляющим MN, NP и PM .

На основании условия коллинеарности соответствующей направляющей и вектора перемещения, расположенного на ней линейного подвижного модуля (рис. 2) окончательно получим параметрическое представление положения точек $D_i(i=1,2,\dots,6)$ в виде:

$$\begin{aligned}x_{D_1} &= \lambda_1(x_N - x_M) + x_M; y_{D_1} = \lambda_1(y_N - y_M) + y_M; \\x_{D_2} &= \lambda_2(x_N - x_M) + x_M; y_{D_2} = \lambda_2(y_N - y_M) + y_M; \\x_{D_3} &= \lambda_3(x_P - x_N) + x_N; y_{D_3} = \lambda_3(y_P - y_N) + y_N; \\x_{D_4} &= \lambda_4(x_P - x_N) + x_N; y_{D_4} = \lambda_4(y_P - y_N) + y_N; \\x_{D_5} &= \lambda_5(x_M - x_P) + x_P; y_{D_5} = \lambda_5(y_M - y_P) + y_P; \\x_{D_6} &= \lambda_6(x_M - x_P) + x_P; y_{D_6} = \lambda_6(y_M - y_P) + y_P,\end{aligned}\tag{1}$$

где $\lambda_i(i=1,2,\dots,6)$ - параметры, определяющие положения соответствующих точек $D_i(i=1,2,\dots,6)$ на направляющих треугольного гибридного привода.

Для формирования математической модели решения прямой задачи кинематики [2], через переменный угол α_1 , для рассматриваемой в статье системы перемещений рассмотрим фрагмент исполнительного механизма D_1AD_2 .

Рассмотрим вывод базовой системы из трёх уравнений, анализируя параллельный фрагмент D_1AD_2 .

Для этого для вектора \vec{q}_1 найдём его орт \vec{q}_1^0 по выражению:

$$\vec{q}_1^0 = (m_1, n_1) = \left(\frac{x_N - x_M}{MN}, \frac{y_N - y_M}{MN}, 0 \right), \text{ где } MN = \sqrt{(x_N - x_M)^2 + (y_N - y_M)^2}.\tag{2}$$

С учётом найденных ортов по геометрической модели окончательно получим искомые координаты точки A через обобщённую координату α_1 :

$$\begin{aligned}x_A &= \frac{1}{2}(x_{D_1} + x_{D_2}) - n_1 AT_1 \cos \alpha_1; \\y_A &= \frac{1}{2}(y_{D_1} + y_{D_2}) + m_1 AT_1 \cos \alpha_1; \\z_A &= AT_1 \sin \alpha_1,\end{aligned}\tag{3}$$

где $AT_1 = \sqrt{AD_1^2 - \frac{1}{4}((x_{D_1} - x_{D_2})^2 + (y_{D_1} - y_{D_2})^2)}$.

Аналогичные расчётные модели могут быть использованы при алгоритмизации двух других фрагментов исполнительного механизма D_3BD_4 и D_5CD_6 , которые являются подобными по кинематике фрагменту D_1AD_2 . В результате математические модели формирования обобщённых координат α_1 и α_2 будут отличаться только на уровне обозначений используемых переменных. В первом случае вместо α_1 будем использовать α_2 , а во втором – α_3 .

Расчётные координаты точек A, B и C из приведенных расчётных формул позволяют получить систему нелинейных уравнений относительно неизвестных α_1, α_2 и α_3 в виде формулы (4).

$$\begin{cases} (x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2 = a^2; \\ (x_C - x_B)^2 + (y_C - y_B)^2 + (z_C - z_B)^2 = a^2; \\ (x_A - x_C)^2 + (y_A - y_C)^2 + (z_A - z_C)^2 = a^2. \end{cases}\tag{4}$$

В результате решения полученной системы (4) находим текущие значения углов α_1, α_2 и α_3 однозначно соответствующие текущим положениям точек $D_i(i=1,2,\dots,6)$. Подставляя найденные значения α_1, α_2 и α_3 в расчётные формулы для нахождения координат точек A, B и C , а значит получим текущее положение и ориентацию подвижного треугольного звена ABC в трёхмерном пространстве.

Последующее компьютерное моделирование [3] позволит получить необходимые характеристики конкретной системы перемещений, включающие геометрию рабочей области, калибровку траекторий по скорости и точности, выбор оптимальных траекторий в зависимости от характера технологической операции.

Список использованных источников:

1. Системы многокоординатных перемещений и исполнительные механизмы для прецизионного технологического оборудования / В.В. Жарский [и др.]; под ред. д-ра техн. наук, проф. С.Е. Карповича. – Минск : Бестпринт, 2013. – 208 с.
2. Прикладные задачи по высшей математике: Аналитическая геометрия : учеб.-метод. пособие / С.Е. Карпович [и др.]. – Минск : БГУИР, 2013. – 64 с.
3. Моделирование механизмов параллельной кинематики в среде MATLAB/Simulink / С.Е. Карпович [и др.]. – Минск : Бестпринт, 2013. – 153 с.

СЕКЦИЯ «ИНФОРМАТИКА»

APPLICATION OF IBM WATSON FOR THE BEST SPECIALITY OPTION PREDICTION FOR PROSPECTIVE STUDENTS

*Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus*

Victor Kozub, Alexander Alexandrov

Ivan Piletski – associate professor

This paper investigates a problem of choosing a speciality by prospective students. Every year, it's a difficult problem for school graduates to choose a speciality that's the most suitable for them. Furthermore, there is a diversity of specialities and faculties at one university to choose from. Often when choosing a speciality, prospective students don't take into account their own personal characteristics. We've developed an application in IBM Bluemix cloud that predicts the best speciality option for prospective students of our university. The application uses IBM Watson service to get psychological characteristics of a person and then to predict a proper speciality. In this paper, we describe the IBM Bluemix cloud platform as a service, tools and philosophy of it by the example of the developed application. Obtained results are planned to be extended to satisfy business needs and to search for a staff needed for a particular project at a company.

The object recognition task can be solved using different methods. It depends on structures and algorithms that are used to store data. Simple example is when data is stored in a relational database: we can answer if a new object is known by simply accessing the database. Similarly, when objects are stored as elements of a set with known properties, enumeration method can be used to find whether an object is within the set or not. When it is impossible to compare a new object with a known one, new key characteristics of the object are picked. Then, these key characteristics are used by deep search algorithms to make a prediction with high probability rate. For example, car properties can be predicted by using a photo of the car.

But the task of making a personality portrait of a particular person is much more difficult than tasks above. That is because every person is different, and its personality portrait can be examined through its behavior and actions as well as its writings. It is practically impossible to create a database of personality portraits for entire human population in advance. Thus, methodology of incremental development of such databases for small groups of people needs to be developed. This task is especially important when applying for a job (e.g., pilot) or when choosing a speciality (e.g., mathematician) or a university. In this case, a personality portraits for a certain job position, speciality, university need to be created. Remember that it is more difficult to create an organization personality portrait than a human personality portrait. When a person needs to choose a speciality, more informed choice decision can be done by comparing these portraits.

In the last few years cognitive systems have had a boost in development. Now it is possible to use cognitive computing to solve problems of this kind.

There were psychological research in USA that have proven connection between one's writing style, word order, emotional tone and personality traits [1]. A prediction about one's predisposition to certain types of activities can be made based on particular personality traits. The research results were practically applied to a development of cognitive systems. Cognitive system can simulate person's perception system that was established in his mind during a process of development a character, education, learning, observation and reflection of the world.

Cognitive modelling allows us to find the best management decisions or case scenarios based on concepts that describe the evolving situation both qualitatively and quantitatively. Interference of different factors is also taken into account. Cognitive modelling can be done within companies for a purpose of supporting executives and managers who make decisions concerning future development of a company.

IBM Watson is one of the first cognitive systems in the world. It has many functions and also can understand and work with natural language. It uses large amounts of global unstructured data (Big Data). IBM Watson is a technology platform that uses natural language processing and machine learning to reveal insights from large amounts of unstructured data [2]. Today IBM Watson is available within a new cloud platform called IBM Bluemix that has introduced last year.

IBM Bluemix is an open-standards cloud platform for building, running, and managing applications. With Bluemix, developers can focus on building excellent user experiences with flexible compute options, choice of DevOps tooling, and a powerful set of IBM and third-party APIs and services [3].

There is a service available in Bluemix that allows developer to build personality portraits. It is called Personality Insights. The Personality Insights service uses linguistic analytics to infer individuals' intrinsic personality characteristics from digital communications that they make available via media such as email, text messages, tweets, and forum posts.

Personality Insights analytics are developed based on the psychology of language in combination with data analytics algorithms. The characteristics are described in terms of three models [1]: Big Five, Needs, Values.

One of the main goals of this work is to explore possibilities provided by IBM Watson and its application in everyday life. We have decided to create an application to help school graduates choose a speciality. Using Personality Insights service, the application gets personality profiles of a prospective student and specialities in our university and then compares them. Furthermore, it is possible to modify the application and use it in different IT companies when looking for candidates to a specific job.

The majority of school graduates don't know what they want to do in life because they unaware about how their interests and hobbies can be applied in life and what universities can offer to them. There are different psychological

tests that they can take in order to identify predisposition for a particular job. But it is possible to cheat on these tests, aware or unaware of it. Cognitive modelling allows building the most accurate personality portrait based on just writings. This approach takes into account word order, emotional tone and other characteristics of the text to make a prediction about person who wrote it.

We have developed an application called 'Choice of a speciality at a university', or simply 'Speciality helper'. It works as follows:

- User loads his writing (or provides his Twitter-ID);
- Application connects to IBM Watson services and gets user's personality profile;
- Obtained user profile is being compared to personality profiles of specialities at our university in order to find the best match.

The result is a prediction about the most suitable speciality to study. When comparing very different profiles it is possible to inform a user to consider other universities.

The method of obtaining specialities personality profiles requires clarification. There was no significant difference between specialities because of the texts we have used to get their profiles. Due to analysis of Personality Insights service, we have discovered two significant characteristics that affect the results:

- The text that is sent to the service should be written by user himself;
- The text should have some emotional tone, express the user's position and his opinion regarding the object or phenomenon described in it.

Now, when we know what kind of text we need, we have struggled with another problem: how to obtain speciality personal profile because speciality is not a person. Then we came to the assumption that the text about particular speciality should be written by graduates of this speciality. Speciality is a collective image of its graduates. We assume that there is minimum number of reviews about speciality that we can summarize in order to get its personality profile.

Thus, the main problem we are working on now is collecting and summarizing specialities reviews in order to create sufficient personality portraits. That is, we are developing optimal methodology of creating collective personality profiles.

Now we are working on methods and application that can be applied in other areas of human life. E.g., it is possible to create personality portrait of a desired candidate for a certain job position in company using collective personality portrait of successful employees of this particular position.

The application under development is based on an IT company template and will be used to upgrade the Talent Constructor application [4]. Talent Constructor is a packaged software product for effective management of organizational capacity.

In IT company template, the structure of a company is represented by a tree view, which starting node contains information about company at all, and child nodes contain information about departments. The edges show the structural subordination of the departments. Each node has associated competences: general, professional, leadership. The following entities form the basis of this approach. Employee (personality portrait, professional characteristics) – information is extracted from projects, emails, social networks. Candidate (requirements for a specific project) – information is extracted from projects, emails, social networks. Customer – information is extracted from documentation, emails, social networks, activity history. Project (completed projects, their characteristics, possible projects). Vacancy (requirements for a specific department).

The application compares a concrete entity profile with an abstract profile required by organization. Some reports are being formed.

The most effective administrative decision can be made using the results of comparison. Thus, company executives can manage the company more effectively.

As you can see, general solution of the problem can be effectively applied in different businesses. The comparison of personality profiles allows us to make a more informed decision, whether we are choosing a speciality to study or a company to work for.

The methodology and technology of creating a personality portrait of a person or speciality has been developed. This methodology has been tested using an application that helps prospective students choose a speciality. The application is written in Python using Django framework. It uses IBM Watson Personality Insights service available in IBM Bluemix cloud platform. Personality Insights service accepts text which should have at least three thousand words and returns so called 'Big Five' personality model. When we have person's personality portrait and speciality profile the comparison can be made. In developed application we use Euclidean distance to find suitable speciality.

References:

1. IBM Watson Developer Cloud (No date) *The science behind the Personality Insights service* [Online] IBM Watson Developer Cloud. Available: <http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/science.shtml> [Accessed 21 March 2016]
2. IBM Watson (No date) *What is Watson* [Online] IBM Watson. Available: <http://www.ibm.com/smarterplanet/us/en/ibmwatson/what-is-watson.html> [Accessed 21 March 2016]
3. I.Piletski et al. Virtual IT-environment for Big Data and VCL research: BIG DATA and Predictive Analytics. Conference proceedings, pp. 21-32. Minsk, BSUIR (2015). / И.И. Пилецкий и др. Виртуальная ИТ среда БГУИР для исследования Big Data и VCL, стр. 21-32, BIG DATA and Predictive Analytics. Использование BIG DATA для оптимизации бизнеса и информационных технологий : сборник материалов междунар. науч.-практ. конф. / редкол. : М. П. Батура [и др.]. – Минск : БГУИР, 2015. – 220 с.
4. IBA Group (No date) *Talent Constructor* [Online] IBA Group. Available: <http://talentconstructor.by/> [Accessed 21 March 2016]

ПРОБЛЕМА ДОСТИЖЕНИЯ КОНСЕНСУСА В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

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

Янковский А. Н.

Волорова Н. А. – канд. техн. наук, доцент

Одной из главных целей распределенных вычислений является достижение надежности всей системы в целом при наличии сбоев в некоторых ее компонентах. Это зачастую требует от участников согласия относительно некоторого значения, необходимого во время вычислений, – т.е. достижения консенсуса. В общем случае, некоторые из участников могут быть испорченными (например, взломанные сервера, неисправные сенсоры) и пытаться нарушить работоспособность системы и препятствовать достижению консенсуса среди других участников.

Несмотря на то, что разработка распределенных систем ведется уже не первое десятилетие, в этой области до сих пор остается множество открытых вопросов. Не упрощает дело и тот факт, что с каждым годом требования к надежности и производительности подобных систем растут, равно как и объемы обрабатываемых данных. Однако перед тем, как вести речь о проблемах и вариантах их решения, необходимо сначала определиться с терминологией и дать определения ключевым понятиям.

Консенсус – это задача получения согласия всех участников о некотором конкретном значении на основе их голосов. Все участники должны должны признать одно и то же значение, и оно должно быть предложено одним из участников (т.е. алгоритм достижения консенсуса не может взять значения из ниоткуда). В самом простом случае, значение может быть бинарным (0 или 1), что может использоваться участниками для принятия решение вроде “делать что-либо или нет”. Примеры ситуаций, когда консенсус оказывается полезным:

- Синхронизация реплицированных конечных автоматов и обеспечение гарантии, что все реплики имеют одинаковое (консистентное) представление состояния системы
- Выборы лидера (например, для взаимного исключения)
- Распределенное журналирование с глобальной последовательной согласованностью

В идеальном мире консенсуса достичь легко. Например, обсуждая алгоритмы распределенного взаимного исключения, мы приходим к одной из форм консенсуса, когда все принимают одно и то же решение касательно того, кто должен получить доступ к ресурсу. Наиболее простая реализация – это назначить общего координатора, которые будет отвечать за определение исхода. Двухфазный коммит также является примером системы, когда мы полагаем, что координатор и участники функционируют и могут общаться. Но подвох в этих алгоритмах заключается в том, что все участники должны быть работоспособны. Сбои все усложняют. Под сбоями понимаются отказы участников и сетевые ошибки.

Невозможно гарантированно достичь консенсуса при наличии сбоев у полностью асинхронных участников. Это означает отсутствие каких-либо предположений касательно скоростей участником или сетевых коммуникаций. Суть проблемы заключается в том, что невозможно проверить, сломался ли участник или же участник в порядке, но сообщения доходят до него неприемлемо медленно. Это было доказано Фишером, Динчем и Паттерсоном. Также, консенсус невозможен в условиях ненадежности коммуникаций, т.к. мы можем не иметь возможности общаться с участником.

Сеть никогда не является 100% надежной и это необходимо учитывать. В формальных доказательствах распределенных систем обычно полагается, что сеть является асинхронной, т.е. сообщения между участниками могут дублироваться, теряться, задерживаться или перемешиваться произвольным образом. Это довольно слабое предположение: некоторые физические сети могут предоставить большие гарантии, но на практике в IP сетях встречаются все эти виды сбоев, поэтому теоретические ограничения асинхронных сетей применяются и к реальным системам.

При возникновении сетевых разделов многие системы переходят в специальный режим с ограниченной функциональностью. CAP теорема гласит, что мы можем обеспечить либо согласованность (технически, линейную линейность на операции чтения-записи), либо доступность (все сервера продолжают обрабатывать запросы), но не одновременно. Что примечательно, немногие базы данных достигают теоретических ограничений CAP, многие просто теряют данные.

Другой вид сбоев в распределенных системах – это ошибки участников (но не fail-stop) в условиях надежной сети. В таких ситуациях сервера продолжают работать, но вместо того, чтобы молчать, они посылают неправильные данные. Эти ошибки называются византийскими в честь известной Проблемы Византийских генералов. В этой проблеме есть n генералов, которые управляют разными отрядами. Коммуникации надежны, однако m генералов – предатели и стараются помешать остальным прийти к согласию, посылая им некорректную информацию. Вопрос заключается в следующем: могут ли верные генералы достичь консенсуса?

Лэмпорт продемонстрировал решение, которое работает в некоторых случаях. Согласно ему, любое решение задачи с m предателями требует как минимум $3m+1$ участников (т.е. $2m+1$ верных генералов). Это означает, что более $\frac{2}{3}$ генералов должны быть верными. Более того, было доказано, что один алгоритм не может справиться с m сбоями менее, чем за $m+1$ раундов обмена сообщениями и $O(m^2)$ самих сообщений.

Если $n < 3m+1$, то задача не имеет решения. Очевидно, что это довольно дорогое решение. Существует вариация Проблемы Византийских генералов, в которой используются подписанные сообщения. Это означает, что сообщения от верных генералов не могут быть подделаны или модифицированы. Для таких случаев существуют алгоритмы, позволяющие достичь консенсуса при условиях $n \geq m + 2$.

Архитектура отказоустойчивых распределенных систем очень сложна. Существует множество различных проблем и ошибок, которые могут возникать в условиях распределенности и система должны уметь их обрабатывать. Корректно обрабатывать ошибки – наиболее сложная задача. На практике большинство ошибок вызваны проблемами с сетью, а византийские модели редко используются в силу своей сложности. Существуют готовые алгоритмы для решения проблемы консенсуса, однако они часто полагаются на различные допущения и поэтому не всегда могут быть использованы на практике без дополнительных модификаций.

Список использованных источников

1. Michael J. Fischer, Nancy A. Lynch and Michael S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process," Journal of the ACM, April 1985, 32(2):374-382. <https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf>
2. D. Dolev. "The Byzantine Generals Strike Again," Journal of Algorithms, 3, 1982, pp. 14-30.
3. L. Lamport. 1983. The Weak Byzantine Generals Problem. J. ACM 30, 3 (July 1983), 668-676. DOI=<http://dx.doi.org/10.1145/2402.322398>
4. Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. 1985. Easy impossibility proofs for distributed consensus problems. In Proceedings of the fourth annual ACM symposium on Principles of distributed computing (PODC '85), Michael Malcolm and Ray Strong (Eds.). ACM, New York, NY, USA, 59-70. DOI=<http://dx.doi.org/10.1145/323596.323602>

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ КОГНИТИВНЫХ СЕРВИСОВ IBM WATSON, ДОСТУПНЫХ В РАМКАХ ОБЛАЧНОЙ ПЛАТФОРМЫ IBM BLUEMIX

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

Александр Александров, Виктор Козуб

Иван Пилецкий, канд. физ-мат. наук, доцент

IBM Watson представляет собой когнитивную систему, которая способна понимать, делать выводы и обучаться. Она также позволяет преобразовывать целые отрасли, различные направления науки и техники. Например, предсказывать появление эпидемий или возникновения очагов природных катастроф в различных регионах, вести мониторинг состояния атмосферы больших городов, оптимизировать бизнес-процессы, узнавать, какие товары будут в тренде в ближайшее время. Доступ к части сервисов IBM Watson доступен через использование API облачной платформы IBM Bluemix. О некоторых возможностях IBM Watson и пойдет речь в данной работе.

IBM Watson — одна из первых когнитивных систем в мире. Эта система умеет очень многое, благодаря чему возможности Watson используются во многих сферах — от кулинарии до предсказания аварий в населенных пунктах. В целом-то, большинство возможностей Watson не являются чем-то уникальным, но в комплексе все эти возможности представляют собой весьма мощный инструмент для решения разнообразных вопросов. Например — распознавание естественного языка, динамическое обучение системы, построение и оценка гипотез [1]. Все это позволило IBM Watson научиться давать прямые корректные ответы (с высокой степенью достоверности) на вопросы оператора. При этом когнитивная система умеет использовать для работы большие массивы глобальных неструктурированных данных, Big Data. Получить доступ к возможностям Watson можно используя облачную платформу IBM Bluemix.

В США были проведены психологические исследования, которые подтвердили, что стиль письма человека, порядок слов, эмоциональный окрас напрямую связаны с чертами характера данного человека [2]. Предсказание к предрасположенности к определенным видам деятельности может быть сделано на основании анализа определенных черт характера данного человека. Результаты таких исследований напрямую легли в разработку алгоритмов когнитивных систем.

Одним из сервисов IBM Watson является сервис, позволяющий получить портрет личности человека, основываясь на лингвистическом анализе сообщений из соцсетей, электронной почты или любых других источников текста, который написал данный человек. Этот сервис носит название Personality Insights. Personality Insights базируется на психологии языка вкупе с алгоритмами обработки данных. Выходными данными данного сервиса является представление черт характера исследуемой личности в трёх моделях: модель Большой пятёрки (Big Five), модель потребностей (Needs) и модель ценностей (Values). В рамках данной работы мы исследовали возможности IBM Watson для применения в повседневной жизни. Одну из задач, которые мы попробовали решить, была задача выбора специальности для абитуриента, основываясь на портрете личности данного человека.

Для получения портрета личности, пользователь должен ввести текст о себе, либо предоставить свой

Twitter-ID. Текст должен быть написан непосредственно пользователем и иметь эмоциональный окрас. Например, пользователь в данном тексте может рассказать о себе, своей жизни, приятных моментах из детства либо своём взгляде на какую-либо проблему. Далее данный текст через API отправляется на обработку в IBM Watson, в ответ мы получаем портрет личности, представленный численно в трёх моделях. Полученные значения сверяются с эталонными для специальностей и в качестве ответа выбирается специальность, значения которой оказываются наиболее «близко» от значений пользователя. Самой большой проблемой является выбор этих самых эталонных значений для специальностей. В данный момент используются тексты с описанием специальностей, представленные на сайте университета и пропущенные также через Personality Insights. Выбор этих текстов нельзя назвать оптимальным. Наиболее лучшим вариантом был бы преданализ текстов выпускников данных специальностей, и не просто выпускников – а тех, что действительно осознанно сам выбрал эту специальность и считает, что он попал на учёбу в нужное место. Далее, проанализировав данные тексты, можно будет получить более точную оценку эталонных значений.

Нами было написано приложение, реализующее данный функционал с использованием когнитивных возможностей IBM Watson, развёрнутое на базе облачной платформы IBM Bluemix. Приложение было написано и успешно представлено на хакатоне IBM Bluemix Hackathon Moscow в 2015-м году, где вошло в тройку лучших проектов.

Список литературы:

1. IBM Watson. *What is Watson* [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.ibm.com/smarterplanet/us/en/ibmwatson/what-is-watson.html>
2. IBM Watson Developer Cloud. *The science behind the Personality Insights service* [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/science.shtml>

АНАЛИЗ ИСПОЛЬЗОВАНИЯ МЕТОДОВ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ ДЛЯ ЗАЩИТЫ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ

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

Рогов М. Г.

Егорова Н. Г. – доцент, канд. техн. наук

Защита персональных данных – это целый комплекс мероприятий технического, организационного и организационно-технического характера, которые необходимы для защиты информации, относящейся к определенному или определяемому на основании такой информации физическому лицу (субъекту персональных данных) [1].

Проблема защиты данных с использованием криптографических преобразований имеет множество вариантов решения. Для осуществления безопасности имеющихся в наличии данных нужно соблюдать основные правила:

- обеспечивать конфиденциальность информации, передаваемой или хранимой в памяти;
- всегда подтверждать подлинность и целостность информации;
- при каждой установке соединения и входе пользователя в систему производить аутентификацию абонента.

Возможность осуществления какой-либо операции преобразования и изменения информации, которая выполняется одним либо несколькими пользователями системы, владеющими так называемым секретом, не зная которого нет возможности осуществить эту операцию, является основой всех криптографических преобразований информации [2].

Криптографические технологии аутентификации, шифрования и цифровой подписи используются для реализации указанных функций.

К криптографическим методам защиты в общем случае относятся:

- шифрование (дешифрование) информации;
- формирование и проверка цифровой подписи электронных документов.

Чтобы обеспечить конфиденциальность данных используются различные методы симметричного и ассиметричного шифрования, а также взаимная аутентификация на основе одноразовых и многократных паролей, смарт-карт, цифровых сертификатов и многого другого [3].

Современные криптографические системы и методы защиты информации должны обладать рядом общепринятых свойств:

- возможность расшифровать сообщение существует только при наличии;
- владение алгоритмом никоим образом не должно влиять на защищенность информации;
- без изменений остаются структурные элементы алгоритма шифрования;

- изменение самого вида зашифрованного сообщения происходит даже при незначительном изменении ключа шифрования;
- нет понятных зависимостей между ключами, если в процессе шифрования последовательно используются несколько ключей;
- все дополнительные биты, если таковые вводятся в сообщение, хорошо скрыты в зашифрованном сообщении;
- любой ключ из множества возможных способен обеспечить надежную защиту информации.

В ходе работы было изучено множество методов шифрования, используемых как в информационных системах, так и в повседневной жизни. Также были изучены различные особенности защиты информационных систем. Был проведен анализ стойкости различных методов криптографических преобразований. Выявленные в результате данного анализа положительные и отрицательные особенности различных методов криптографических преобразований позволили выделить наиболее верные пути защиты информации. Также были выявлены особенности некоторых методов защиты информации с точки зрения удобства для рядового пользователя.

Список использованных источников:

1. Скембрей Дж. Секреты хакеров / Дж. Скембрей, Ст. Мак-Клар – Москва: «Вильямс», 2004. – 512 с.
2. Сمارт Н. Сурптоgraphy: An Introduction / Н. Сمارт – Москва: «Техносфера», 2006. – 528 с.
3. Сингх С. Книга шифров / С. Сингх – Москва: «Астрель», 2006. – 447 с.

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА ВЫСОКОНАГРУЖЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ JAVA

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

Вакульчик Е. Н.

Куликов С. С. – к.т.н., доцент

Современным веб-приложениям приходится обрабатывать сотни и даже тысячи запросов в секунду. Чтобы обеспечить необходимую производительность, необходимо определить потенциальные узкие места и предпринять меры по их устранению. Стадии оптимизации производительности должна предшествовать стадия тщательного исследования и анализа.

В настоящее время существуют две конкурирующие платформы для разработки веб-приложений одинаковой сложности, использующих в качестве основы язык Java: Java EE и Spring. Целью данного исследования являлась разработка двух веб-приложений на платформах Java EE и Spring соответственно, осуществляющих запись и выборку данных из БД, и сравнение их производительности. В качестве СУБД использовался MySQL, а в качестве сервера приложений – GlassFish. Для проведения нагрузочного тестирования и отметки ключевых показателей использовалось средство автоматизации нагрузочного тестирования JMeter.

В связке Spring и Hibernate для эффективного доступа к базе данных использовался пул соединений, представляющий собой кэш для открытых соединений между приложением и СУБД. Использование такого решения позволяет сильно сократить накладные расходы на установку и инициализацию нового соединения между приложением и СУБД.

Нагрузочное тестирование веб-приложений проводилось с помощью варьирования количеством пользователей и регистрируя ключевые показатели. Наиболее важные регистрируемые показатели: пропускная способность (з\с - число запросов в секунду), время реакции, процент загрузки ЦП серверного компьютера, процент загрузки ЦП БД.

Анализируя результаты зависимости з\с и времени ответа от количества пользователей (запись в БД) для Java EE приложения можно сделать вывод о том, что до 750 количества пользователей производительность линейно возрастает, а время отклика остается относительно небольшим. От 750 до 1250 время отклика резко повышается, а производительность достигает своего максимума в точке 1250. Дальнейший рост числа пользователей только уменьшает производительность и значительно увеличивает время отклика.

Анализируя результаты зависимости з\с и времени ответа от количества пользователей (выборка из БД) для Java EE приложения можно сделать вывод, что до 1000-1250 пользователей производительность растет, а время отклика остается небольшим, после этого значения время отклика резко возрастает, а производительность при этом остается практически неизменной.

Проводя анализ зависимости з\с и времени ответа от количества пользователей (запись в БД) для Spring приложения, наблюдается, что до 1500-1750 пользователей время отклика остается на низком уровне, а производительность растет. Далее время отклика резко возрастает, а производительность остается на

относительно неизменном уровне.

Проводя анализ зависимости з\с и времени ответа от количества пользователей (выборка из БД) для Spring приложения видно, как до 1000 пользователей время отклика остается низким, а производительность растет. Но после этого значения время отклика резко возрастает, а производительность падает. Производительность перестает расти после 2000.

Сравнивая результаты для Java EE и Spring при записи данных, можно сделать вывод, что для времени отклика оптимизированный Spring сработал более продуктивно. Но не на всей области сравнения, а только после 100-250 пользователей. При выборке данных Java EE дольше показывал стабильные результаты для времени отклика. Для производительности (з\с) Spring практически сразу стал показывать более высокие результаты.

В результате выполнения данной работы были разработаны веб-приложения в соответствии со спецификацией Java EE и основанное на технологии Spring и библиотеке Hibernate. Создана MySQL база данных, в которую веб-приложения производили запись и выборку данных. При создании приложения в связке Spring и Hibernate были предложены способы оптимизации работы с данными. Был проведен ряд тестов для определения производительности веб-приложений. Была определена точка насыщения – ключевое значение, при котором рост числа клиентов не повышает общей производительности приложения.

Список использованных источников:

1. Тестирование и анализ производительности с помощью сервера приложений WebSphere [Электронный ресурс]. Дата публикации: 13.06.2013. URL: http://www.ibm.com/developerworks/ru/library/wes-1208_hare/
2. Оптимизация и масштабирование Web приложений [Электронный ресурс]. Дата обновления: 04.10.2015. URL: <http://ruhighload.com/>
3. Spring MVC 3, Аннотации Hibernate, MySQL. Хабрахабр [Электронный ресурс]. Дата публикации: 23.01.2015. URL: <http://habrahabr.ru/post/248541/>

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЙ

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

Грученков В. В.

Новиков В. И. – канд. техн. наук, доцент

Растущий спрос на гибкость и масштабируемость для удовлетворения быстро растущих потребностей бизнеса создает потребность в применении новых подходов к проектированию архитектуры веб-приложений. В докладе рассматривается микросервисная архитектура, анализируются ее достоинства и недостатки.

Большинство современных Web-приложений включают три основные части: пользовательский интерфейс, база данных и сервер. Серверная часть обрабатывает HTTP запросы, запрашивает и обновляет данные в БД, заполняет HTML страницы, которые затем отправляются браузеру клиента. В подобных системах вся логика по обработке запросов выполняется в единственном процессе, любое изменение в системе приводит к пересборке и развертыванию новой версии серверной части приложения, а масштабирование осуществляется путем запуска дополнительных серверов, либо обновлением конфигурации существующих.

Решением вышеописанных проблем является использование микросервисной архитектуры. Микросервисная архитектура – это стиль построения сложных систем, предлагающих разбиение приложения на мелкие сервисы, каждый из которых работает в собственном процессе и взаимодействует с остальными используя легковесные механизмы, как правило протокол HTTP. Идея подобной архитектуры, однако, не является революционной - ее корни уходят далеко в прошлое, как минимум к принципам проектирования, использованным в системах Unix.

На рисунке 1 приведена приложение с микросервисной архитектурой:

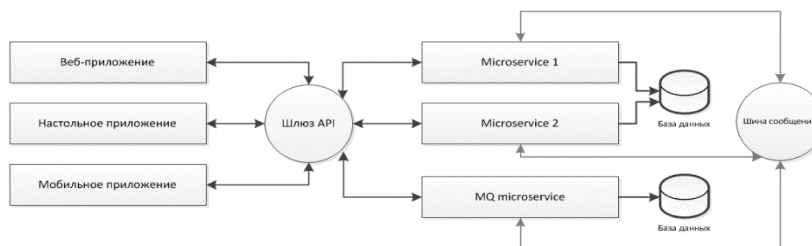


Рис. 1. Структурная схема приложения с микросервисной архитектурой

Преимущества микросервисной архитектуры:

- Техническое разнообразие - то, что каждый компонент изолирован от остальных, дает

- возможность использования наиболее подходящий для каждой задачи язык программирования;
- Горизонтальное масштабирование и отказоустойчивость – выполнение каждого сервиса в отдельном процессе позволяет увеличить производительность системы путем разнесение сервисов по отдельным физическим машинам, без необходимости внесения изменений в приложение. Отказ какого-либо из сервисов не ведет к отказу всей системы;
- Простота развертывания – в отличие от монолитных приложений, внесение изменений в которое требует перезапуска всей программы, внесение изменений в микросервисные приложения требуют перезапуска только изменившихся компонентов;
- Возможность повторного использования функциональности каждого из сервисов сторонними системами;
- Распределение работы между командам – поскольку каждый сервис представляет собой отдельный проект, работа над ними может быть легко разделена между разработчиками;
- Несмотря на очевидные достоинства, микросервисная архитектура не лишена недостатков, основными из которых являются:
 - Увеличение сложности системы в целом;
 - Дополнительные накладные расходы на передачу данных между микросервисами, сериализацию и десериализацию;
 - Безопасность транзакций. Поддерживать безопасность транзакций при работе с независимыми процессами иногда оказывается нетривиальной задачей.

Несмотря на весь положительный опыт, нельзя однозначно утверждать, что микросервисы — это будущее проектирования ПО. Однако, уже сейчас ясно, что микросервисная архитектура обладает большим потенциалом и предлагает серьезные преимущества для разработки и реализации корпоративных приложений.

Список использованных источников:

1. Sam Newman. Building Microservices. Designing Fine-Grained Systems / Sam Newman // . – O'Reilly Media, 2015. – 280 p.
2. Martin Fowler. Microservices. A definition of this new architectural term [Электронный ресурс] – Режим доступа: <http://martinfowler.com/articles/microservices.html>

ОБРАБОТКА И АНАЛИЗ ИЗОБРАЖЕНИЯ ОТПЕЧАТКА ПАЛЬЦА ВЫДЕЛЕНИЕ ОСОБЫХ ТОЧЕК ОТПЕЧАТКА

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

Козак А. В.

Теслюк В.Н. – канд. физ-мат. наук, доцент

Каждый человек наделен особой меткой, которая отличает его от других людей, - отпечатками пальцев. Отпечатки пальцев делают нас особенными, именно с помощью них (и не только) можно идентифицировать человека. Так дактилоскопия – это наука, изучающая способы распознавания человека по отпечаткам ладони и пальцев рук, основанная на неповторимости папиллярных линий. Впервые дактилоскопию применили для распознавания преступника в Великобритании в 1902г.

В каждом отпечатке можно выделить две группы признаков: локальные и глобальные. Глобальные признаки – те, которые можно увидеть невооруженным глазом. К ним относится, например, тип папиллярного узора («петля», «дуга» и «спираль»). Локальные признаки – минуции (особые точки). Минуции – это уникальные для каждого отпечатка признаки, определяющие пункты изменения структуры папиллярных линий (окончание, раздвоение, разрыв и др.), ориентацию папиллярных линий и координаты в этих пунктах. Каждый отпечаток может содержать от 70 минуций.

Не трудно понять, что глобальные признаки некоторых людей одинаковы, но совершенно невозможно наличие одинаковых микроузоров минуций. Поэтому глобальные признаки используют для разделения базы данных на классы и на этапе аутентификации. На втором этапе распознавания используют уже локальные признаки.

Выделяют несколько подходов к сравнению отпечатков пальцев.

Наиболее популярный подход основан на локальных признаках. Этот подход основан на выделении минуций и сопоставления их с другими образцами. Для определения соответствия входного отпечатка со сравнимым образцом используется формула: $K = \frac{D^2}{pq} \cdot 100\%$, где D – количество совпавших минуций, p – количество минуций эталона, q – количество минуций опознаваемого отпечатка. В случае, если результат превышает 65 %, отпечатки считаются идентичными (порог может быть понижен выставлением другого уровня бдительности).

Рассмотрим механизм и алгоритмы обработки, анализа и выделения минуций отпечатка.

Существуют стандарты для изображения отпечатка. Так считается, что для лучшей и качественной

обработки изображения отпечатка, необходимо, чтобы оно было полутоновым и его размер должен быть 256*256 точек. Для работы с изображением и, в дальнейшем, для выделения минучий необходимо улучшить его качество. Для достижения этого используется ряд фильтров. Рассмотрим и проанализируем основные из них.

На первом этапе улучшения происходит нормализация изображения. Нормализация позволяет удалить высокочастотные шумы и выровнять гистограмму изображения. Для отпечатков пальцев используют следующую формулу:

$$I^*(i, j) = \begin{cases} M_0 + \sqrt{\frac{VAR_0(I(i, j) - M)^2}{VAR}}, & \text{если } I > M \\ M_0 - \sqrt{\frac{VAR_0(I(i, j) - M)^2}{VAR}}, & \text{иначе.} \end{cases}, \text{ где } M = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j), \quad VAR = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (I(i, j) - M)^2$$

Здесь $I^*(i, j)$ – нормализованное изображение, $I(i, j)$ – исходное изображение размером $N \times N$.

Вообще говоря, для улучшения изображения, помимо нормализации, использует ряд фильтров: медианный, Гаусса и др. Но для отпечатков пальца эффективен фильтр Габора. Рассмотрим его реализацию.

Для начала строится ориентационное изображение отпечатка пальцев, которое показывает направление папиллярных линий. Для этого в некоторой локальной области отпечатка находят градиент функции $I(i, j)$ по оси X и оси Y (G_x и G_y соответственно). Далее определяют направление по формуле:

$$O(i, j) = \frac{1}{2} \arctg\left(\frac{v_y}{v_x}\right), \text{ где } v_x = \sum \sum 2G_x(i, j)G_y(i, j), \quad v_y = \sum \sum (G_x^2(i, j) - G_y^2(i, j)).$$

Далее строится частотное изображение, обозначим его через $F(i, j)$. $F(i, j) = \frac{1}{\gamma}$, где γ – количество пикселей между соседними гребнями отпечатка в некоторой локальной области.

Теперь перейдем непосредственно к самой фильтрации Габора, в основе которой стоит модуляционно-передаточная или равно-симметричная функция Габора. Таким образом отфильтрованное изображение получаем по формуле:

$$I^*(i, j) = \sum \sum h(u, v; O(i, j), F(i, j)) \cdot I(i - u, j - v), \text{ где } h(x, y; \varphi, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\varphi^2}{\delta_x^2} + \frac{y_\varphi^2}{\delta_y^2}\right]\right\} \cdot \cos(2\pi f x_\varphi)$$

$$x_\varphi = x \cdot \cos \varphi + y \cdot \sin \varphi$$

$$y_\varphi = -x \cdot \sin \varphi + y \cdot \cos \varphi$$

Пересчет нового изображения (отфильтрованного), т.е. значения яркости в данном пикселе, осуществляется в некоторой локальной области, размер которой определяет размер фильтра Габора. Данный фильтр позволяет не только отфильтровать, но и восстановить отпечаток, если в силу неточного получения его изображения были потеряны некоторые детали.

Фильтр Габора является основным для фильтрации отпечатка, так как большинство других фильтров не учитывают особенности папиллярных линий отпечатка.

Рассмотрим альтернативу фильтру Габора – фильтр, основанный на коэффициенте отражения. Для применения этого фильтра сначала необходимо произвести бинаризацию изображения (рассмотрим это чуть позже) и вычислить для каждого пикселя коэффициент отражения. Таким образом, новое изображение получается из формул:

$$I^*(i, j) = k(i, j) \cdot 255, \text{ где } k(i, j) = \frac{N(i, j)}{n^2}, \quad N(i, j) – \text{ количество белых пикселей, попавших в область фильтра.}$$

После обработки (улучшения) изображения любым способом, для дальнейшей работы с отпечатком, необходимо произвести его бинаризацию, т.е. представить изображение в виде “0” и “1”. Существует множество способов бинаризации, основными из которых являются пороговая и адаптивная бинаризации.

Суть пороговой бинаризации заключается в том, что если значение интенсивности пикселя больше порогового значения, то этот пиксель окрашивается «черным», в противном случае – «белым».

Если пороговая бинаризация неустойчива к перепадам яркости, то адаптивная бинаризация учитывает все эти моменты, поэтому адаптивный подход является более точным. В адаптивной бинаризации, значение порога вычисляется на основе локальных признаков изображения, а не глобальных, как это происходит в пороговой бинаризации.

Следующим этапом после бинаризации является скелетизация, т.е. утоньшение папиллярных линий до толщины в один пиксель. Существуют различные способы скелетизации, и основным из них является метод, основанный на шаблонах. Изображение пробегается фильтром размера 3x3, и если пробегаемая область соответствует шаблону, то средний пиксель закрашивается в белый цвет.

И наконец, последний этап – выделение минучий. Здесь уже алгоритм тривиален: если в окрестности пикселя только один из восьми пикселей черный, то эта точка окончания, если их 3 – точка ветвления. Таким образом, мы получили набор особых точек отпечатка пальца, по которому и производится сравнение.

В ходе работы была произведена оценка алгоритмов фильтрации и бинаризации. Оценка ставилась на основе трех составляющих: быстрота, качество выходного изображения и сложность реализации. Каждый критерий оценивался по шкале от 0 до 7. Например, быстрота в 7 баллах означает, что алгоритм очень быстрый, сложность – 7 баллов означает, самую легкую реализацию.

Следовательно, на основе такой системы был определен наиболее оптимальный алгоритм обработки

изображения отпечатка и выделения особых точек.

Так фильтр Габора показал себя очень хорошо, но его «собрат», основанный на коэффициентах отражения, при тех же данных, дает почти такой же коэффициент качества при своей скорости и легкости реализации.

Что касается бинаризации, то пороговая бинаризация уступает адаптивной по качеству, а скорость и сложность этих алгоритмов отличается незначительно.

Исходя из полученных в ходе исследовательской работы результатов, следует следующие: наиболее оптимальным по качеству, скорости и сложности является алгоритм обработки и анализа отпечатка пальца, основанный на фильтрации с использованием коэффициента отражения и на адаптивной бинаризации.

Список использованных источников:

1. В. В. Жуковский, С. В. Сай, Способ улучшения качества изображения отпечатка пальца, 2009
2. Zin Mar Win, Myint Myint Sein, An Efficient Fingerprint Matching System for Low Quality Images, 2011
3. Rishabh Mishra, Prashant Trivedi, Student Attendance System Based On Fingerprint Recognition and One-to-Many Matching, 2011

ИСПОЛЬЗОВАНИЕ ОБУЧАЮЩИХСЯ АЛГОРИТМОВ ДЛЯ АНАЛИЗА СПРОСА НА ПРОДУКЦИЮ И ОРГАНИЗАЦИИ ЦЕПОЧЕК ПОСТАВЩИКОВ

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

Лось Л. А.

Волорова Н. А. – канд. техн. наук, доцент

В последнее время в условиях постоянных изменений в экономической сфере особое значение придается изучению механизмов рынка, мгновенному реагированию на изменение спроса, регулированию цен, экономии временных и финансовых ресурсов различными способами. Нас будут интересовать способы с применением достижений сферы информационных технологий.

Машинное обучение – направление, формально относящееся к искусственному интеллекту, но также включающее в себя использование разработок математической статистики, теории вероятности, методов оптимизации. Это направление предполагает построение алгоритмов, способных обучаться на предоставленных данных. Т. е. при получении данных алгоритм исследовании закономерностей (обучение) и после этого позволяет сделать предсказания о данных, которые могут быть получены в будущем. Алгоритмов машинного обучения в настоящее время разработано большое количество, для разных сфер и разных задач они различаются, имеют свои плюсы и минусы.

Исходя из определения алгоритмов машинного обучения, к ним можно отнести корреляционный и регрессионный анализ. Эти подходы мы будем использовать для прогнозирования спроса на основе информации о совершенных ранее покупках.

Предположим, что у нас есть интернет-магазин с сохраненными в базе данных операциями покупок, просмотров товаров, добавлений товаров в список желаний, информацией о пользователях и т. д. С помощью *корреляционного анализа* мы можем отобрать те факторы, которые влияют на спрос в большей степени (более взаимосвязаны со спросом). Для этого достаточно рассчитать коэффициент корреляции, он может принимать значения от -1 до 1. При коэффициенте корреляции равном 0 мы можем говорить об отсутствии взаимосвязи между выбранными параметрами. Чем ближе абсолютное значение коэффициента к 1, тем о большей степени зависимости мы можем говорить. После определения значимых факторов мы можем приступить к прогнозированию, используя для этого *регрессионный анализ*. Он позволит нам предсказать значения зависимой переменной (величины спроса) с помощью независимых переменных (факторов), определить, какой вклад принесет изменение независимых переменных.

С развитием информационных технологий стало возможным не только наблюдать и прогнозировать колебания спроса, но и оказывать на него влияние, провоцировать увеличение спроса на конкретную продукцию. В этой сфере нам могут пригодиться многие алгоритмы машинного обучения.

Сбор предпочтений пользователей на данный момент осуществляется во многих интернет магазинах, новостных сайтах, сайтах с медиа контентом и д. р. для рекомендации вам товаров, фильмов, музыки, новостей, ссылок на другие сайты и многого другого. Рассмотрим некоторые подходы.

Коллаборативная фильтрация работает следующим образом: рассматриваются большие объемы данных, например, о покупателях. Находятся более мелкие группы со сходными покупками, которые объединяются в один список и сортируются по какому-либо рейтингу (по количеству выбравших этот товар людей). Затем эти товары, за исключением уже приобретенных, рекомендуются покупателю. Для определения схожести каждый покупатель сравнивается со всеми другими и находится коэффициент подобия. Этот

коэффициент может быть рассчитан разными путями, выбирать которые следует в каждом конкретном случае, например, определение *евклидова расстояния*, нахождение *коэффициента корреляции Пирсона*, *коэффициента Жаккарда*, установление *манхэттенского расстояния* и т. д.

В приложениях, обрабатывающих большие объемы данных, часто используется *кластеризация* – метод обнаружения и визуализации групп, связанных между собой предметом, людей. Алгоритм *иерархической кластеризации* на каждом шаге объединяет две самые похожие группы на основе самого близкого расстояния между группами (чем ближе они друг к другу, тем более похожи). Алгоритм продолжает работу до тех пор, пока не останется только одна группа. Результаты такого прохода по данным можно представить в виде графа, на котором видно из чего состоит кластер и расстояния между группами. Минусами алгоритма является то, что древовидная структура не разбивает данные на группы, и на больших наборах этот алгоритм будет работать медленно, так как приходится попарно вычислять соотношения между группами. Эти недостатки не появляются при использовании *кластеризации методом К-средних*. В отличие от предыдущего решения, этот алгоритм начинается с выбора k случайно расположенных точек, которые станут центрами кластеров. Каждому элементу назначается ближайший к нему центр, и центр переносится в новую точку, вычисленную как среднее по всем его элементам. Цикл продолжается до тех пор, пока центры не перестанут передвигаться. Алгоритмы кластеризации удобно использовать для определения групп сходных товаров предлагаемых покупателю для сравнения.

Цепочка поставок – это совокупность организаций, людей, технологий, информационных и финансовых потоков, в процессе прохождения продукта от поставщика к потребителю. Управление цепочкой поставок включает несколько стадий. В проекции на интернет-магазин они могут выглядеть следующим образом:

- планирование – разработка планов соответствия товаров ожиданиям и потребностям потребителей (будем считать, что методы и алгоритмы, изложенные в предыдущей части, помогли определить выбор продукции и ее объемы);

- выбор поставщика – выбор поставщиков сырья/готового продукта, планирование способов доставки, оплаты, размещения и т.д.;

- производство – допустим, что этот этап для интернет-магазина, реализующего готовую продукцию не актуален;

- поставки – создание сетей складов, организация приема заявок от клиентов, транспортировка и т.д.;

- возврат – организация возврата дефектных товаров организация технического обслуживания.

Организовать цепь поставок с такими этапами можно множеством способов, перебор такого количества возможных вариантов вручную займет не один день. Поэтому в данной ситуации весьма логично будет использование *методов оптимизации*. Такие методы используются в случаях с множеством возможных решений, зависящих от множества переменных. Оптимизация сводится к поиску наилучшего решения путем подбора различных сочетаний переменных и их сравнения между собой. Для решения любой оптимизационной задачи необходимо правильно подобрать целевую функцию. После ее выбора задача сводится к подбору таких входных параметров, которые будут минимизировать значение этой функции.

В нашем случае входными переменными могут быть цены поставщиков на товары, стоимости доставок товаров на склады магазина, время доставки, стоимость аренды складов, состав комплектаций поставок, стоимость транспортировки товара покупателю, время ожидания товара покупателем. Таких факторов можно подобрать множество, и не все их сразу выражены в денежном эквиваленте. Так придется выбрать учитывать ли время ожидания товара или оценивать только стоимость его доставки. Для таких факторов можно придумать их выражение в денежном эквиваленте. Определившись с переменными, влияющими на стоимость нужно составить функцию, в результате которой мы получим одно число (например, общую стоимость всех этапов), - это и будет наша *целевая функция*.

Для решения этой задачи могут использоваться разные алгоритмы оптимизации. *Алгоритм случайного поиска* (генерируется конечное число гипотез и для них значение целевой функции, возвращается наилучшая гипотеза) не гарантирует нахождение оптимального решения.

Алгоритм спуска с горы начинается со случайного решения и ищет лучшие решения возле него. Возможна ситуация нахождения локального минимума, который может не совпадать с глобальным минимумом (оптимальным решением).

Алгоритмы имитации отжига пытаются бороться с проблемой скатывания в локальный минимум. Они включают возможность перехода к худшему решению, чтобы найти в итоге лучшее.

Генетические алгоритмы состоят в том, чтобы создать набор случайных решений (популяцию). Значение целевой функции вычисляется для всех выбранных гипотез на каждом шаге, затем формируется новая популяция. В нее входят лучшие решения из текущей и новые (полученные из лучших с помощью модификации). Итерации повторяются заданное число раз или до прекращения улучшения целевой функции.

От условий задачи зависит, будет ли работать конкретный алгоритм. Так большинство из них основаны на том, что оптимальное решение находится близко к другим хорошим решениям, и если это условие не выполняется, любой из методов оптимизации будет работать не лучше, чем случайный поиск.

Использование алгоритмов машинного обучения во многих сферах дает значительный выигрыш во времени, потраченном на нахождение закономерностей в больших объемах данных, позволяет делать на них прогнозы. Что в свою очередь сокращает расходы, улучшает функционирование компаний и предоставляет преимущество перед конкурентами.

Список использованных источников:

1. Сегаран. Т. Программируем коллективный разум / Сегаран. Т. // Пер. с англ. – СПб: Символ-Плюс, 2008. – 368 с., ил.
2. Введение в Управление цепочками поставок (SCM) [Электронный ресурс] – Режим доступа. – URL

<http://www.galaktika.ru/amm/introduction-to-scm.html> (дата обращения 04.04.2016).

3. Построение алгоритма отбора факторов, влияющих на конъюнктуру товарного рынка [Электронный ресурс] – Режим доступа. – URL <http://dis.ru/library/531/26996/> (дата обращения 04.04.2016).

ВЕРОЯТНОСТНАЯ МОДЕЛЬ РЕЙТИНГОВ ИГРОКОВ В КОМАНДНЫХ ВИДАХ СПОРТА

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

Михолап А. А., Стержанов М. В.

Рассмотрена баесова модель оценки уровня игры спортивных команд и отдельных участников. Приведен способ уменьшения вычислительной сложности модели с помощью фактор-графа. Предложен способ повышения предсказательной точности путем агрегации результатов различных систем.

Ключевые слова: системы рейтингов, баесовы методы, вероятностные модели.

Задача ранжирования и построения рейтингов возникает в различных ситуациях и особенно часто в контексте спортивных соревнований. Объективная оценка навыков игроков и команд помогает принимать решения при покупке новых игроков, построении списка лучших команд и прогнозировании исходов будущих матчей.

В 2006 году корпорация Microsoft опубликовала статью с описанием новой рейтинговой системы TrueSkill. Она была разработана коллективом авторов для подбора соперников в многопользовательской онлайн игре Halo 2 и, согласно мнению авторов, может рассматриваться как обобщение Elo.

Недостаток Elo заключается в дискретных уровнях коэффициента скорости изменения рейтингов K . Они были подобраны вручную, на основании истории игр и здравого смысла. Однако, желательно, чтобы система могла подстраивать K автоматически. Например, если новичок играет со старым игроком, уровень игры которого стабилен и достаточно точно известен, то результат этого матча намного полезнее, чем матча против другого новичка, о котором ничего не известно. Чтобы сделать это очевидным, можно упростить и принять уровень навыка опытных игроков за точку. Тогда, сравнивая новичка с ними, мы получим для него довольно точные координаты. Если же случайная величина рейтинга противника также имеет широкий интервал, мы не можем сделать таких выводов, так как оценка очень неопределенная, а сами интервалы даже могут перекрываться.

TrueSkill помимо среднего значения μ варьирует также и дисперсию σ для каждого игрока, благодаря чему может автоматически сжимать и увеличивать интервал неопределенности без ручного вмешательства, основываясь лишь на истории игр.

Второй предпосылкой создания нового алгоритма была необходимость моделирования ничьих в явном виде. Вероятность ничьей это один из критериев качества разбиения игроков на команды в Halo 2 и других играх и видах спорта. Матч наиболее интересен, когда силы обеих сторон равны и исход непредсказуем.

Также в систему встроена детализация на уровне отдельных игроков. Чтобы автоматизировать управление случайными величинами приходится неизбежно усложнять алгоритм. Если в Elo использовалась базовая теория вероятности, логистическая функция и простое правило обновления, в TrueSkill за счет введения переменной дисперсии применяются значительно более сложные конструкции.

С начала двухтысячных годов в статистике начали набирать значительную популярность Баесовы методы. Они представляют собой набор правил работы со случайными величинами, позволяя упростить идеальную вероятностную модель до практически применимого уровня. Это возможно за счет ручного определения значимых факторов (которые описываются случайными величинами) и связей между ними.

TrueSkill использует именно Баесов подход, выражая модель встречи двух команд фактор графом (рисунок 1).

Система инициализируется определенными значениями, называемыми априорной информацией. В нее можно включить всевозможные сведения от экспертов, но в данном случае это просто фиксированный стартовый рейтинг. Важно различать априорную информацию в контексте всей системы и одного матча.

До игры осуществляется прямой проход по фактор графу сверху (от узлов, содержащих рейтинг игроков) вниз (к узлам, обозначающим исход матча). С его помощью оцениваются вероятности различных исходов. После завершения игры случайные величины в нижних узлах заменяются на реальные числовые

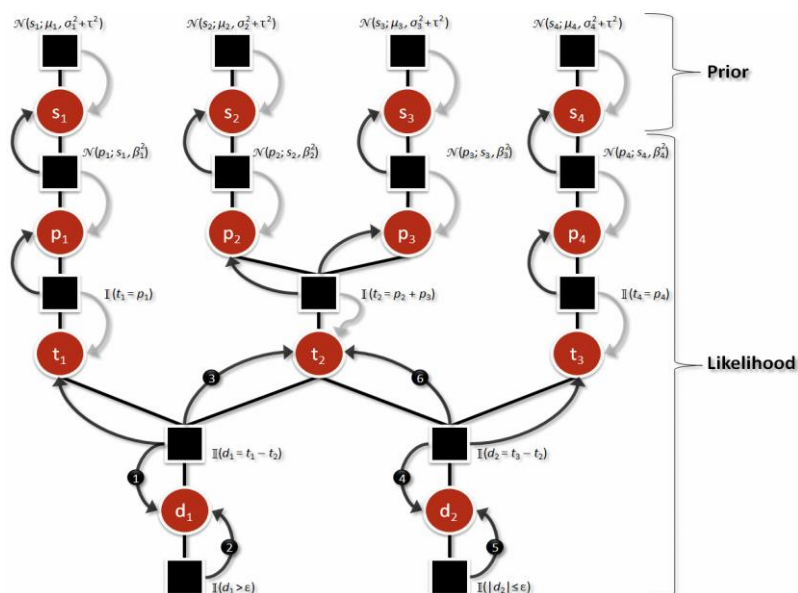


Рис. 1. Фактор граф TrueSkill

результаты и происходит обратный проход снизу вверх. Он обновляет рейтинги так, чтобы только что произошедшее событие стало более вероятным. То есть, если команда выигрывает — увеличивает рейтинг игроков, если проигрывает уменьшает.

Баесовы методы известны своей вычислительной сложностью — необходимо приближенно брать множество интегралов. Фактор граф значительно ускоряет вычисления, накладывая довольно мягкие в контексте этой задачи ограничения. Как и Elo TrueSkill имеет настраиваемый параметр влияния удачи. Здесь он интерпретируется как длина «цепочки умений». Эта «цепочка» состоит из упорядоченных по возрастанию фиксированных уровней игры. Если звено связывает уровни A и B, то игрок уровня A имеет 20% шанс выиграть у игрока уровня B. Чем цепь длиннее, тем больше разница между худшим и лучшим игроком и, следовательно, игра больше зависит от умения и меньше от везения.

Из других достоинств TrueSkill можно отметить поддержку «неполного матча», когда человек выходит из игры до ее завершения, поддержку команд различной численности и явное моделирование ничьих путем введения дополнительного параметра, отвечающего за ее вероятность.

Для дополнительного улучшения качества прогнозирования предлагается использовать композицию различных моделей TrueSkill. Уникальность моделей обеспечивается случайной подвыборкой данных для настройки их коэффициентов. Усреднение по оценкам множества таких моделей дает оценку, стабильно превосходящую по точности каждую из своих компонент. Улучшение наблюдалось на всех протестированных наборах данных.

В настоящее время вероятностный подход в моделировании рейтингов превосходит другие способы по точности и интерпретируемости, но требует больших вычислительных ресурсов. Исследованная модель TrueSkill решает эту проблему, используя фактор-граф для кэширования промежуточных результатов вычислений. Применение агрегирования моделей, настроенных на различные случайные подвыборки исходных данных дало устойчиво лучший результат.

Список литературы:

1. Википедия. Эло, Арпад – Википедия. – accessed 24-November-2015]. https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%BE_%D0%90%D1%80%D0%BF%D0%B0%D0%B4.
2. Herbrich, Ralf. TrueSkillTM: A Bayesian Skill Rating System / Ralf Herbrich, Tom Minka, Thore Graepel // Advances in Neural Information Processing Systems 19 Ed. by B. Schölkopf, J.C. Platt, T. Hoffman. – MIT Press, 2007. – Pp. 569–576. <http://papers.nips.cc/paper/3079-trueskilltm-a-bayesian-skill-rating-system.pdf>.
3. Kschischang, F. R. Factor Graphs and the Sum-product Algorithm F. R. Kschischang, B. J. Frey, H. A. Loeliger // IEEE Trans. Inf. Theor. – 2006. – sep. Vol. 47, no. 2. – Pp. 498–519. <http://dx.doi.org/10.1109/18.910572>.
4. Minka, Thomas P. A Family of Algorithms for Approximate Bayesian Inference: Ph.D. thesis. – Cambridge, MA, USA : Massachusetts Institute of Technology, 2001. – AAI0803033.
5. Wikipedia. Bradley–Terry model – Wikipedia, The Free Encyclopedia. – 2015 – [Online; accessed 17-November-2015]. https://en.wikipedia.org/wiki/Bradley%E2%80%93Terry_model.

ПРОБЛЕМЫ ДИНАМИЧЕСКОГО СОЗДАНИЯ И МОНИТОРИНГА ВИРТУАЛЬНЫХ МАШИН

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

Побыванец Е. Н.

Волорова Н. А. – канд. техн. наук, доцент

Аппаратные мощности современных вычислительных систем постоянно растут, что неудивительно. Как следствие, программы, которые раньше требовали значительных ресурсов для своего выполнения, теперь потребляют только часть ресурсов вычислительной системы (и не редки случаи, когда это лишь малая часть). Многие приложения теперь можно (а в большинстве случаев и запускаются) одновременно на одном сервере, при этом иногда из виду упускаются проблемы безопасности или распределения нагрузки между приложениями. Одним из возможных решений данной проблемы может стать виртуализация.

Виртуализация – это логическое объединение некоторых ресурсов системы в отдельное изолированное программное окружение (Application Environment). При этом стоит отметить, что изолирование может происходить над различными ресурсами системы, в частности:

- Виртуализация оборудования (Виртуальная машина)
- Виртуализация представлений (Терминальные серверы \ службы)
- Виртуализация программного обеспечения (Портативные приложения, с эмуляцией необходимых сервисов)
- Виртуализации памяти (Изолированная область памяти, применяется практически в каждой ОС)
- Виртуализация хранилища данных (Представление нескольких носителей данных как единый носитель)
- Виртуализация базы данных (Абстрагирование от конкретной СУБД для представления чистых данных)
- Виртуализация сети (в частности, VPN)

В контексте решения поставленной проблемы, наибольший интерес представляют первые 3 подхода. Виртуализация представлений является достаточно знакомым подходом – это терминальные сервера или терминальные службы. Суть этого подхода весьма проста - терминальный сервер представляет своим клиентам ресурсы для выполнения определенного приложения, и все вычисления происходят на стороне сервера, в то время как клиенту показывается лишь результат или представление. Такая модель имеет ряд преимуществ: клиенты могут использовать практически любое аппаратное обеспечение, не требуется большая пропускная способность канала, терминальные системы осуществляют проверку действия пользователя, что повышает безопасность, и из-за того, что все клиенты подключаются к одному терминальному серверу, облегчается обслуживание и поддержка сервера. Но есть и недостатки: так как множество пользователей работают с одним сервером, увеличиваются требования к аппаратному обеспечению сервера, проблемы сервера сказываются на всех пользователях, а также пользователи в большинстве случаев работают с одинаковым набором ПО.

Виртуализация программного обеспечения – относительно новый подход. Он подразумевает под собой то, что приложение запускается в некоторой изолированной среде, внешне похожей на обычную структуру папок (часто называемой «Песочницей» или «Sandbox»). Для каждой «песочницы» эмулируются необходимые ресурсы системы (сервисы, драйвера, записи реестра), что позволяет запускать даже конфликтующие приложения в своих изолированных средах. Чаще всего виртуальные приложения доставляются на компьютер из определенного контейнера или поставщика, что позволяет контролировать актуальность версий данных приложений. Рассматривая плюсы данного подхода, легко выделить следующие плюсы: безопасность работы с приложениями, независимость от конфликтных приложений, простота администрирования. Но при этом стоит отметить, что данный подход имеет некоторые сложности с пониманием и начальной настройкой, что затрудняет его интеграцию.

И самый интересный подход – виртуализация оборудования. Данный подход позволяет создавать полностью изолированные виртуальные машины, полностью эмулирующую работу нормального сервера \ компьютера. Реализуется этот подход с помощью установки на сервер программы-гипервизора – программы, обеспечивающей созданию параллельно работающих операционных систем и предоставляющий набор инструментов для управления этими машинами и общими ресурсами.

В ходе работы я разработал прототип веб-приложения, позволяющего создавать такие виртуальные машины обращаясь уже к предварительно установленному на сервер гипервизору (Microsoft Hyper-V), используя веб-интерфейс и предустановленные образы с необходимым ПО и анализируя возникающие проблемы, а также производительность гостевых ОС.

Как показывает анализ, при разработке и использовании таких систем возникают следующие проблемы

- Доступность – при реализации возникают проблемы с доступом к гостевой ОС расположенной на другом сервере.
- Мониторинг состояний виртуальных машин – по умолчанию, MS Hyper-V предоставляет не

самый легкий способ получения текущего состояния виртуальной машины, а хотелось бы видеть эти данные в динамике, в связи с чем требуется написание отдельного модуля для сборки статистики.

Рассматривая анализ производительности, стоит выявить, что гостевая ОС показывает сравнимую производительность с ОС, схожей по аппаратному обеспечению с созданной гостевой ОС (Используя бенчмарки, разница составляет примерно 4-7%)

В целом виртуализация на уровне оборудования имеет много преимуществ:

- Виртуальные машины не занимают места в серверных стойках
- Достаточно просты в администрировании, если есть доступ к гипервизору
- Предоставляют широкие возможности по автоматизации процесса
- Имеют много возможностей по перераспределению нагрузки и миграции систем

При этом стоит отметить, что даже у этого подхода есть недостатки, присущие виртуализации, основной на представлении – требуется более мощное аппаратное обеспечение для серверов-хостов, а также в случае отказа физического сервера – отказывают и все виртуальные машины.

Исходя из вышесказанного, можно сделать вывод, что виртуализация имеет много преимуществ, и в определенных ситуациях может стать решением многих проблем. При этом в зависимости от конечной цели можно использовать различные подходы, предоставляющий разный уровень нагрузки как на сервер, так и разный уровень затрат на обслуживание.

Список использованных источников:

1. Электронный ресурс, Hyper-V Started Guide, <https://technet.microsoft.com/en-us/library/cc732470%28v=ws.10%29.aspx?f=255&MSPPError=-2147217396>
2. Aidan Finn, Mastering Hyper-V Deployment, 2010

ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ ВЕБ-СЕРВИСОВ НА ОСНОВЕ МОДЕЛИ АКТОРОВ

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

Смоляков К. Ю.

Ганжа В. А. - канд. физ.-мат. наук, доцент

В настоящее время широкое распространение получила парадигма «облачных вычислений». Она подразумевает использование нескольких вычислительных узлов и хранилищ различными клиентскими устройствами: персональными компьютерами, смартфонами, встроенным оборудованием, датчиками и т.д. Интерфейсом и связующим звеном для всех этих устройств обычно выступают ТСП/HTTP веб-сервисы. Для того чтобы упростить создание распределенных систем с низким временем отклика и высокой доступностью, используются фреймворки на основе модели акторов: Microsoft Orleans для платформы .NET, Akka для платформы JVM, и другие реализации.

Основной задачей данной работы является обзор и анализ высокоуровневых паттернов проектирования распределенных систем для дальнейшего использования при решении прикладных задач. Основная цель — максимально эффективное использование системных ресурсов и преимуществ программной модели акторов.

Модель акторов впервые была упомянута в 1973 году. Широкое распространение в решении задач параллельного программирования в распределенных системах получила сравнительно недавно. Модель акторов представляет собой систему изолированных друг от друга объектов, которые общаются между собой посредством асинхронной передачи сообщений. Внутреннее состояние каждого актора может быть изменено только самим актором при получении сообщения. Каждая операция по обработке полученного сообщения выполняется в контексте одного программного потока. Таким образом прикладному программисту не нужно беспокоиться о проблемах параллельного программирования.

В дальнейшем для решения прикладных задач модель акторов была расширена. Так, например, в компании Microsoft была разработана библиотека Orleans, которая использует собственную расширенную модель «виртуальных акторов», основными принципами которой являются:

1. **Вечное существование.** В Orleans нет возможности создавать и удалять экземпляры акторов. Акторы являются чисто логическими сущностями и всегда существуют (виртуально), к ним всегда можно обратиться по известному ключу.
2. **Автоматическая активация.** Среда выполнения Orleans автоматически создает и удаляет экземпляры акторов на основе запросов клиентов и имеющихся ресурсов. В любой момент времени у каждого актора может быть несколько активаций, либо не быть ни одной.
3. **Независимость от размещения.** Экземпляр актора может быть создан в разных узлах системы, в зависимости от времени и других факторов. Иногда актор может оставаться чисто виртуальной сущностью, то есть не иметь физического размещения.
4. **Автоматическое масштабирование.** В данный момент Orleans поддерживает два режима

активации акторов. Первый разрешает существование только одного экземпляра актора в системе (single activation mode). Второй режим позволяет создавать несколько экземпляров в зависимости от загрузки (stateless worker mode).

Основными преимуществами модели «виртуальных акторов» являются высокая продуктивность разработки и автоматическая масштабируемость. Таким образом для использования нет необходимости быть экспертом в области распределенных систем.

В процессе исследования были выявлены следующие критерии, при выполнении которых использование модели акторов может быть оправданным:

- Большое количество слабо связанных программных сущностей (100 – 1 млн).
- Программные сущности достаточно малы и могут быть однопоточными.
- Взаимодействия типа запрос - ответ или начало – мониторинг – завершение.
- Заранее известно, что понадобится более одного сервера.
- Нет необходимости в глобальной координации операций, только между небольшим количеством программных сущностей за один раз.

Критерии, при которых применение модели акторов может быть проблематичным:

- Необходимость прямого доступа к памяти одного объекта другому.
- Маленькое количество больших сущностей, необходима длительная (возможно многопоточная) обработка данных
- Необходима глобальная координация и целостность данных

Для решения наиболее часто возникающих задач при помощи модели акторов могут быть использованы следующие паттерны:

1. Умный кэш (Smart Cache).

Используется для оптимизации производительности доступа к хранилищу данных. Позволяет ускорить операции чтения, так как кэширующий актор хранит нужные данные в оперативной памяти. Позволяет буферизовать операции записи, снизив при этом нагрузку на базу данных.

Кэш называют «умным», так как кэшируемые данные хранятся в локальных переменных экземпляра актора и могут быть представлены различными сложными структурами данных: очередями, сортированными множествами, хеш-таблицами.

Паттерн можно использовать в следующих случаях:

- Хранилище данных сильно замедляет время отклика системы в целом
- Операции чтения преобладают над операциями записи
- Нет необходимости делать сложные запросы к данным, либо данные представлены документами
- Необходимы быстрые операции записи, но допустима частичная потеря данных

Примеры использования: таблица результатов для real-time игры, очередь задач, сервис конвертирования валют

2. Распределенная сеть / граф (Distributed Network / Graph)

Является разновидностью известного паттерна publish / subscribe, позволяет моделировать сложные взаимосвязи с помощью объектной модели.

Паттерн можно использовать, когда о произошедшем событии нужно асинхронно уведомлять нескольких участников системы. Может быть реализован различными способами: с помощью потоков данных (например, Orleans Streaming), либо прямой передачей сообщения от актора-источника к акторам-подписчикам.

Примеры использования: социальные сети, групповые чаты, «интернет вещей»

3. Редуктор (Reducer)

Обеспечивает иерархическую структуру вида, необходимую для получения агрегата значений, хранящихся во внутреннем состоянии многих акторов. Иерархия в этом случае выстраивается в следующем порядке (снизу-вверх): актор-значение, актор-промежуточное значение, актор-агрегат. Обновление значение происходит одновременно, либо по таймеру. В качестве агрегата можно вычислять среднее значение, сумму, количество или другие показатели.

Примеры использования: «интернет-вещей», сбор телеметрических и статистических данных

Существуют также и другие, более специфичные паттерны использования модели акторов, такие как Cadence, Resource Manager, Distributed Computation, Stateful Service Composition и др.

Список использованных источников:

1. Bernstein P., Bykov S., Geller A., Kliot G., Thelin J. Orleans: Distributed Virtual Actors for Programmability and Scalability, no. MSR-TR-2014-41, 2014 – 13p.
2. Параллельные вычисления с помощью акторов [Электронный ресурс]. — Режим доступа: <https://github.com/anton-k/ru-neophyte-guide-to-scala/blob/master/src/p14-actors.md>
3. Tuna H., Martinez C. Applied Actor Model with Orleans [Электронный ресурс]. — Режим доступа: <https://github.com/hataytuna/Distributed/blob/master/Applied%20Actor%20Model%20with%20Orleans%20v1.1.pdf>
4. Bykov S., Kliot G., Roberts M., Thelin J. Orleans Best Practices, Microsoft Research, 2014 — 14p.

ИССЛЕДОВАНИЕ СОВРЕМЕННЫХ МЕТОДОВ ПОСТРОЕНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

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

Асташкевич М.Г.

Волорова Н. А. - заведующая кафедрой информатики

Развитие информационных технологий, в частности, увеличение вычислительной мощности техники и удешевление хранения данных, позволяет собирать и анализировать огромные объемы информации о пользователях, в том числе данные о предпочтениях и поведении пользователей. Эти данные представляют большую ценность для различных организаций. Например, поисковые гиганты, такие как Google, Яндекс и другие, используют такие данные о том или ином пользователе сети для ранжирования результатов поисковых запросов, а также для более эффективного подбора показываемых рекламных объявлений. Также в качестве примера можно привести эксперимент исследователей из бизнес-школы Уорика (Warwick Business School), которые предположили возможность прогнозирования поведения рынка ценных бумаг на основе анализа открытых данных сервиса Google Trends. В результате исследователи смогли увеличить портфель акций на 8% за две недели - что является достаточно впечатляющим результатом, учитывая, что исходные данные брались из открытых источников. Таким образом, преследуемая автором цель заключается в исследовании современных подходов к прогнозированию и построению рекомендаций на основе данных, находящихся в открытом доступе.

Задача об анализе данных о поведении и предпочтениях пользователя с целью последующей обработки и построения определенных прогнозов относительно дальнейшего его поведения не нова. Крупные интернет-сервисы десятилетиями предлагают пользователям такую функциональность - Last.fm предлагает музыку, Netflix предлагает фильмы, Amazon предлагает товары и т.д. Способность предложить пользователю то, что ему нужно, может сильно повлиять на успех коммерческого проекта.

Входные данные таких алгоритмов можно обобщить к следующему виду: пользователь x совершает действие g_x по отношению к объекту y . Действие в данной трактовке - это просмотр информации об объекте, "лайк", выставленная оценка, оставленный комментарий - словом, все, что показывает интерес пользователя к данному объекту. Таким образом образуется входная матрица рейтингов - большая и сильно разреженная.

Рекомендательные системы, которые и занимаются обработкой такого рода информации, делятся, в общем, на два вида - основанные на содержимом рекомендательные системы и методы коллаборативной фильтрации, причем коллаборативная фильтрация является более универсальным подходом. Простейшие алгоритмы основываются на группировке пользователей по оцененным объектам, или на группировке объектов по оценившим их пользователям. В обоих случаях сценарий схожий: определить способ группировки пользователей, рекомендовать пользователю объекты, интересные его группе пользователей (и наоборот - для рекомендательных систем, основанных на группировке по объектам). Среди коллаборативных рекомендательных систем выделяется алгоритм SVD, который работает несколько другим образом и признан на данный момент самым перспективным и наиболее точным.

SVD (Singular Value Decomposition), переводится как сингулярное разложение матрицы. В теореме о сингулярном разложении утверждается, что у любой матрицы A размера $n \times m$ существует разложение в произведение трех матриц: U , Σ и V^T . Зная о существовании такого разложения, можно использовать принципы машинного обучения для того, чтобы подобрать такие матрицы U и V , которые получат наименьшую погрешность при прогнозировании рейтинга для известных элементов матрицы g (обучающая выборка) и прогнозировать с их помощью события в будущем. Для определения погрешности можно использовать, например, среднеквадратичное отклонение (RMSE). Для того, чтобы алгоритм не только находил наилучшее решение для известных данных, но также имел хорошие результаты предсказывая будущие события, нужно к данной метрике погрешности добавить регуляризатор. Регуляризация заключается в том, что оптимизируется не просто ошибка, а ошибка плюс некоторая функция от параметров (например, норма вектора параметров). Это позволяет ограничить размер параметров в решении, уменьшает степень свободы модели. Таким образом задача о построении рекомендательной системы сводится к задаче о нахождении таких матриц U и V , для которых метрика ошибки плюс регуляризатор минимальны. С этим можно справиться, например, с помощью алгоритма градиентного спуска.

Таким образом, в проведенном исследовании были рассмотрены основные современные алгоритмы построения рекомендаций, разработан алгоритм тестирования рассмотренных алгоритмов на основе анализа информации о комментариях пользователей крупного белорусского новостного сайта. Были также рассмотрены существующие программные решения для решения подобных задач. Также были определены возможные дальнейшие направления исследования в данной области.

Список использованной литературы:

1. Алексей Кострикин. Введение в алгебру. Часть 1. Основы алгебры / Алексей Кострикин. — 1-е изд. — Физико-математическая литература, 2001. — 272 с.
2. Владимир Зорич. Математический анализ / Владимир Зорич. — 1-е изд. — МЦНМО, 2007. — 1458 с.
3. Константин Воронцов. Машинное обучение (курс лекций) / Константин Воронцов.

ВЗЛОМ ПОЛИНОМИАЛЬНЫХ ХЕШЕЙ ПО РАВНЫМ СТЕПЕНИ ДВОЙКИ МОДУЛЯМ С ПОМОЩЬЮ ПОСЛЕДОВАТЕЛЬНОСТИ МОРСА-ТУЭ

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

Бутыма В.С. Духовник А.Н.

Стройникова Е. Д. - ассистент кафедры информатики

В данной работе будет рассмотрен случай неправильной работы программ, в особенности используемых для решения задач на олимпиадах по программированию, основанных на сравнении (либо любого другого действия) с помощью хешей строки на самом популярном модуле 2^{64} (как верхняя граница Int64).

Строка, или рекурсивная последовательность, Морса-Туэ — это набор единиц и нулей, которая может принимать любую длину, равную степени двойки. В 1906 году Акселем Туэ была предложена данная строка, которую также называют “слово Туэ”, как аперриодическая рекурсивно-вычислимая последовательность. Рекурсивное правило получения строки: $s_n = \text{not}(s_{n-1}) + s_{n-1}$, not – логическое отрицание, с начальным значением 1 или 0 (в итоге получатся 2 разные последовательности, различаемые инверсированием битов).

Свойства последовательности, начальное значение для которой равно 1 (для начального значение 0 второе и третье свойство меняются четностью):

- Фрактал, поэтому применима в алгоритмах фрактального сжатия.
- Если удалить элементы, которые находятся на четных местах, последовательность не изменится (если длину примем бесконечность)
- Если удалить элементы, которые находятся на нечетных местах, последовательность инверсируется. (если длину примем бесконечность)
- Последовательность не изменяется, если подействовать на нее алгоритмом сжатия Хаффмана.

Некоторая информация о хешах. Если нам дана строка $s_{0..n-1}$, длины n , то хеш(или полиномиальный хеш, т.к. он основан на сумме степеней некоторого основания с коэффициентами) - это число $h = \text{hash}(s_{0..n-1}) = [s_0 + p*s_1 + p^2*s_2 + \dots + p^{n-1}*s_{n-1}] \pmod{B}$ где p - основания, B - модуль — натуральные числа. Наиболее важное свойство хешей – это то, что у равных строк хеши всегда равны. Таким образом, хеши позволяют быстро проверять, являются ли 2 строки одинаковыми, только сравнив их хеш. Однако существует вероятность коллизий, когда разные строки, будут давать одинаковый хеш.

В ходе работы была найдена интересная особенность этой последовательности. Этой особенностью является то, что она является как бы анти-тестом для программ (в основном применяемых в олимпиадном программировании), которые используют полиномиальные хеши по модулю $[B=2]^{64}$ просто пользуясь переполнением типа. Этот модуль является самым распространенным, т.к. ограничивает стандартный тип `long int(64 бита)` в C++ . Строка, сгенерированная по правилу $1 = 'b', 0 = 'a'$ на строке Морса-Туэ вызывает коллизии уже на размерах строки порядка $[10]^3$, что намного меньше модуля B . Соответственно все модули, являющиеся степенями 2, также будут неверно работать на таких строках, при этом p может быть абсолютно любым. Далее это будет показано.

Полиномиальный хэш от строки s длины l равен $(s_0 + p*s_1 + p^2*s_2 + \dots + p^{l-1}*s_{l-1}) \pmod{B}$. В качестве p мы берём простое число большее длины алфавита. Докажем, что $\text{hash}(s[0..(2k-1)])$ при некотором k совпадёт с $\text{hash}(s[(2k)..(2k+1-1)])$. $\text{not}(SQ)$ и SQ встретятся в больших строках не малое количество раз, что явно следует из рекуррентного соотношения, определения данной строки, поэтому вероятность коллизии очень высока. Вместо a и b в строке можно сразу использовать 0 и 1. $\text{hash}(\text{not}(SQ)) - \text{hash}(SQ) = T = P_0 - P_1 - P_2 + P_3 - P_4 + P_5 + P_6 - P_7 \dots \pm P_{2Q} - 1$, после подстановки $'a' = 0, 'b' = 1$ все части, в которых $s_i = 'a'$ сокращаются.

Применим некоторые преобразования и получим: $T = (P_1 - 1)(P_2 - 1)(P_4 - 1) \dots (P_{2Q} - 1 - 1)$. Заметим, что $P_{2i+1} - 1 = (P_{2i} - 1)(P_{2i} + 1)$ делится на i -ую и ещё на какое-то чётное число $P_{2i} + 1$. Получаем, что если i -ая скобка делится на $2r$, то $(i+1)$ -ая скобка делится по меньшей мере на $2r+1$.

Таким образом, получаем: $(P_1 - 1)(P_2 - 1)(P_4 - 1) \dots (P_{2Q} - 1 - 1)$, что делится, по меньшей мере, на $2 \cdot 2^2 \cdot 2^3 \cdot \dots = 2Q(Q-1)/2$. А значит, что уже при $Q > 11$, хеш какой-либо нужной нам подстроки будет зануляться, т.к. его значение делится на B . Получается, что строка будет вызывать коллизии в полиномиальном хеше, следовательно, полиномиальные хеши по модулю 2^{64} будут неприменимы ко многим задачам на строках такого типа, т.к. разные строки будут приниматься как равные, если сравнивать их хеши. Заметим, что существуют строки, которые вызывают коллизии для хешей, для абсолютно любого p и B . Для малых B и r достаточно просто найти такую строку простым перебором “в лоб”. Для больших значений, полный перебор строк для которых слишком ресурсоемкий, найти такую строку сложно и приходится использовать специальные свойства B и p .

Список использованных источников:

1. Neerc.ifmo [Электронный ресурс]. – Электронные данные. – Режим доступа: http://neerc.ifmo.ru/wiki/index.php?title=Слово_Туэ-Морса

2. e-maxx [Электронный ресурс]. – Электронные данные. – Режим доступа: http://e-maxx.ru/algo/string_hashes

СТЕГАНОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

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

Губский М. Д.

Стройникова Е. Д. – ассистент кафедры информатики

Во все времена ценность информации была высока. На протяжении столетий люди использовали всевозможные приемы защиты важных данных от несанкционированного доступа, всякий раз придумывая все более сложные и надежные. Как правило, это были криптографические методы. Но порой лишь зашифровать необходимые сведения являлось недостаточным. Для усиления защиты использовались методы сокрытия информации — стеганографические методы.

На сегодняшний день существует множество способов сокрытия данных. Одним из простых в реализации, но не таким простым в обнаружении, является стеганография.

Стеганография (греч. тайнопись) — наука, изучающая передачу и хранение информации при сокрытии самого факта ее существования. Уже в IV в. до н. э. использовали стеганографию, чтобы передавать важные сообщения. Так, например, наносилось необходимое сообщение на обритуемую голову раба. Когда волосы отрастали, он отправлялся к адресату, который, сбрав голову, считывал сообщение.

За время своего существования стеганография претерпела множество изменений и дополнений. На современном этапе сформировалось три направления стеганографии: классическая, компьютерная, цифровая.

Классическая стеганография — исторически сложившиеся методы сокрытия сведений, которые применяются в повседневной «реальной» жизни, другими словами, некомпьютерные методы. Например, по одной из версий древние шумеры наносили сообщения на глиняные дощечки, после покрывали их слоем глины и вновь наносили надпись, но уже не секретную. Также можно вспомнить запись сообщений на боковой стороне колоды карт, «жаргонные шифры», акrostихи и т. д.

Компьютерная стеганография, как видно из названия, включает в себя методы, основанные на особенностях конкретной платформы компьютеров, а также свойствах компьютерных форматов данных. Примерами компьютерных стеганографических методов являются следующие:

- Метод с использованием регистра букв. Его суть заключается в том, что каждый символ секретного сообщения переводится в байт-код. Затем у каждого символа скрывающего текста, которому соответствует единица сообщения, следует поменять регистр. Таким образом, можно зашифровать максимум $N/8$ символов, где N — количество символов в скрывающем тексте.
- Метод, использующий специфику файловых систем. Как известно, операционные системы для хранения файлов выделяют целое число блоков (для удобства адресации). Соответственно, для хранения маленьких файлов выделяется лишняя память, в которой как раз можно хранить необходимую информацию.
- Использование зарезервированных полей форматов данных также является отличным методом сокрытия сведений. Метод полагается на то, что большинство мультимедийных форматов имеют поля расширения, не используемые программой, как правило, они заполнены нулевой информацией.

На сегодняшний день самым важным и наиболее используемым направлением является цифровая стеганография. Это направление характеризуется тем, что необходимая информация внедряется и скрывается в цифровые объекты. К сожалению, цифровая стеганография накладывает некоторые обязательства, такие как сохранение целостности и аутентичности файла, поэтому обычно в качестве контейнеров (хранилищ данных) используют медиафайлы. Существуют следующие алгоритмы встраивания скрываемой информации:

- работающие с цифровым сигналом напрямую (метод LSB);
- внедрение скрытой информации (наложение секретного изображения, аудиофайла, текста поверх оригинала; часто используется для внедрения цифровых водяных знаков);
- использование форматов файлов (к примеру, запись в метаданные).

Одним из самых известных алгоритмов встраивания является метод LSB (Least Significant Bit — англ. наименьший значащий бит). Он основан на замене последних, незначущих бит в контейнере (графическом, аудио, видеофайле) на биты секретного сообщения. Метод опирается на низкий порог чувствительности человеческих органов. Например, изменение в 8-битном изображении двух последних бит приводит к изменению в цвете максимум на 3 бита, такие градации не отображают многие программы (считая их несущественными), не говоря уже о человеческом глазе.

В общем случае система маскирования имеет следующий вид (рис. 1):



Рис. 1 — Процесс сокрытия данных с помощью цифровой стеганографии

Пояснения к рис. 1:

- Контейнер — любая побочная информация (аудио, видео, изображение), служащая для сокрытия секретных сообщений.
- Встраиваемое сообщение — секретные данные, нуждающиеся в сокрытии.
- Встраивание — внедрение сообщения в контейнер, происходит на компьютере с использованием одного из методов стеганографии и ключа.
- Стего — стеганографический канал, канал передачи внедренного сообщения.
- Извлечение — получение секретного сообщения с использованием ключа.
- Ключ — сведения, необходимые для сокрытия/получения информации. Ключи бывают секретные и открытые. Системы с секретным ключом используют его как для встраивания, так и для извлечения. Системы с открытым ключом — только для встраивания, для извлечения используется независимый ключ.

В ходе исследования была произведена оценка устойчивости метода LSB с использованием изображений на языке Python. Процесс внедрения производился приведенным выше способом (см. рис. 1). В качестве контейнера использовалось изображение формата PNG. Для примера секретным сообщением было выбрано «success_1». Вмонтирование сведений производилось посредством резервирования 2-х последних бит каждого цвета, из которых состоит пиксель изображения в системе RGB. Для работы с изображениями и их трансформацией использовался модуль PIL языка Python. Анализ производился по двум параметрам: незаметность изменений контейнера при добавлении данных и валидность извлеченного сообщения. Примеры работы программы в зависимости от вида трансформации приведены в табл. 1.

Таблица 1. Оценка устойчивости LSB в зависимости от метода трансформации контейнера

Вид трансформации	Без трансформации	EXTENT	AFFINE	PERSPECTIVE
Ввод	success_1			
Вывод	1	ÿÿÿÿÿÿÿÿ	ÿÿÿÿÿÿÿÿ	ÿÿÿÿÿÿÿÿ
	2	ÿÿÿ8Àöÿ@	ÝXØÙ\Ü×ÿÿ	ÝÇqÇqÇ
	3	qV6ĭ_?ÿÿ	ß}÷ ß}÷ ß}÷	ÿÿÿÇqÇ
Видимость	—	—	—	—

Пояснения к табл. 1:

- EXTENT — извлекает область и помещает в новое изображение с данным размером (использовался как `resize`);
- AFFINE — аффинное преобразование;
- PERSPECTIVE — перспективное преобразование.

Полученные данные показали, что невооруженным глазом определить наличие скрытого сообщения невозможно. Однако быть уверенным, что сообщение будет доставлено в целостности и сохранности, нельзя. Таким образом, метод LSB является не устойчивым ко всякого рода трансформациям контейнера, т. е. метод хорош лишь при использовании статических контейнеров (не подверженных изменениям).

В заключение хотелось бы отметить, что стеганография не является должной заменой криптографии и полагаться только на нее не стоит. Но в сочетании с криптографическими методами стеганография становится еще одним рубежом защиты вашей бесценной информации.

Список использованных источников:

1. Грибунин, В. Г. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев. — М. : СОЛОН-Пресс, 2002. — 272 с.
2. Рябко, Б. Я. Основы современной криптографии и стеганографии / Б. Я. Рябко, А. Н. Фионов. — 2-е изд. — М. : Горячая линия — Телеком, 2013. — 232 с.
3. CitForum [Электронный ресурс]. — Режим доступа : <http://citforum.ru>.

ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ВОЗМОЖНОСТЕЙ ПРОГРАММНЫХ ПЛАТФОРМ NODE.JS, PYTHON И GO

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

Марчук М. С., Саскевич А. В., Цыркунов Д. А.

Стройникова Е. Д. – ассистент кафедры информатики

В данной работе исследованы криптографические возможности программных платформ Node.js, Python, Go по следующим критериям: 1) основные особенности программных платформ, причины их популярности, тенденции в развитии; 2) криптографические возможности, наличие собственных и сторонних библиотек шифрования и хеширования, их популярность и поддержка; 3) сравнительная характеристика основных криптографических алгоритмов, реализованных на данных платформах; 4) особенности применения на практике.

Основные особенности программных платформ, причины их популярности, тенденции в развитии.

Node.js – на данный момент одна из наиболее популярных программных платформ в сфере веб-технологий. Благодаря четко сформулированной идеологии, простоте, переносимости и достаточно быстрой скорости работы платформа нашла свое применение у таких крупных компаний, как Ebay, PayPal (банковские системы), Wikipedia (online-базы знаний), Mozilla и Yahoo (информационные технологии), New York Times и некоторых других СМИ и пр. В последние годы Node.js набирает все большую популярность и пользуется широкой поддержкой среди разработчиков, которые предлагают множество готовых реализаций различных видов алгоритмов и программных продуктов. Данные решения любой желающий может применить в своей работе.

Платформа Python занимает почетное место среди других программных платформ. Особенности одноименного языка программирования являются простота синтаксиса и наличие множества библиотек. Python зарекомендовал себя в основном как платформа для веб-разработки и машинного обучения, однако он применим и для более широкого круга задач. Список компаний, которые используют Python, длинный. Среди них Google, Facebook, Yahoo, NASA, Red Hat, IBM, Instagram, Dropbox, Pinterest, Quora, Яндекс, Mail.Ru. Но еще больший список можно составить из программного обеспечения, написанного с применением Python. Это BitTorrent, GIMP, World of Tanks, Battlefield 2, Blender, Sublime Text и пр. Стремительно растущее количество Python-специалистов говорит о том, что программная платформа, действительно, может быть применена в самых различных областях промышленности.

В отличие от вышеописанных платформ, которые основаны на интерпретируемых языках, платформа Go реализует компилируемый язык программирования, который по скорости работы приближается к таким языкам, как C и C++. Кроме того, платформа реализует экосистему управления модулями и библиотеками, близкую к экосистемам npm (Node.js) и pip (Python). В последнее время Go также набирает популярность, однако на данный момент популярность Go еще не соизмерима с популярностью Node.js и Python. Поддерживаемая командой разработчиков Google, платформа применяется такими компаниями, как Facebook, Dropbox, Google, GitHub, IBM, Twitter (информационные технологии) и пр.

Криптографические возможности программных платформ, наличие собственных и сторонних библиотек шифрования, их популярность и поддержка.

Для работы с алгоритмами шифрования и хеширования платформа Node.js предлагает модуль `crypto`, оснащенный довольно большим функционалом, реализующим такие алгоритмы, как DHE, EDH, AES, RC4, RC2, MD5, SHA256, SHA384, ГОСТ, RSA и др. У данного модуля есть популярный аналог `crypto-js`, реализующий алгоритмы SHA256, AES, MD5, RC4 и др. Помимо этого, есть еще несколько менее популярных криптографических модулей.

Платформа Python «из коробки» предлагает модули `hashlib` и `hmac`, предназначенные для вычисления хешей и цифровых подписей. Данные модули реализуют такие алгоритмы шифрования, как SHA1, SHA224, SHA256, SHA384, SHA512 и MD5. Кроме того, в модуле `zlib` присутствуют алгоритмы CRC32 и ADLER32. Из популярных сторонних библиотек можно отметить модули `ruscrypto` (реализует криптографические алгоритмы шифрования AES, ARC2, ARC4, Blowfish, DES, DES3, хеш-функции CMAC, HMAC, MD5, SHA, SHA1, RIPEMD и дополнительно несколько алгоритмов цифровой подписи) и `cryptography` (реализует стандарты X.509, хеш-функции SHA, MD5, RIPEMD, Whirlpool, алгоритмы шифрования Blowfish, ARC4, RSA и DSA).

Платформа Go также имеет богатый функционал для хеш-функций (MD4, MD5, SHA1, SHA224, SHA256, SHA384, SHA512, RIPEMD и др.) и алгоритмов шифрования (RC4, RSA, AES, DES, DSA). Данного набора хватает для большинства нужд, потому сторонние крупные и популярные библиотеки, реализующие дополнительный функционал, отсутствуют.

Сравнительная характеристика основных криптографических алгоритмов, реализованных на программных платформах.

Для сравнения были выбраны наиболее популярные и известные алгоритмы хеширования MD5, SHA256, SHA512 и алгоритмы цифровой подписи и шифрования RC4, RSA, DSA. В качестве основных критериев для сравнения алгоритмов выбраны скорость работы, основные параметры работы (разрядность, криптостойкость, тип). В табл. 1 и табл. 2 ниже представлены сравнительные характеристики скорости работы алгоритмов с использованием процессора Intel Core i7 2.4MHz 4GB RAM.

Таблица 1. Среднее время хеширования случайных данных размером 512кб

Алгоритм	Node.js	Python	Go
MD5	2.23 мс	0.98 мс	0.75 мс
SHA256	2.39 мс	1.82 мс	0.38 мс
SHA512	2.12 мс	2.14 мс	0.49 мс

Таблица 2. Среднее время генерации случайного ключа размером 1024 бита

Алгоритм	Node.js	Python	Go
RSA	320.5 мс	580.5 мс	60.7 мс
DSA	N/A	674.8 мс	N/A

Особенности применения криптографических алгоритмов на практике.

В соответствии с результатами исследования, при выборе конкретного алгоритма следует в первую очередь оценить, насколько критичными являются скорость и уровень защищенности алгоритма для данной задачи. Наилучшим вариантом является применение алгоритмов, реализованных на уровне ОС или оборудования. Помимо этого, следует обратить внимание на то, что гораздо безопаснее применять алгоритмы, реализованные «из коробки», – таким образом можно минимизировать вероятность появления ошибок и увеличить устойчивость к попыткам взлома.

Список использованных источников:

1. Node.js Crypto Documentation [Electronic resource] / Node.js v5.9.1 Documentation. – Node.js Foundation, 2016. – Mode of access: <https://nodejs.org/api/crypto.html>. – Date of access: 26.03.2016.
2. go/crypto documentation [Electronic resource] / The Go Programming Language. – Build version go1.6, 2016. – Mode of access: <https://golang.org/pkg/crypto/>. – Date of access: 27.03.2016.
3. Python hashlib documentation [Electronic resources] / Python. – Python Software Foundation, 2016. – Mode of access: <https://docs.python.org/3/library/hashlib.html>. – Date of access: 26.03.2016.
4. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 816 с.

СОБЫТИЙНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ОБРАБОТКИ НАБОРОВ ДАННЫХ

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

Сафонов А. А.

Ганжа В. А. – кандидат физико-математических наук, доцент

С ростом объёмов информации, требуются новые способы их обработки. Пользователям требуется быстрый доступ к новой информации и возможность получения данных в реальном времени. Это стимулирует разработчиков создавать отзывчивые интерфейсы и модели для обработки.

Разработанная модель позволяет создавать и преобразовывать наборы данных и реагировать на их изменения. Она направлена на повышение качества кода при написании решений для обработки данных. В текущей реализации только на платформе .NET Framework.

На рисунке 1 представлена схема обработки запроса на вставку в набор чисел:

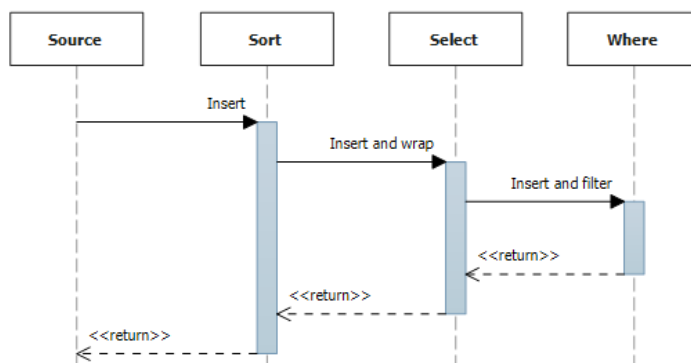


Рисунок 1 – Схема запроса на обновление дочерних наборов

Код на языке C#:

```

var resultCollection = source.SortRc(...).SelectRl(...).WhereRl(...);
source.Add(number);
    
```

Листинг 1 – Пример построения обновляемого посредством событий набора на языке C#

В результате исполнения кода из листинга 1, мы получим новый набор данных, который будет реагировать на изменения исходной коллекции. Операции сортировки, преобразования и фильтрации будут применяться к новым элементам набора и реагировать на изменения текущих.

В рамках проекта создана библиотека с базовым набором функций, которые повторяют функциональность операций языка SQL. Например: Select, Where, Join и т. д.

Для создания рабочего проекта использовался язык C#. В ходе исследования были изучены материалы, предоставленные Microsoft Research по построению систем, основанных на событийных моделях

Таким образом, была разработана система, реализующая спектр операций по обработке данных учитывая принципы реактивного программирования. Данный проект может применяться при построении пользовательских интерфейсов, где пользователю важно получать актуальную информацию в кратчайшие сроки.

Список использованных источников:

1. Lee Campbell, Introduction to Rx - 2012
2. Джеффри Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# - 2015
3. [Электронный ресурс] LINQ (Language-Integrated Query) - <https://msdn.microsoft.com/ru-ru/library/bb397926.aspx> - 2016

СЕКЦИЯ «ФИЛОСОФИЯ И ЖИЗНЬ»

ПРОБЛЕМЫ ФОРМИРОВАНИЯ ЛИЧНОСТИ В СОВРЕМЕННОМ ОБЩЕСТВЕ

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

Аксёнов О. Д., Воронов А. О.

Володин В. М. – преподаватель каф. философии

Вопрос личностной самоорганизации человека в наше время настолько не востребован, что уже с раннего возраста нормы культурно-нравственного поведения замещаются на навязываемые обществом извне. В данной работе анализируются истоки формирования человека как личности, препятствующие этому причины у современного человека, и предлагаются возможные решения этой проблемы для реализации человеком себя как полноценной личности.

Проблема становления и развития человека как гармоничной личности – одна из наиболее актуальных проблем на сегодняшний день. Её решение заключается не только в выявлении закономерностей и определенных изменений социализации человека на разных возрастных этапах, но и в раскрытии общих принципов индивидуального воспитания личности на самых ранних этапах её становления. Проблема личности в психологии до сих пор не может быть до конца исследована, так как есть одна существенная трудность, – между людьми много различий, которое проявляется, в частности, в поступках: сложных, противоречивых, непредсказуемых. Эти различия усложняют решение данной проблемы. Сегодня проблема природы личности приобретает особую значимость. Ее решения может предотвратить негативные изменения в экономике, политике, социальных процессах, которые являются следствием поведения людей.

Сегодня, в связи со всеобщим упадком пропаганды этической культуры и воспитания наблюдается торможение развития социального общества. Все духовные и нравственные ценности, которые служили для человечества как иммунитет и защита от внешнего воздействия, теряют свою востребованность и культивацию, именно поэтому современный молодой человек становится так ограничен и попадает в зависимость от мнения социума и средств массовой информации, включая интернет.

Существует множество определений понятия личность, но, на наш взгляд, большинство из них являются не совсем верными и не отображают действительную сущность данного термина. Большинство из проанализированных нами источников приводят следующую формулировку: «Личность – сознательный индивид, занимающий определенное положение в обществе и выполняющий определенную общественную роль» [1, с.48]. В данном случае понятие «личность» определяется через понятие «индивид», поэтому возникает не совсем верное понимание и путаница между этими понятиями.

Индивид – это отдельный представитель биологического вида homo -sapien- (человек разумный), представитель человеческой общности [1, с.118]. К индивидуальным свойствам человека относятся: пол, телосложение, характер, ценностные ориентации. Когда рождается индивид, он постепенно начинает усваивать человеческую культуру, приобретать отличительные особенности, самоутверждаться среди других людей и, соответственно, становится личностью. Как нам кажется, наиболее развёрнутую для понимания формулировку привел доктор педагогических наук Тарантей Виктор Петрович: «Личность – человек, субъект отношений и сознательной деятельности, способный к самопознанию и саморазвитию; устойчивая система социально значимых черт, отношений, установок и мотивов, характеризующая человека как члена общества» [2, с.54]. Мы считаем, что данное определение включает в себя ряд таких качеств человека как: независимость суждений, непоколебимость взглядов на мир и общество в целом, свободу мышления, культурность, целеустремленность, ответственность, огромное желание к самосовершенствованию и самореализации.

Современная молодёжь пытается выделиться из толпы посредством внешних преобразований, путем «демонстрации на показ», т.е. размещением содержания своей личной жизни в социальных сетях, таких как «инстаграм», «вконтакте», «твиттер», «фэйсбук» и других. При этом большинство таких людей считают себя творческой личностью, выкладывая в интернет различные фотографии с бессмысленным содержанием, но это не так. Подростками руководит наивная детская мысль: «если я буду не как все, значит, я и есть уникальная личность». В основном молодые люди поступают так, потому что, с одной стороны, следуют современной моде «интернет-демонстрации», а с другой, – хотя выделиться из безликой толпы, но не знают как, потому что не нашли себя и не могут реализовать свой потенциал, который, безусловно, есть в каждом человеке. Но почему не выделиться посредством преобразований в себе самом? Очевидно, что это слишком сложно было всегда и особенно сложно для современного общества. На данную тему мы нашли высказывание Николая Бердяева: «Быть личностью, быть свободным есть не легкость, а трудность, бремя, которое человек должен нести. От человека сплошь и рядом требуют отказа от личности, отказа от свободы и за это сулят ему облегчение его жизни. От него требуют, чтобы он подчинился детерминации общества и природы. С этим связан трагизм жизни.» [3, с.110].

С самого раннего возраста ребенок уже страдает от избытка недифференцированной информации, поступающей извне, перед которой он открыт и незащищен. Интернет, телевидение, радио, мобильная связь – всё доступно современному ребенку и именно поэтому это очень опасно, так как несформированная и неподготовленная к реальной, а не виртуальной жизни личность, не может разграничивать и фильтровать информацию. Формирование личности, т.е. организация человека начинается с рождения, поэтому для

ребенка семья будет первым воспитателем, на которого он в дальнейшем развитии будет опираться, что станет началом становления его индивидуальности.

Но всегда ли семья является основополагающим фактором в становлении личности? Собрав статистику, можно смело сказать - нет! В настоящее время среди несовершеннолетних правонарушителей достаточно много подростков, которые характеризуются окружающими положительно, воспитываются в полных и благополучных семьях, однако совершают правонарушения под влиянием конфликтных или проблемных жизненных ситуаций. По данным социологического опроса, проведенного Генеральной прокуратурой, Научно-практическим центром проблем укрепления законности и правопорядка Генпрокуратуры и представительством Детского фонда ООН (ЮНИСЕФ) в Беларуси, среди несовершеннолетних правонарушителей 42,6% из них воспитывались в семье с отцом и матерью, а остальные проживали либо с одним родителем (33,1%), из них с матерью — 29,9%, с отцом — 3,2%, либо с другими родственниками (3,6%) или опекунами (2,3%). В детском доме (интернате) воспитывались 4,5% опрошенных подростков [4]. Как мы видим, 42,6% правонарушителей – подростки из полных семей, это наивысший процент, поэтому следует понимать, что большинство людей, которые говорят, что в подростковых преступлениях виноваты только родители, не совсем правильно рассуждают. Конечно, родители являются началом развития ребенка, дают ему старт, основу для формирования, но в нашей современной ситуации немалая доля вины лежит на характере социальной среды, которую, порой, несформированная личность не может преодолеть и попадает под влияние, теряя свое «Я». Поэтому так и получается, что при попадании под воздействие толпы, человек и становится частью её, психологически начинает думать и вести себя в соответствии с главной целью – быть как все.

Из вышесказанного вытекает следующая задача - проблема самовоспитания. В настоящее время данный вопрос встал очень остро, ведь у современной молодежи нет стремления к самосовершенствованию: другие интересы. «Самовоспитание – это процесс усвоения человеком опыта предшествующих поколений посредством внутренних душевных факторов, обеспечивающих развитие» [5, с.158]. Но нужно понимать, что без участия самих детей в этом процессе - это невозможно. Как следствие, без самовоспитания человек не может и самообразовываться, что имеет крайне негативные последствия, так как, на наш взгляд, лучшее образование – это самообразование, а не силовое давление, навязывающее обучение.

Необходимо разобраться: влияет ли культура на становлении личности? Следует понимать, что культура формирует внутренний мир человека, раскрывает содержание его личности. Она воспитывает, просвещает, дает возможность познать окружающий мир не зрительно, а через слово, звук, цвет. Но кто же такой культурный человек? Тяжело ли им стать? И нужно ли вообще стремиться к этому? Культурный – это вовсе не тот человек, которого принуждают к духовным ценностям. На наш же взгляд это личность, которая старается формировать себя в процессе созидательной деятельности и получает от этого радость, эстетическое удовольствие и черпает энергию из поэтических строк, картин, музыки. Искусство в целом воздействует на человека психологически очень сильно, оно пробуждает в человеке ответное творческое чувство. Однако, если этого не происходит, то, на наш взгляд, такому человеку стремиться стать культурным не зачем, потому что тогда будет искусственная имитация.

По мнению кандидата философских наук, доцента кафедры философии БГУИР, Габруся Ивана Федоровича, которое он высказал в своей недавней работе «Торжество-вампира культуры», что «современная культура «европейского человечества» отбрасывает стыд, списывает его как устаревший пережиток. Она в прямом смысле «заголилась», как предвидел еще в XIX веке Достоевский.

По результатам проведенных нами опросов среди студентов БГУИР, главным препятствием в самовоспитании себя в наши дни является размытость в представлениях о таких понятиях, как нравственность, культурность, личность. В опросе было задействовано более 50 студентов, и мы убедились, что у молодежи практически полностью отсутствует представление о вышесказанных понятиях. Мы задавали два вопроса: «Как вы понимаете понятия личность?» и «Считаете ли вы себя личностью?» Развёрнутый ответ дали единицы, а определения других сводились к следующему:

• каждый по-своему личность(11%); это позитивный и творческий человек(14%); мне кажется, что все - личности(18%); мне «до лампочки»(12%); не знаю и не хочу знать(15%); никогда не задумывался(9%); индивид, человек есть личность(21%).

Вывод: стоит объединить усилия в изучении и воспитании подлинной личности всем, кто не желает исчезновения с лица Земли сильных, независимых, и самое главное – разумных и мыслящих людей. Чтобы пробудить в человеке потребность быть личностью, необходимо пропагандировать, что не каждый человек вправе называться ею. Человек должен пересмотреть свое «Я» и сказать себе честно, соответствует ли он приведенным выше критериям личности. И снова хочется вспомнить слова Николая Бердяева, что личность есть боль, а борьба за становление личности болезненна, ведь эти люди берут на себя «грязь» и идут против течения, поэтому для развития себя как личности нужно произвести трудоемкую работу по преодолению всех моделей поведения, которые навязываются массой извне. А вы - личность? Начните с себя!

Список использованных источников:

1. Личность и индивидуальность [Электронный ресурс] - Режим доступа: <http://psylist.net/obh/00066.htm>. Дата доступа 26.11.2015.
2. Тарантей В. П., Тарантей Л. М. Сборник качественных задач: Общая педагогика. – Тула, 2011.
3. Бердяев Н.А. Самопознание. Л.: Лениздат, - 1991
4. Дети-преступники: осудить или пожалеть? [Электронный ресурс] Режим доступа: http://naviny.by/rubrics/society/2012/6/1/ic_articles_116_178034/. Дата доступа: 26.09.2015.
5. Личность и индивидуальность [Электронный ресурс] - Режим доступа: <http://psylist.net/obh/00066.htm>. Дата доступа 28.11.2015.

ФИЛОСОФИЯ СМЕРТИ ИЛИ ЕСТЬ ЛИ ЖИЗНЬ ПОСЛЕ СМЕРТИ?

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

Крупская Ю. И., Батура Н. А.

Володин В. М. – преподаватель

Человек отличается от всех других существ наличием сознания. Поразмыслив над этим, можно сделать вывод, что человек осознает свою смертность, и смерть приобретает в его глазах определенную значимость. Многие люди не раз задавали себе один и тот же вопрос: «Что будет со мной после смерти?». Однако, можно смело заявить, что все мысли и идеи касательно этой (и не только этой) темы из далекого прошлого, из далекого «до», повторяются. Даже не так, – дублируются. Возможно, в далеком или недалеком будущем будет какой-то кардинально новый подход. Поразмыслив, приходишь к выводу, что все подходы уже имели место отразиться, ведь существовало и множество положительных, и множество отрицательных мнений по факту существования жизни «загробной». Остаётся лишь выбрать, на чьей ты стороне.

Ни на солнце, ни на смерть нельзя смотреть в упор. (Франсу де Ларошфуко)
Когда ты постигнешь все тайны жизни, то будешь стремиться к смерти,
ибо она не что иное, как еще одна тайна жизни. (Халиль Джебран)
Три проклятых вопроса - это «откуда мы, кто мы и куда мы идем?» (Ф. Достоевский)

Единственное, что без сомнения можно утверждать: то, во что ты веришь – это ты сам. Так, выбрав для себя ориентиром существование после смерти, ты, вероятнее всего, ведешь жизнь соразмерную и обдумываешь свои поступки, или после какой-то ситуации «глубокого смысла» начал придерживаться жизненных правил. Ответив на поднятый нами вопрос отрицательно, ты, вероятнее всего, опрометчив в поступках.

Рассмотрим типичного человека нашего времени. Множество религиозных учений, этических норм, эстетических ценностей, покинутых нашими предками. Кажется, что все просто! Выбирай идею или школу, взгляды которой тебе наиболее близки. А если тебе близки идеи различных школ, и ты не относишь себя к определенной, то, что тогда? Если правдой окажется совсем не то, чего ты придерживаешься, совсем не то, что думают другие, то, как поступить?

Жизнь после смерти — это один из главных вопросов религии, философии и науки о продолжении человеческой жизни. Это вопрос, который не покидает задумывающихся людей и требует ответа. В большей степени это обусловлено верой в вечность души, бессмертие. Люди спорят и высказывают свое «но» по поводу основных идей, озвученных кем-то другим. Перечисляя основные, можно выделить следующие:

1. Воскрешение мёртвых — избранные люди будут воскрешены богом после смерти;
2. Реинкарнация — душа человека возвращается в материальный мир в новых воплощениях в зависимости от земной жизни человека;
3. Посмертное воздаяние — после смерти душа человека попадает в «Ад» или «Рай» в зависимости от земной жизни человека;
4. Конечность бытия — душа человека, накопившая за жизненный период энергию, рассеивается, растворяется в небытии.

Если не вдаваться в подробности, что есть жизнь, а что есть смерть (ведь это тоже вопросы, которые требуют философских рассуждений, а не ответа «наобум»), из этих однозначно антонимичных определений понятно лишь то, что «смерть» не обесмысливает «жизнь», что «смерть» подчеркивает ценность жизни и её смысл. Жизнь же, в свою очередь, освещена светом сознания.

Что обуславливает эпоху? Некоторые ценности, учения, нормы поведения и многое другое. И не удивительно то, что целые эпохи связаны и в чем-то схожи, а иногда даже копируют друг друга. Все наши взгляды и представления и назвать-то «нашими» сложно, подавняя их часть уже успела отразиться в истории человечества. Первые пункты ответов на волнующий нас вопрос – ответы, которые характерны для людей, живших, когда неповторимость человека и его судьбы обязывали задумываться о целесообразности совершенного и нести ответственность, причем не только в «земной» жизни, но и в «загробной». В основе воспитания лежал страх перед наказанием за грехи, страх за наказание в «Аду».

В Древности, уже на самых ранних этапах развития человеческого общества, складываются особые нормы, определяющие формы общения с умирающим человеком, способы захоронения. В мифологии в то же время происходит попытка понимания смысла смерти, возникают слишком нереальные идеи о возможной связи живых и умерших, мифы о существовании «после». Религия же в свою очередь в большей частности затрагивает то, что связано с культом предков, влиянием «мертвого» на «живое». Сюда можно было отнести как переселение душ, так и бессмертие в потустороннем загробном мире. Одним из примеров могут являться древние египтяне, которые рассматривали жизнь как этап на пути к загробному существованию. Они ставили на главное место такие вещи, как постройка пирамид (мест захоронения умерших) и бальзамирование трупов.

Девиз античных философов-стоиков «Помни о смерти!» становится толчком, стимулом нравственного поведения. Христианство, иудаизм, ислам и другие мировые религии говорят о вере в вечную жизнь (жизнь после смерти), тем самым стараясь заменить его страхом перед наказанием за совершенные грехи при жизни.

Безусловно, эмоции породили веру в жизнь “после”. Но дело тут не только в страхе. Вот, например, еще один представитель, который выразался по мной заданному вопросу весьма однозначно, – епископ Бирмингемский. Он доказывал, что Вселенная создана и управляется разумной целью, и, как итог, было бы неразумно, создав человека, позволить ему исчезнуть с лица земли. Кант считал, что нужно вести себя таким образом, как будто нас ожидает иная жизнь, и при вступлении в нее важное значение будет иметь моральное состояние, с которым мы закончим жизнь нынешнюю. Такой взгляд на проблему смерти и бессмертия человека, противопоставляемый религиозным подходам, получило широкое распространение в философии и обыденности девятнадцатого-двадцатого века.

К «конечности бытия» пришли в те эпохи, когда люди стали углубляться в философские проблемы (в том числе понимаю смерти). Логичное примирение человека со смертью – вот цель, к которой стремились, выдвигая порой кардинально противоположные идеи.

Сократ, Платон и Аристотель отстаивали тезис о бессмертии души. Эпикур пытался освободиться от страха смерти, доказывая, что душа гибнет вместе с телом, поэтому человеку не стоит бояться смерти и стоит воспринимать ее как должное.

Марк Аврелий считал: «Нет никакого различия между жизнью в три дня или в три человеческие жизни, то есть, нет разницы между долгой жизнью и короткой». Философ призывал людей не бояться смерти, стремиться к ней, как естественно стремятся все живые существа, «поскольку и смерть является таким же закономерным событием, как и молодость, старость, рост, рождение... Смыслящий в жизни человек не должен отвергать смерть, ненавидеть ее, нужно подходить к ней без высокомерия, ожидать ее как обычное природное явление».

Материализм эпохи Возрождения и Нового времени с отрицанием идеи вечности души и значимости смерти сформировал представление «конечности бытия». Особую роль в этом процессе сыграли биология и медицина.

Одним из примеров может являться такая система, как «иммортализм». Это система современных взглядов, касающихся данной проблемы. Она основана на стремлении достичь бессмертия или, как минимум, отдалить физическую смерть опираясь при этом на достижения медицины, точных естественных и технических наук. На возможность решения данной проблемы указывает факт существования практически нестареющих многоклеточных организмов.

Современная наука движется в сторону радикального увеличения продолжительности жизни. Благодаря изобретению антибиотиков и другим успехам медицины в течение XX века средняя продолжительность жизни в развитых странах выросла почти в два раза.

“Наука не дает оснований для надежд на преодоление смерти, но и нет закона биологии, который утверждал бы обязательную конечность жизни индивида”, - говорил американский физик, лауреат Нобелевской премии, Фейнман.

Так религия, философия и наука пытаются примирить нас со смертью.

... Вероятно, дочитав до этих строк, возникнет желание узнать мое мнение, мой ответ на волнующий всех вопрос и, наконец, подвести итог. Возможно, мой ответ обусловлен обществом и последними нововведениями в систему общественных взглядов.

Меня эти околосмертные переживания не покидают. Я задаю этот вопрос раз за разом, особенно четко вспоминаю один момент: когда-то, в курсе физики, учителем были сказаны слова о том, что ничего не исчезает бесследно. Тело разлагается, становясь “земным”. Другое дело - душа. Ее существование, как мне кажется бесспорно - не зря отрицать то, что в ней закладывается опыт, жизненная энергия, которая не отправляется ни в Ад, ни в Рай, которая не перерождается, как верили индусы. Душа просто погибает, растворяется в бытии с каким-то временем. А до этого происходит то, что не сможет описать ни один человек. Ни потому что этого никто не знает (были те люди, которые находились в клинической смерти и видели какой-то свет), а потому есть что-то непостижимое для никого. Тот момент, между жизнью и исчезновением души может быть сравним со сном, которого мы не помним на утро - мы лишь помним, что он был, что он есть, но о чем он, что там было?... Это вопросы, которые мы можем задавать, но на которые никто не может ответить.

“Загробный мир, загробная жизнь... О таких вещах я решил не думать... Сколько тут ни думай, правды всё равно не узнаешь, а если и узнаешь — не проверишь никак. Только время зря потратишь. Все так или иначе мы узнаем потом”, – говорил Харуки Мураками, рассуждая над вопросом, который подняли спустя много-много лет после его смерти. Или все-таки “не-смерти”? Он считал это чепухой, над которой не стоит задумываться в данный момент. А может это и есть самый важный вопрос?

Список использованных источников:

1. История философии: Запад-Россия-Восток (книга первая. Философия древности и средневековья).- М.:Греко-латинский кабинет', 1995. - 480 с.
2. Раймонд Моуди. Жизнь после жизни 1976. – электронный источник: http://www.lib.ru/MOUDI/moudi.txt_with-big-pictures.html

ЧЕЛОВЕК КАК СИНЕРГЕТИЧЕСКАЯ СИСТЕМА

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

Бычко А. А.

Ермолович Д. В. – канд. филос. наук, доцент

Синергетика – междисциплинарное направление науки, изучающее принципы самоорганизации в открытых сверхсложных системах. Человек и человеческое общество могут выступать в роли примеров, демонстрирующих эволюционное возникновение и последующее развитие диссипативных структур в материальном мире.

Несмотря на то, что изначально синергетический анализ появился в физике, где применялся при исследовании открытых термодинамических систем, которые, вопреки ожиданиям, проявили склонность к самоорганизации и образованию устойчивых структур, законы синергетики распространяются на любые явления материального мира, которые отвечают ряду требований [1]:

- существование в достаточной степени элементов и составляющих частей, способных самоорганизовываться;

- нахождение самоорганизующегося объекта в открытой среде и обмен веществами, энергией и информацией с окружающей средой;

- нахождение этого объекта в нестабильности;

- развивающийся объект не должен ограничиваться только причинно-следственной необходимостью, но также требует учета и таких процессов, как спонтанное развитие, хаос [2, с. 47].

Синергетика специально названа междисциплинарной концепцией, так как сходные правила самоорганизации проявляют не только физические, но и практически любые открытые системы: в химии, биологии, информатике. Л. В. Лесков приводит три основных принципа теории самоорганизующейся системы [2, с. 44-45]: открытости, нелинейности и когерентности.

Примером синергетического анализа в биологии является объяснение антропного принципа Ричардом Докинзом [3]. Его рассуждения опираются на самую распространенную в настоящее время эволюционную концепцию – дарвинизм, или отбор наиболее приспособленных видов. Другой его концепцией является предположение о целенаправленном развитии живых организмов путем упражнения тех или иных своих частей (ламаркизм).

Человек также является синергетической системой, его существование подчинено общим законам мироздания [1, 4, с. 158]. Примером может служить биологическая эволюция вида *homo sapiens*, что соответствует воззрениям австрийского биолога Людвиг фон Бергаланфи, утверждавшего изоморфизм законов, управляющих функционированием системных объектов. Помимо эволюции генов (биологической), люди смогли запустить собственный эволюционный процесс – развитие мемов и мемокомплексов [5]. Мемами являются любые элементы материальной и нематериальной культуры, например, особенности языка, технологии, религиозные верования или произведения искусства. Все они претерпевают изменения в полном соответствии с общими принципами эволюционизма. В качестве примера можно проследить возникновение и развитие мировых религий, изменение их догматики и культуровой обрядности с течением времени. Уникальным с точки зрения эволюционной парадигмы феноменом является образовательный процесс. «Сама возможность обучения, – как пишут Е. Н. Князева и С. П. Курдюмов, – означает существенное преимущество человеческого существа в достижении того, для чего в ходе эволюции, по Дарвину, потребовались бы тысячелетия неопределенных мутационных изменений. Культурная эволюция, в противоположность историческому развитию биологических видов, является ламаркианской по своему характеру, т. е., выражаясь языком биологии, приобретенные индивидом в течение его жизни изменения, его знания и накопленный опыт, наследуются, передаются следующим поколениям» (цит. по [6]). Еще одним наглядным примером, демонстрирующим развитие человеческого общества, является постепенная смена доминирующего способа производства и технологий.

Открытым остается вопрос о возможных аттракторах, к которым стремится развитие человечества в процессе исторического развития. Помимо идеи построения утопического общества, характерной уже для культуры древности, в последнее время все большее развитие получают идеи антиутопии и технологической сингулярности [7]. Вопрос о выборе целей и средств человеческого развития должен быть детально исследован, так как он имеет непосредственное отношение к настоящему и будущему каждого человека.

Список использованных источников:

1. Рахманов, Б. У. Синергетический подход к самоорганизации человека / Б. У. Рахманов // *European science*. – 2015. – №1(2).
2. Лесков, Л. В. Футуросинергетика: универсальная теория систем / Л. В. Лесков. – М.: Экономика, 2005.
3. Докинз, Ричард. Эгоистичный ген / Ричард Докинз, пер. с англ. Н. Фоминой. – М.: АСТ:CORPUS, 2013.
4. Василкова, В. В. Порядок и хаос в развитии социальных систем / В. В. Василкова. – СПб, 1999.
5. URL: <https://ru.wikipedia.org/wiki/Мем> (дата обращения: 20.03.2016).
6. Рабош, В. А. Синергетика образования человека / В. А. Рабош // *Успехи современного естествознания*. – № 3. – М.: Изд-во Академии Естествознания, 2004.
7. URL: https://ru.wikipedia.org/wiki/Технологическая_сингулярность (дата обращения: 20.03.2016).

АГРЕССИВНОСТЬ ПОДРОСТКОВ КАК СОЦИОКУЛЬТУРНОЕ ЯВЛЕНИЕ

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

Вересковский С. В.

Бархатков А. И. – канд. филос. наук

Биологической основой агрессивности, наблюдающейся в подростковый период, выступают происходящие в этот период гормональные процессы. В то же время, биологический фактор усугубляется социальным, так как именно недостаток воспитания в рамках семьи и других социальных институтов является причиной неприкрытого проявления подростковой агрессии.

Для начала хотелось бы отметить, что *подростки – это будущее страны*, и анализ проблем подростковой агрессии есть ключ к цивилизованному обществу в будущем и, следовательно, процветанию страны и нации.

Агрессия – мотивированное деструктивное поведение, противоречащее нормам сосуществования людей, наносящее вред объектам нападения, приносящее физический, моральный ущерб людям или вызывающее у них психологический дискомфорт. По Эриху Фромму, агрессия подразделяется на несколько подвидов. В результате изучения поведения подростков, можно сделать вывод, что чаще всего встречающийся вид подростковой агрессии — реактивная агрессия. Реактивная агрессия — защита жизни, свободы, достоинства, а также собственного или чужого имущества. Подростковый возраст — период в развитии человека, переходный этап между детством и взрослостью. Согласно терминологии Фонда Организации Объединенных Наций в области народонаселения (ЮНФПА), подростки — лица в возрасте 10—19 лет (ранний подростковый возраст — 10-14 лет; поздний подростковый возраст — 15-19 лет). По данным ООН, на 2011 год в мире насчитывалось более миллиарда подростков.

Основным предметом данного доклада является группа раннего подросткового возраста (10 – 14 лет), т.к., на мой взгляд, особо ярко подростковая агрессия наблюдается как раз в этом возрасте. Прежде всего, стоит вспомнить, что всё поведение человека есть взаимодействие химических веществ организма в целом и нервной системы, в частности. Чем же отличается нервная система взрослого человека от нервной системы подростка? В большинстве случаев агрессивное поведение подростков объясняется переизбытком в организме гормона – тестостерона, который активно вырабатывается в результате полового созревания. Данный гормон повышает активность человека. В результате воздействия тестостерона, у подростка может повыситься познавательная активность, однако также вероятно повышение случаев агрессии.

Анализируя проблему агрессивности подростков, следует поставить вопрос: а всегда ли подростки были агрессивны? В биологии ли проблемы? Может это недостаток воспитания? Послушав некоторые мнения, приходится сделать следующий вывод: агрессивность подростков была всегда, но раньше она не проявлялась из-за воспитания. Теперь же, в эру информационных технологий, подросток может абсолютно безнаказанно оскорбить либо нахамить незнакомому человеку. Но зачем? Самоутвердиться! Самоутвердиться за счёт унижения других людей. Это есть недостаток воспитания как в социальном институте семьи, так и со стороны института школы.

Из этого следует следующий вопрос: чем нынешнее воспитание отличается от воспитания образца, к примеру, 1970 г.?

Для начала рассмотрим информационный фактор. СМИ больше не рассказывает о том, как “у нас всё хорошо и прекрасно”. СМИ рассказывает о разбойных нападениях, войнах, восстаниях, терактах, о боли невинных людей. Если разум взрослого человека (и то не всегда) способен экранироваться от подобных новостей, то разум подростка (по сути ещё ребёнка) впитывает всю эту агрессию, а потом выливает её в своём поведении.

Следующий фактор — социальный. Стоит заметить, что есть различные группы населения. Обычно подростки у которых родители слишком заняты работой или того хуже, алкоголем, и у которых нет времени на воспитание ребёнка, впитывают нормы поведения из СМИ и улиц. Родители не понимают, что каждая ссора откладывается в поведении подростка. Подростковый возраст – процесс формирования личности человека.

В качестве примера подростковой агрессии возьмём поведение подростка в социальных сетях. Чаще всего, когда человек в каком-то сообществе выражает своё мнение в комментариях (в месте, как раз таки предназначенном для выражения собственного мнения), и данное мнение не совпадает с мнением подростка, в частности, школьника (особенно если ваше мнение и мнение школьника являются взаимоисключающими), подросток, не пытаясь даже аргументировать своё мнение, начинает в цензурной форме унижать человека тем самым пытаясь доказать, что его точка зрения является по каким-то причинам лучше. Разумеется, в современном и цивилизованном обществе подобное поведение является по меньшей мере недопустимым.

В результате всего вышперечисленного, следует сделать вывод, что агрессивное поведение подростков вызвано недостатком воспитания. Это либо недостаток воспитания в семье, либо низкокачественная воспитательная работа в школе, либо все эти факторы вместе взятые.

В ЧЁМ СЧАСТЬЕ ЧЕЛОВЕКА?

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

Верещако И. Н.

Володин В. М. – преподаватель каф. философии

Что есть счастье для человека? Ответ на этот вопрос является наполнением смысла его существования. Ответов много и все они разные. Как определиться с правильностью выбора, что считать счастьем и как не ошибиться, чтобы прожитая жизнь не оказалась прожитой впустую?

У каждого человека имеется свое собственное представление о счастье. Для одних счастье состоит в материальных ценностях, карьере, благосостоянии и финансовом благополучии. Конечно, за деньги можно купить дом, мебель, но не домашний уют и очаг; разнообразную еду, но не аппетит; развлечения, но не радость от общения с родителями, друзьями; нельзя купить дружбу, любовь, здоровье и прочее. Многим людям их работа не доставляет никакого удовольствия, но они ходят на нее, делают все, что от них требуется ради денег. А другие наоборот – имеют небольшую зарплату, но при этом работа, коллеги добавляют счастья в общей копилке. Люди не бессмертны и материальные ценности с собой в могилу не захватят, но вряд ли человек, который не имеет средств к существованию, назовет себя счастливым.

Для других людей счастье – это духовное богатство (знания, культура, вера), но наличие образования не гарантирует наличие разумности. Духовно развитый человек – это человек способный мудро, адекватно и гармонично мыслить и действовать для блага своего и других людей. Как говорил один из главных героев фильма «Мирный воин»: «Нет выше цели, чем помогать людям». Для третьих счастье – это физическое здоровье. Конечно, если человек неизлечимо болен, то никакие деньги, ни наличие духовного богатства не смогут заменить здоровья. Но когда говорят, что для счастья главное быть физически здоровым, то в пример можно привести здоровых людей, которые кончают жизнь самоубийством, а также высказывание Сократа: «Здоровье – не всё, но всё без здоровья – ничто». Но в тоже время есть много счастливых людей, которые имеют большие проблемы со здоровьем, финансовым положением и т. д. Ещё одно очень популярное мнение, что счастье в простых вещах: детях, семейной жизни, простой и вкусной еде. Например, обычная семейная пара с ребёнком, живущие на среднюю зарплату, когда мама не работает, так как ухаживает за ребёнком, а отцу приходится чуть ли не жить на работе, для того, чтобы обеспечить семью. Счастья тут не особо много и увидеть его могут только более-менее состоятельные люди.

Может показаться, что счастье – это то, чего не хватает в данный момент человеку или совсем нет в его жизни. Одни считают, что счастье — это подарок судьбы, а другие утверждают, что счастье целиком зависит от человека, от его воли и желания. Но счастливым может быть только тот человек, который самостоятельно делает свой выбор. Большинство людей хотят быть счастливыми, стремятся к счастью, создают необходимые условия для достижения этой цели. Многое зависит от самого человека, так как только он сам может сделать себя счастливым или несчастным. Ведь счастье по сути – это радость, положительные эмоции. Если мы внимательно присмотримся к окружающим людям, то увидим, что все они без исключения только и заняты тем, что пытаются получить удовольствие. Различие заключается только в том, посредством чего человек пытается достичь положительных эмоций. Не обязательно чтобы человек сознательно ставил для себя личное счастье в качестве главной цели. Счастье воспринимается человеком в качестве главной цели на подсознательном уровне, он этого может не понимать и даже отрицать. Главное не перепутать счастье с чем-то другим, заметить его и не упустить и, по возможности, делится с другими, ведь от этого оно только приумножается.

Большинство людей в гонке по жизни перестают замечать красоту окружающего нас мира, людей; в ежедневной суете забывают о многих простых вещах. А ведь на самом деле многого для счастья и не надо – это просто то, что ты живешь в этом мире и радуешься каждому дню. Счастье – это когда сам и твоя семья здоровы; когда тебя понимают, поддерживают родители и друзья; когда у тебя есть возможность учиться или работать. Можно перечислять бесконечно, но достаточно научиться ценить каждый момент жизни, быть позитивным и в будущее смотреть с оптимизмом. Счастье, это не материальные блага, не духовное богатство и не простые вещи, а отношение ко всему этому. Счастливым может быть бедный, бездетный и ничем не занимающийся человек, по-своему, но всё-таки счастлив! Тут важно выделить следующие факторы гармонии, без которых невозможно ощутить себя в полной мере счастливым: физическое и психическое здоровье; семья и близкие люди; учеба или работа; душевное равновесие.

В заключении следует сказать, что каждый человек достоин того, чтобы быть счастливым, несмотря на то, что счастье у каждого свое. А также хочется, чтобы люди не мешали быть счастливыми другим, ведь зависть, злоба, вражда не сопутствуют счастью. Я согласен с высказыванием Ф.Э. Дзержинского: «Счастье – это не жизнь без забот и печалей, счастье – это состояние души». Мне хочется закончить словами Хелен Келлер: «Когда одна дверь счастья закрывается, открывается другая; но мы часто не замечаем ее, уставившись взглядом в закрытую дверь».

Список использованных источников:

1. www.yourfreedom.ru/v-chem-schaste-cheloveka/
2. vot-status.jimdo.com/цитаты.../цитаты-великих-людей-5/
3. www.sunhome.ru/psychology/13336/

НРАВСТВЕННЫЕ ЦЕННОСТИ В ФИЛОСОФИИ АЛЬБЕРА КАМЮ

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

Жуланова А. А., Бондарева О. Ю.

Миськевич В. И. – канд. филос. наук, доцент

Рассматривается поиск А. Камю нравственных оснований гуманистической морали.

В наши дни, когда религия не играет первостепенной роли в жизни общества, актуальным является построение гуманистической морали на светских основаниях и недопущение уничтожения нравственных ценностей. Нахождение основания для нравственного поведения являлось одной из целей творчества французского философа, моралиста и писателя 20 века Альбера Камю.

Следует отметить, что Камю не считал себя атеистом, хотя его воззрения часто характеризуют как иррелигиозные и атеистические. В начале своего философского пути Камю тяготеет к экзистенциализму, испытывая влияние Ж.-П. Сартра. Исходной точкой является понятие абсурда, т.е. противоречивости мира и бытия. Камю считал, что для человека целью является выход из этого абсурда для достойного существования в реалиях своего времени. Бунт, мотивированный солидарностью с угнетёнными, являясь протестом против несправедливости, приближает к достижению данной цели. Таким образом возникают гуманистические ценности: справедливость, свобода, красота и другие.

Камю рассматривает возможности ограничения насилия в революционном движении, что актуально для нашего времени. По мнению философа, одним из проявлений насилия в обществе является смертная казнь, которую он называет узаконенным убийством.

Для более полного понимания моральных ценностей философии Альбера Камю необходимо комплексно рассмотреть его творчество.

По-разному бунтуют против абсурда герои художественных произведений Камю: Калигула, Мерсо и Сизиф. Сизиф внутренне убеждён в ответственности за свою судьбу, Калигула убивает и попирает традиционные ценности, Мерсо же до самой смерти не может осознать уникальность и одновременно общность людей. На примере этих трёх героев философ демонстрирует различные варианты бунта и как бы предлагает каждому выбрать свой «вариант игры». Нужно понимать, что эти пути лишь дают толчок для дальнейшего развития философской мысли, но никак не являются конечным решением.

Только к концу своей жизни Камю смог сформировать целостную гуманистическую мораль из выработанных им нравственных ценностей. В произведении «Бунтующий человек» теоретически разрабатывается базовая ценность его морали – солидарность. Истоком идей, изложенных в «Бунтующем человеке» является роман «Чума», где в результате борьбы со злом люди приобретают ценности, за которые стоит бороться. Мера является ценностью, которая объединяет и опосредует все остальные. Роль меры может играть красота.

Камю выступает против тоталитарных идеологий, отнимающих у человека свободу и ответственность. Он также отвергает идею Бога. Для него Христос мог бы быть бунтарём, принявшим на себя страдание ради ближних, но воскресение делает миссию Христа невыполненной, так как отделяет его от смертных людей.

Тем не менее, Камю уважал ценности верующих и гуманистическую позицию христиан. Чрезмерное развитие каждого из видов гуманистической морали (религиозной и светской) может привести к воинствующему атеизму или клерикализму как идеологии. Таким образом, гуманистическая мораль Камю способна выполнять регулятивные функции в обществе, так как она предполагает диалог и сотрудничество верующих и не верующих.

Список использованных источников:

1. Камю, А. Бунтующий человек // Книга адресована специалистам-гуманитариям и широким читательским кругам. – М. : Политиздат, 1990. – 416 с.
2. Камю, А. Калигула; Падение // Художественная литература. - М.: АСТ, 2003. – 240 с.
3. Камю, А. Посторонний // Художественная литература. – М. : АСТ, 2010. – 320 с.
4. Камю, А. Чума. Миф о Сизифе. Недоразумение // Художественная литература. – Новосибирск: Книжное издательство, 2007. – 352 с.

ЛОГИКА ХОАРА

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

Супринович И. Ю.

Миськевич В. И. – канд. филос. наук, доцент

В последние десятилетия компьютеры стали занимать больше места в каждой сфере нашей жизни. В связи с этим возникла необходимость тщательной проверки компьютерных программ на правильность, ведь всё чаще и чаще компьютерам доверяют человеческие жизни. Это привело к возникновению различных методов, используемых для доказательства корректности компьютерных программ, одним из которых является логика Хоара.

Логика Хоара – формальная система с набором логических правил, используемая для строгого доказательства корректности компьютерных программ. Она основывается на идее того, что пользователь обязуется соблюдать подробное описание программы. Описание состоит из предусловия и постусловия. Предусловие – это предикат, описывающий условие, которое требуется программе для правильного выполнения; пользователь обязан следовать этим условиям, если ему надо получить правильный результат. Постусловие – это предикат, описывающий результат правильного выполнения программы; пользователь может рассчитывать на правильное выполнение программы, если выполнит предусловие.

Программа может быть частично корректной по отношению к её описанию. Если предположить, что предусловие верно как раз перед выполнением программы, и если затем программа завершается, то постусловие верно. Программа также может быть абсолютно корректной. Абсолютная корректность подобна частичной за исключением одного факта: абсолютная корректность гарантирует завершение программы, в отличие от частичной корректности.

Следует отметить, что если пользователь использует программу без выполнения её предусловия, то она может повести себя как угодно и при этом остаться корректной. Поэтому, если требуется, чтобы программа была устойчива к ошибкам, предусловие должно включать в себя возможность ошибочного ввода, а постусловие – описание того, что может произойти в случае неправильного ввода.

Логика Хоара использует Тройки Хоара для рассуждения о корректности программы. Тройка Хоара имеет вид $\{Pre\} S \{Post\}$, где Pre – предусловие, $Post$ – постусловие, а S – одно или несколько выражений, которые являются реализацией программы. Тройка $\{Pre\} S \{Post\}$ в случае абсолютной корректности означает, что если Pre является истинным и выполнит S , то после этого S завершится в таком состоянии, где $Post$ истинно.

Рассмотрим в качестве примера тройку $\{x=-7\} x:=x^2 \{x<0\}$. Очевидно, что эта тройка абсолютно корректная, ведь если подставить -7 в выражение $x:=x^2$, действительно окажется, что $x<0$. Несмотря на то, что эта тройка корректная, она не является самой точной. Она будет более точной, если мы напишем в постусловии $x<0 \ \&\& \ x>-30$. Сильнейшим постусловием будет $x=-14$. Таким образом, сильнейшее постусловие $sp(S, Pre)$ – это предикат, определяющий все те состояния, в которые попадёт S из начальных состояний, удовлетворяющих до начала выполнения программы условию Pre .

Также существует такое понятие, как слабейшее предусловие. Слабейшее предусловие $wp(S, Post)$ – это предикат, определяющий все те начальные состояния, из которых программа S после её завершения попадёт в состояния, удовлетворяющие $Post$. Попробуем определить $wp(x:=x+3, x>z+5-x^*y)$, т.е. такие значения x до начала работы программы, чтобы после прохождения через $x:=x+3$ мы получили $x>z+5-x^*y$. Для этого мы просто подставляем $x:=x+3$ в $x>z+5-x^*y$: $wp(x:=x+3, x>z+5-x^*y) = \{x>z+2-x^*y-3y\}$.

Для доказательства частичной корректности программ можно использовать как слабейшее предусловие, так и сильнейшее постусловие. Пусть задана сама программа S , предусловие Pre и постусловие $Post$. Пусть также мы смогли построить сильнейшее постусловие этой программы. Вспомним, что тройка Хоара $\{Pre\} S \{Post\}$ означает, что если Pre истинно до начала выполнения S , и S завершается, то после завершения S утверждение $Post$ станет истинным. Тогда можно сформулировать теорему о корректности программ обработки данных: программа S частично корректна тогда и только тогда, когда $sp\{S, Pre\}$ целиком лежит в $Post$.

Аналогичным образом можно сформулировать теорему о корректности программ обработки данных в случае со слабейшим предусловием: пусть задана сама программа S , предусловие Pre и постусловие $Post$ и мы смогли определить слабейшее предусловие программы. Вспомним, что тройка Хоара $\{Pre\} S \{Post\}$ означает, что если Pre истинно до начала выполнения S , и S завершается, то после завершения S утверждение $Post$ станет истинным. Тогда программа S частично корректна тогда и только тогда, когда $wp\{S, Post\}$ целиком лежит в Pre .

Таким образом, логика Хоара применяется для доказательства частичной корректности программ.

Список использованных источников:

1. Hoare Logic. – [Электронный ресурс]. – Режим доступа: <https://www.cs.cmu.edu/~aldrich/courses/654-sp09/notes/3-hoare-notes.pdf>. – Дата доступа: 19.03.2016.

ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ КАК ОДНА ИЗ ПАРАДИГМ ПРОГРАММИРОВАНИЯ

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

Чистяков М. Ю.

Миськевич В. И. – канд. филос. наук, доцент

В статье рассматривается нетрадиционный подход к программированию, основанный на логике предикатов первого порядка. В качестве примера использования данного подхода демонстрируется решение известной загадки Эйнштейна “О пяти домах” на языке логического программирования Prolog.

На сегодняшний день программирование в сознании многих неразрывно связано с понятием **алгоритма**. Алгоритм компьютерной программы – это набор инструкций, описывающих порядок действий ЭВМ для достижения некоторого результата. Такой подход к программированию, когда на программиста ложится задача составления алгоритма решения задачи и занесения его в память ЭВМ, называется **императивным**. Колоссальное распространение императивных языков программирования обусловлено, в первую очередь тем, что они естественным образом соответствуют архитектуре современных ЭВМ. В качестве своей алгоритмической модели современные ЭВМ используют так называемую **машину фон Неймана**, основанную, в свою очередь, на машине Тьюринга, которые базируются на понятии алгоритма.

Если взглянуть на развитие языков программирования, то можно заметить, что с течением времени они становились все ближе к естественному языку. Внесение набора команд в первые ЭВМ представляло собой скорее аппаратный процесс по установке переключателей и перемычек. Затем, после того как была воплощена в жизнь идея фон Неймана о представлении управляющей программы как набора данных в памяти, в качестве языка стали выступать машинные коды. Вскоре появились языки ассемблера, а после – языки высокого уровня, такие как язык **Си**. Однако, хоть языки высокого уровня значительно приблизились к естественным (яркий пример – условный оператор **if**, который буквально означает “если”), задача программирования по-прежнему сводилась к построению алгоритма.

Другой подход к решению различного рода задач на ЭВМ предлагает парадигма **декларативного программирования**. Суть данного подхода заключается в том, что алгоритм решения задачи генерируется машиной автоматически, и для решения необходимо формальное описание поставленной задачи. Подвидом декларативной парадигмы принято считать **логическое программирование**. Логическое программирование для формального описания задачи использует **логику предикатов**. Логика предикатов широко используется в математике для записи различных теорем и утверждений и удобна для формализации. Кроме того, законы логического вывода позволяют оперировать формально записанными данными для получения новой информации. Пользуясь этими законами, машина пытается самостоятельно вывести необходимую информацию из данных. Таким образом, задача программиста состоит уже не в разработке алгоритма решения задачи, отвечая на вопрос ‘**как?**’, а в правильной формальной записи входных данных, концентрируясь на том, ‘**что**’ ему необходимо получить. Логическое программирование активно применяется в исследованиях искусственного интеллекта.

Одним из первых языков логического программирования является **Prolog**. Название Prolog есть сокращение, означающее “программирование в терминах логики”. Идея использовать логику в качестве основы языка программирования возникла в середине 70-х годов. В 1980-х годах, например, Prolog активно использовался во время разработок японской национальной программы “Компьютеры пятого поколения”.

Программа на языке Prolog состоит из фактов, правил логического вывода и запросов, по которым осуществляется поиск решения задачи, пользуясь механизмом **поиска с возвратом** и **унификацией**. Получив запрос, программа пытается найти описание этого запроса в своём коде. Затем, найдя факт, она производит унификацию и возвращает ответ по запросу. Встретив правило, машина преобразует исходный запрос в набор других запросов, необходимых для удовлетворения первого и пытается решить их.

Рассмотрим применение данного языка на конкретном примере. В качестве задачи возьмём известную загадку Эйнштейна. Вот её формулировка:

- 1) На улице стоят пять домов
- 2) Англичанин живёт в красном доме
- 3) У испанца есть собака
- 4) В зелёном доме пьют кофе
- 5) Украинец пьёт чай
- 6) Зелёный дом стоит сразу справа от белого дома
- 7) Тот, кто курит Old Gold, разводит улиток
- 8) В жёлтом доме курят Kool
- 9) В центральном доме пьют молоко
- 10) Норвежец живёт в первом доме
- 11) Сосед того, кто курит Chesterfield, держит лису
- 12) В доме по соседству с тем, в котором держат лошадь, курят Kool
- 13) Тот, кто курит Lucky Strike, пьёт апельсиновый сок

- 14) Японец курит Parliament
 15) Норвежец живёт рядом с синим домом

Кто держит зебру?

Автор данной статьи попытался решить задачу самостоятельно в уме. На это ушло примерно 54 минуты©. Решение данной задачи популярным императивным подходом на одном из популярных языков программирования возможно, но очень затруднительно. В то же время, решение этой задачи с использованием декларативного подхода сводится к грамотному формальному описанию данных и составлению запроса. Ниже приведён пример решения задачи. Даже не вдаваясь в конструктивы языка видно, что программа почти целиком состоит из перечисления входных условий и занимает приблизительно столько же места, сколько её формулировка. Алгоритм поиска ответа формируется машиной самостоятельно, пользуясь механизмом поиска с возвратом и унификацией.

```
%Порядок следования элементов в списке одного дома:
```

```
%Цвет, национальность, напиток, сигары, животное
```

```
solve(X):-
```

```
  HouseList = [_,_,_,_],
  add([red,english,_,_,_], HouseList),
  add([_,spanish,_,_,dog], HouseList),
  add([green,_,coffee,_,_], HouseList),
  add([_,ukrainian,tea,_,_], HouseList),
  isRight([white,_,_,_], [green,_,_,_], HouseList),
  add([_,_,old_gold,snail], HouseList),
  add([yellow,_,_,kool,_], HouseList),
  HouseList = [_,_[_,milk,_,_,_],_],
  HouseList = [_[_,norwegian,_,_,_],_,_,_],
  near([_,_,chesterfield,_], [_,_,_,fox], HouseList),
  near([_,_,_,horse], [_,_,_,kool,_], HouseList),
  add([_,_,juice,lucky_strike,_], HouseList),
  add([_,japanese,_,parliament,_], HouseList),
  near([_,norwegian,_,_,_], [blue,_,_,_], HouseList),
  add([_,X,_,_,zebra], HouseList).
```

```
%Предикат отношения "справа" для X и Y
```

```
isRight(X, Y, [X,Y|_]).
```

```
isRight(X, Y, [_|L]):-
```

```
  isRight(X, Y, L).
```

```
%Предикат поиска соседних элементов
```

```
near(X, Y, L):-
```

```
  isRight(X, Y, L);
```

```
  isRight(Y, X, L).
```

```
%Предикат добавления в список
```

```
add(X, [X|_]).
```

```
add(X, [_|L]):-
```

```
  add(X,L).
```

Стоит также отметить, что множества алгоритмически решаемых задач для разных алгоритмических моделей на сегодняшний день совпадают. Если задача является алгоритмически решаемой, то она решается в любой алгоритмической модели. То есть если задача решается в императивной форме, то она непременно решается и в декларативной форме. Однако удобство различных подходов к решению для различных парадигм, конечно, различно.

Таким образом, логическое программирование – удобный инструмент, который позволяет успешно решать круг задач, связанных с искусственным интеллектом, поиском решений и даже доказательством теорем. Знакомство с одним из представителей данной парадигмы – Prolog-ом будет полезно для программистов любой квалификации для расширения взглядов на программирование в целом и расширения арсенала методов решения задач.

Список использованных источников:

1. Сошников, Д. В. Парадигма логического программирования / В. Д. Сошников. – М.: "Вузовская книга", 2006. – 220 с.
2. Братко, И. Программирование на языке Пролог для искусственного интеллекта / И. Братко. – М.: "Мир", 1990. – 560 с.

АНТОЛОГИЧЕСКИЙ ЭТАП АНТИЧНОЙ ФИЛОСОФИИ: СОФИСТЫ И СОКРАТ

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

Ярошевич В. В.

Миськевич В. И. - канд. филос. наук, доцент

Антропологический, или, другими словами, гуманистический период в развитии античной философии в первую очередь интересен тем, что пристальное внимание древнегреческих мыслителей переносится с макрокосма на человека, который становится основным объектом изучения и в дальнейшем предстает перед нами в качестве микрокосма. Основной задачей древнегреческих философов этого периода становится определение природы человека, его сущности как такового, не забывая при этом об этико-социальных проблемах. Именно этим и можно охарактеризовать деятельность софистов, Сократа и сократических школ второй половины V в.

Если рассматривать софистов как некое определенное философское течение, то в нем невозможно выделить точной однородности. Само философское явление, как софистика, развивалось относительно долго, около столетия. Основной ее чертой является смещение интереса и рефлексии с физиса и космоса на проблему более узкую, а именно человека и его бытия. В качестве основных вопросов софисты рассматривали этические, политические, языковые, религиозные, связанные с искусством и процессом воспитания человека темы. То есть внимание софистов в первую очередь было направлено на изучение культуры и определение места, существование человека в ней. Поэтому принято считать софистов первооткрывателями так называемого «гуманистического периода в истории философии».

Интересно, что софисты за вознаграждение учили искусству красноречия и навыкам правильно вести спор. Тогда в городах Афинского союза обществу как раз требовались люди, способные хорошо владеть своим языком. Поэтому софисты и пришлись как раз кстати. Им не важно было какова истина, они занимались своим делом. Это хорошо объясняется тем, что они в один момент могли доказать тезис, а уже через некоторое время антитезис, чему софисты и обучали своих последователей, что стало одной из основных причин в бесповоротном крахе догматизма традиции в миропонимании греков периода античности.

Догматизм древних греков стоял на авторитете; софисты же настаивали на подтверждении и доказательстве, что не могло не разбудить человеческую идею от догматического сна. В процессе своей работы они смогли спровоцировать образование науки о слове и положить начало логике путем нарушения до этого еще не существовавших и не определенных к этому времени законов правильного логического мышления, способствуя этим их зарождению.

Софисты являлись агностиками, по их мнению мир вокруг невозможно познать. Агностицизм софистов был следствием их релятивизма – учения об относительности всех знаний, всех этических, эстетических и религиозных ценностей; в гносеологии под этим подразумевается относительность истины, ее зависимость от таких факторов, как время, местоположение, условия, взгляды человека и т.д. Софисты считали, у каждого человека истина своя, они признавали только субъективные истины, коих бесчисленное множество, а истину объективную истину отвергали. Поэтому и говорят о том, что агностицизм софистов находился в рамках их гносеологического релятивизма. К тому же, релятивизма софисты придерживались и в нравственном аспекте: для них не было определенной меры добра и зла, они считали, что как кому выгодно, так и хорошо, так и надо. Поэтому с точки зрения этики агностицизм софистов становился едва ли не аморализмом.

Хотелось бы остановиться на наиболее известном представителе софистики и ее основателе как таковом – Протагоре. Известен он, в первую очередь, за следующее высказывание: «Человек есть мера всех вещей: существующих, что они существуют, и несуществующих, что они не существуют». Это изречение и стало своеобразным девизом софистики. На его взгляд, то, как человек осознает мир вокруг него, зависит от особенностей его чувственного восприятия. Человек для него является лишь механизмом приема и передачи информации о бытии, которую он принимает и отправляет исходя из своих способностей и, собственно, чувственного восприятия. Также Протагор сумел сформулировать законы демократического образа управления и обосновать равноправие свободных людей.

Не менее интересными представителями софистики, помимо Протагора, были также Горгий и Продик. В своем самом известном труде «О природе» Горгий доказывает три следующих тезиса: ничего не существует; если бы что-нибудь было, его нельзя было бы познать; если бы оно существовало и было познаваемо, его нельзя было бы высказать. Следствием этого было то, что, с его точки зрения, ни о чем невозможно сказать наверняка. Продика же больше интересовал язык со стороны уподобления одинаковых по значению слов и правильному использованию слов в риторике. Это обосновывается его интересом к составлению этимологических ветвей близких по смыслу слов. Не прошла мимо его внимания и полемика, где он исследовал проблемы различных методов опровержения.

Как ни странно, но к закату софистики сформировалась достаточно негативная оценка деятельности ее представителей, философия софистов и их методы подвергались многочисленной критике и осуждению. Среди многочисленных критиков софистики, самым известным был Сократ. Он, как и софисты, изучал проблематику человека, при этом для него человек был созданием высокоморальным. Поэтому, в отличие от софистов, его философские взгляды изображаются как этический антропологизм. Смысл своих философских

поисков он сформулировал следующим высказыванием: "Я никак еще не могу, согласно дельфийской надписи, познать самого себя". Сократ понимал ничтожность своей мудрости по сравнению с мудростью высших сил, и только поэтому, на его взгляд, был мудрее других. Аристотель говорил, что "Сократ занимался вопросами нравственности, природу же в целом не исследовал", - у Сократа были свои философские взгляды, в которых не было места вопросам космоцентрического характера, а также релятивизму, характерному софистам.

По мнению Сократа, каждая личность может обладать субъективными взглядами и мнениями, однако истина, независимо от этого, всегда будет единственной. С постижением этой самой истины и связано создание Сократом метода, называемого «майевтикой», который представляет собой субъективную диалектику – искусство вести беседу, в результате которой достигается истинное знание путем движения основной мысли через противоречия в идеях спорящих и преодоления предвзятости в их точках зрения. Сократ давал возможность истине «родиться в душе собеседника», говоря при этом, что не имеет за собой точных истинных знаний. Основной задачей майевтики является определение понятия предмета мысли через всестороннее его рассмотрение. Тем самым Сократ стал первым мыслителем, у которого получилось вывести знание на уровень понятия, до этого философы делали это бессознательно, не преследуя таких целей, к тому же метод Сократа ставил целью получение понятийного знания, что говорит нам о Сократе, как о человеке, придерживавшегося рационалистических взглядов. Сократ соглашался, что мир в целом для человека не может быть познаваем, однако понять возможно лишь только самого человека, его душевный мир, в чем Сократ и видел основную задачу философии своего времени. Узнать, постичь самого себя - это обнаружить суть нравственных принципов, общих у людей;

Характерной чертой философии Сократа является то, что он положил начало телеологическому мировосприятию. Он был убежден, что все существующее в макрокосме происходит во благо человека. Для Сократа осознание мира является делом ненужным и безбожным, ведь целью своей философии он видел в фундировании религиозно-нравственного миропонимания. Телеология Сократа является достаточно тривиальной. Здесь мы наблюдаем определенную взаимосвязь явлений макрокосма и нужд и функции человеческого организма: глаза должны видеть, боги же, в свою очередь, дают свет; ночь подарена людям для отдыха, равно как и свет небесных светил для того, чтобы узнавать примерное время и т.д.

Центром философии Сократа является исследование и создание идеалистической морали. Целью Сократа было понять суть морального совершенства человека. Она считал, что делать добро возможно лишь понимая что такое добро. Поэтому честный и добронравный человек обязательно должен отдавать себе отчет что есть благо. Понимая значение добра, человек будет избегать плохих, безнравственных действий, а совершать только положительные относительно морали поступки. Благородство, честность, доброта является итогом знания, аморальность и распутство же следствием незнания понятия добра. Из этого следует, что мораль и знание в данном случае сходятся в одной точке, ведь чтобы быть высоконравственным человеком, обязательно нужно прекрасно понимать что есть добродетель.

Тремя основными добродетелями Сократ считал:

1. Умеренность (знание, как сдерживать страсти)
2. Храбрость (знание, как справиться с угрозами)
3. Справедливость (знание, как следовать законам божественным и людским)

При этом Сократ не приравнивает знание и добродетель, поэтому источником аморальных и дурных человеческих поступков является как раз незнание. Истокование сущности добродетели в этом случае превращается в своеобразный источник нравственного прогресса и совершенствования человеческой души. Диалектика же становится методом развития своего внутреннего «Я», благодаря ей человек может приблизиться к пониманию значения своего пребывания в этом мире.

Агностицизм и релятивизм, тезис об условности истины несомненно оказали большое влияние на формирование философской мысли древних греков. Правда нельзя точно сказать, что именно цели, которые преследовали софисты, представляют основную ценность. Своими взглядами и тезисами об отрицании истинности любого знания у софистов получилось создать определенный интерес мыслителей к знанию, что способствовало появлению логики и диалектики знания. Если говорить про Сократа, то свое собственное существование и бытие для него было философской проблемой, равно как и вечный вопрос о значении жизни и смерти. Главным достижением Сократа было то, что он сумел произвести своеобразный переворот в философской мысли античности и развернуть основные задачи и интересы философии на познания человеком нравственности и себя самого в целом. Задачей человека Сократ видит культурное и умственное самосовершенствование, что еще раз доказывает его высокий, благородный, человеческий взгляд на окружающий мир.

Список использованных источников:

1. Философия классической эпохи. Софисты и Сократ: [Электронный ресурс]. URL: <http://studopedia.ru>. (Дата обращения: 15.03.2016).
2. Философия античности: [Электронный ресурс]. URL: <http://filosofia.ru>. (Дата обращения: 12.03.2016).
3. Проблема человека в философии софистов и Сократа: [Электронный ресурс]. URL: <http://ref.by>. (Дата обращения: 13.03.2016).

**СЕКЦИЯ «ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ
СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»**

СИСТЕМА АНАЛИЗА ИНТЕРНЕТ-ТРАФФИКА И ЭФФЕКТИВНОСТИ КАНАЛОВ ЕГО ПРИВЛЕЧЕНИЯ

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

Яцевич П. Б.

Бибило П. Н. – д-р. техн. наук, профессор

Автоматизация сбора информации об активности пользователей веб-сайтов, а также нахождение зависимостей действий пользователей от источников их попадания на сайт требует разработки определенных программных средств. Существующие системы веб-аналитики не позволяют устанавливать связь между действиями пользователя на сайте и вне его, такими как успешное совершение покупки в интернет-магазине, а установка и настройка данных систем является трудоемким процессом.

В настоящее время все шире развивается сеть интернет-магазинов и других организаций, предоставляющих услуги в сети Интернет. Эффективность работы таких организаций в большой степени зависит от привлечения пользователей на их веб-сайты. Многие из них вкладывают денежные средства в рекламу на других сайтах и в поисковых системах. Анализ поведения пользователей сайтов позволит сделать определенные выводы об эффективности вложения средств в отдельно взятые источники их привлечения.

Существует множество систем анализа посещаемости и поведения пользователей сайтов. Большинство из них (например, Рейтинг@mail.ru, HotLog, LiveInternet) позволяет получить информацию о количестве просмотров страниц, количестве и длительности сессий, браузерах и операционных системах, а также страницах, с которых пользователи переходят на сайт. Данные системы являются довольно простыми и не предоставляют возможности анализа продаж, а также расширения собираемой информации пользовательскими параметрами.

Наиболее популярными и многофункциональными системами являются Google Analytics (и его обновленная версия Universal Analytics) и Яндекс.Метрика. Обе системы имеют весьма широкие возможности анализа трафика и интернет-торговли (модули «Ecommerce» и «Электронная коммерция» соответственно) и могут подавать информацию в виде таблиц, разнообразных графиков и диаграмм, применять фильтры по огромному числу критериев и характеристик. Они позволяют объединять данные о действиях пользователя с различных устройств, используя уникальный идентификатор пользователя, однако реализация алгоритма генерации идентификаторов и определения пользователя выполняется разработчиками каждого отдельно взятого сайта. Также эти системы предоставляют API для импорта данных из внешних источников, но реализация клиентской части алгоритма тоже остается за разработчиками сайта.

Ввиду сложности настройки данных систем и необходимости постоянного привлечения квалифицированных разработчиков, подавляющее большинство веб-сайтов используют лишь их базовые возможности. Поэтому разработка системы веб-аналитики, являющейся простой в установке и настройке, и позволяющей отслеживать множество параметров, а также осуществлять связывание действий пользователя на сайте и вне его, является актуальной задачей.

Разработка многофункциональной и универсальной системы анализа действий пользователей сайтов возможна для какой-либо из стандартных платформ. Одними из наиболее популярных систем управления сайтом в СНГ, особенно для интернет-магазинов, являются «1С-Битрикс: Управление сайтом» и «1С-Битрикс: Корпоративный портал». При разработке сайтов на основе данных систем используется платформа Bitrix Framework и язык PHP.

В качестве основы для построения отчетов и сбора базовой информации был выбран сервис Universal Analytics, т.к. он является хорошо расширяемым и позволяет создавать собственные параметры и отчеты.

Разрабатываемая система позволяет установить и настроить отслеживание Universal Analytics на любом сайте, работающем на платформе Bitrix Framework с помощью графического интерфейса, без необходимости иметь специальные навыки программирования.

С помощью разрабатываемой системы можно собирать информацию о дате первого и последнего заходов на сайт, источниках перехода на сайт, посещенных страницах и многое другое для каждого посетителя сайта. Также был разработан оптимальный алгоритм определения пользователя при использовании им различных устройств и браузеров, и определения различных пользователей на одном устройстве / браузере, основанный на использовании файлов cookie и механизме авторизации на сайте, обеспечивающий минимально возможную долю ошибок в отчетах Universal Analytics.

При создании заказов на сайте интернет-магазина, система поддерживает автоматический экспорт информации о заказе, включая список товаров с ценами, основными параметрами, себестоимостью и прибылью от продажи.

Также в рамках создания данной системы разработан модуль для CRM (системы управления взаимоотношениями с клиентами) «Битрикс-24», которая является одной из самых распространенных в СНГ. В случае, если пользователь сайта связывается с продавцом напрямую, (как правило, по телефону), продавец может оформить заказ на сайте от имени пользователя, либо отметить цель звонка (уточнение информации и т.п.). Отмеченные продавцом данные автоматически экспортируются в Universal Analytics, что позволяет получить полный отчет о действиях пользователя.

Имея информацию о источниках привлечения каждого пользователя на сайт, а также о сделанных им

звонках, оформленных им заказах, прибыли, которую он принес и затратах магазина на рекламу в различных источниках, можно сделать выводы о эффективности вложения денежных средств по каждому источнику.

Разрабатываемая система не имеет аналогов, автоматически интегрирующихся с сайтом под управлением 1С-Битрикс и позволяющих осуществлять и автоматизировать связывание действий пользователей на сайте и вне его.

Список использованных источников:

1. Web Analytics Tools, Event Tracking & More | Google Analytics Features [Электронный ресурс] / – Режим доступа: <http://www.google.com/analytics/standard/features/> – Дата доступа: 21.03.2016.
2. Google Analytics | Google Developers [Электронный ресурс] / – Режим доступа: <https://developers.google.com/analytics/> – Дата доступа: 21.03.2016.
3. 1С-Битрикс Разработчикам - Документация по CMS "1С-Битрикс: Управление сайтом" [Электронный ресурс] / – Режим доступа: <https://dev.1c-bitrix.ru/docs/> – Дата доступа: 23.02.2016.

РАЗРАБОТКА USBPOS-КЛАВИАТУРЫ НА БАЗЕ 32-ХРАЗРЯДНОГО МИКРОКОНТРОЛЛЕРА ARMCORTEX-M0

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

Григорьев В. Ю.

Ключеня В. В. – ассистент кафедры ЭВС

В настоящее время вопрос автоматизации рабочего процесса стоит как никогда остро. Проблемно-ориентированное программное обеспечение для персональных компьютеров имеет в своём составе не один десяток специализированных функций, быстрый доступ к которым осуществляется посредством «горячих клавиш». Увеличение их количества приводит к усложнению назначаемых на периферийные устройства комбинаций, создавая дополнительные трудности в работе для пользователя. Разрабатываемая клавиатура призвана разрешить эту проблему, обеспечивая гибкость и персонализацию работы.

POS-клавиатура (PointofSale – англ.) представляет собой периферийное устройство ввода информации. Ввиду того, что все клавиши программируемы, данное устройство относится к программно-аппаратным комплексам. Его использование позволяет существенно повысить степень автоматизации рабочего процесса при использовании в связке с USB-HOST-устройством, таким как персональный компьютер, серверная стойка, промышленный станок и т.д.

Так, англоязычное название «точка продажи» исторически сложилось в результате того, что впервые подобные комплексы стали использоваться именно в качестве устройства ввода к электронным кассовым аппаратам на торговых местах. Помимо клавиш в корпусе размещались также замочные скважины для блокировки касс, считыватели магнитных карт. Добавление подобных узкоспециализированных интерфейсов легло в основу философии класса POS-клавиатур.

На рисунке 1 представлена структурная схема разрабатываемого программно-аппаратного комплекса:

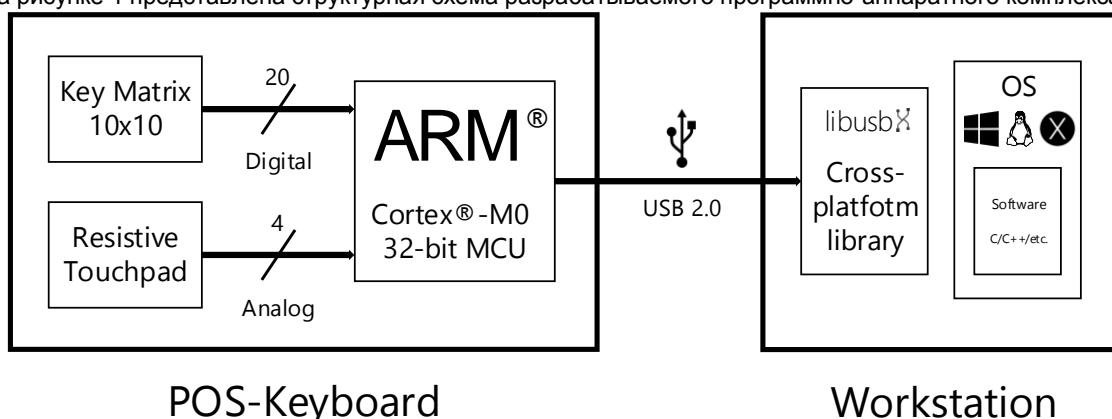


Рис. 1. Структурная схема программно-аппаратного комплекса

Аппаратная часть включает в себя цифровую матрицу кнопок размерностью до 10x10, а также аналоговую резистивную панель. В то время, как последняя занимает от 4 до 6 аналоговых портов микроконтроллера в зависимости от типа панели, матрица кнопок требует столько цифровых выводов, сколько будет результат сложения её размерностей по горизонтали и вертикали: 20 портов ввода-вывода общего назначения (GPIO–англ.).

Наличие резистивной сенсорной панели позволяет сделать POS-клавиатуру более гибкой в использовании. Помимо программирования одиночных нажатий клавиш, либо же их комбинаций, существует возможность назначения программных действий на те или иные жесты, привычные пользователям современных смартфонов.

Программная часть состоит из двух жизненно важных компонентов. Так, для связи с HOST-устройством используется кроссплатформенная библиотека libusb, обеспечивающая связь с клавиатурой по одноимённому протоколу. На её основе создаётся драйвер, включаемый в специализированное программное обеспечение, с помощью которого пользователь назначает сценарии на клавиши и тактильные жесты, а также сохраняет их в память клавиатуры.

Программное обеспечение для SLAVE включает в себя инициализацию интерфейса USB, опрос GPIO на предмет воздействия пользователем, описание алгоритма общения клавиатуры с HOST-устройством. Перезапись внутренней памяти микроконтроллера осуществляется также посредством USB.

Разработанное устройство имеет возможность хранить около сотни различных команд в качестве реакции на воздействие со стороны пользователя. Так, 32-х разрядный микроконтроллер NXP LPC1114FBD48 имеет 32 кБ энергонезависимой флеш-памяти и 6 кБ оперативной, обладает максимальной тактовой частотой 50 МГц, что позволяет проводить высокочастотный опрос матрицы клавиш, точно распознавать траекторию нажатий на резистивную панель.

Спецификацией USB ограничена множественность одновременного нажатия клавиш для периферийных устройств (не более 6 клавиш). В связи с этим необходимо внести программное ограничение на программирование комбинаций «горячих клавиш».

Использование библиотеки libusb, относящейся к категории свободного программного обеспечения (OpenSource–англ.), положительно влияет на программную часть ввиду её открытости. Это даёт возможность пользователю самостоятельно внести изменения в стандартную логику работы устройства.

Разработанная POS-клавиатура несёт в себе потенциал модернизации. Так, допустима замена проводной связи с HOST-устройством беспроводной посредством подключаемого к последнему радиомодуля-приёмника гражданского диапазона частот 2.4 ГГц. В свою очередь, к микроконтроллеру посредством интерфейса SPI добавляется антенна передатчика со специализированной интегральной микросхемой.

Список использованных источников:

1. Агуров П. В. Интерфейсы USB. Практика использования и программирования / П. В. Агуров. – СПб. : БХВ-Петербург, 2004. – 576 с.
2. libusb [Электронный ресурс]. – Режим доступа : <http://libusb.info/>.
3. LPC1114FBD48: 32kB flash, 6kB SRAM, LQFP48 package [Электронный ресурс]. – Режим доступа : <http://www.nxp.com/products/microcontrollers-and-processors/arm-processors/lpc-cortex-m-mcus/lpc-cortex-m0-plus-m0-mcus/lpc1100-cortex-m0-plus-m0-mcus/32kb-flash-6kb-sram-lqfp48-package:LPC1114FBD48>.

АЛГОРИТМ УЧЁТА КОЛЛИЗИЙ ПЛАНАРНЫХ ПОЗИЦИОНЕРОВ НА ОДНОМ СТАТОРЕ

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

Форутан М. М., Кузнецов В. В.

Карпович С. Е. – д-р. техн. наук, профессор

Рассмотрена математическая модель и алгоритмизация анализа коллизий при одновременном перемещении нескольких планарных позиционеров на одном статоре. Получены условия бесколлизионных перемещений в виде систем неравенств. На основании предложенного алгоритма разработана программа в среде MATLAB с удобным пользовательским интерфейсом.

Проблема учёта коллизий при формировании программируемых движений системами перемещений со многими степенями свободы, до шести включительно, всегда возникает при разработке многокоординатных систем перемещений с одновременным задействованием нескольких автономных координатных модулей или роботов в одном рабочем пространстве. Это в полной мере относится и к предложенным нами системам перемещений на механизмах параллельной кинематики, которые приводятся в движение планарным приводом, построенным на композиции нескольких автономно-управляемых двухкоординатных линейных шаговых двигателей на одном статоре [1]. В работе авторов [2, 3] была предложена алгоритмизация идеализированной математической модели одной из таких систем без учёта геометрических размеров планарных позиционеров. Учёт геометрии позиционеров приводит к необходимости алгоритмизации задач кинематики с ограничениями по траекторным реализациям программируемых движений.

В настоящей работе при создании алгоритма учёта коллизий планарных позиционеров на одном статоре, учитывалось, что каждый позиционер представляет собой физический объект, рёбра которых образуют параллелепипеды, у которых плоскости образуют параллельные и ортогональные между собой

грани в плоскости x_0Oy_0 (рис. 1). Перемещения каждого из позиционеров, осуществляются по двум взаимно-ортогональным направлениям.

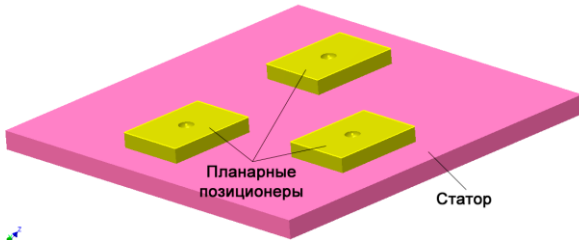


Рис. 1. Пространственная система перемещений

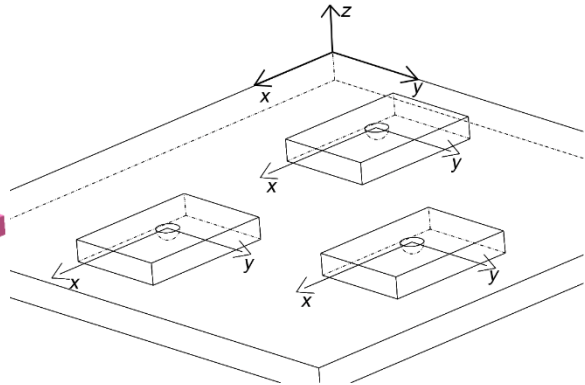


Рис. 2. Структурно-кинематическая расчётная схема

Пусть планарные позиционеры геометрически представляет собой прямоугольный параллелепипед с размерами основания b и c . Коллизии позиционеров понимаются нами как возможная геометрическая интерференция прямоугольных параллелепипедов располагаемых на общей плоскости статора. Расчётная модель для трёх планарных позиционеров показана на рис.2. С учётом этого условия их пресечения (геометрическая интерференция) будет характеризоваться системами неравенств:

$$\begin{cases} |x_F - x_D| < b \\ |y_F - y_D| < c, \end{cases} \quad \begin{cases} |x_F - x_E| < b \\ |y_F - y_E| < c, \end{cases} \quad \begin{cases} |x_D - x_E| < b \\ |y_D - y_E| < c. \end{cases} \quad (1)$$

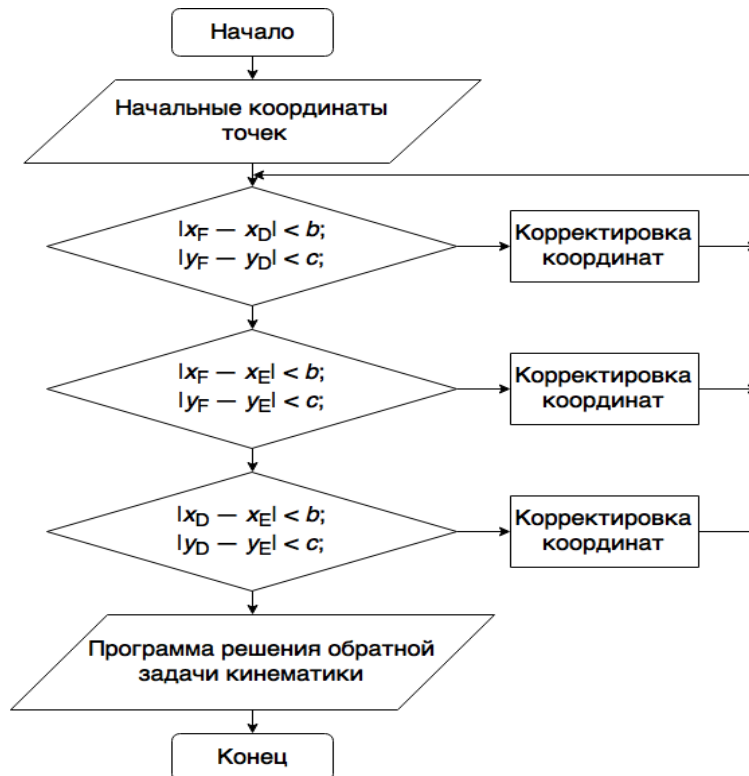
Данные системы неравенств характеризуют условия наличия коллизий. принято называть условиями коллизий.

Таким образом на при пересечении планарных позиционеров, будут общие области множества точек, линейные размеры которых определяются следующими выражениями:

$$\begin{aligned} \Delta x_{FD} &= x_F - x_D, & \Delta x_{FE} &= x_F - x_E, & \Delta x_{DE} &= x_D - x_E \\ \Delta y_{FD} &= y_F - y_D, & \Delta y_{FE} &= y_F - y_E, & \Delta y_{DE} &= y_D - y_E. \end{aligned} \quad (2)$$

Следовательно, для того чтобы избежать пересечения рассматриваемых множеств, необходимо выполнить следующие граничные условия:

$$\begin{aligned} \Delta x_{FD} &= 0, & \Delta x_{FE} &= 0, & \Delta x_{DE} &= 0, \\ \Delta y_{FD} &= 0, & \Delta y_{FE} &= 0, & \Delta y_{DE} &= 0. \end{aligned} \quad (3)$$



Исходя из (1) и (3), условия аналитические условия отсутствия коллизий будут иметь вид:

$$\begin{cases} |x_F - x_D| \geq b \\ |y_F - y_D| \geq c, \end{cases} \quad \begin{cases} |x_F - x_E| \geq b \\ |y_F - y_E| \geq c, \end{cases} \quad \begin{cases} |x_D - x_E| \geq b \\ |y_D - y_E| \geq c. \end{cases} \quad (4)$$

Учитывая, что планарные позиционеры это реальные физические объекты с конкретными геометрическими размерами и погрешностями перемещений, то условия отсутствия коллизий могут быть представлены в виде:

$$\begin{cases} |x_F - x_D| \geq b + \varepsilon \\ |y_F - y_D| \geq c + \varepsilon, \end{cases} \quad \begin{cases} |x_F - x_E| \geq b + \varepsilon \\ |y_F - y_E| \geq c + \varepsilon, \end{cases} \quad \begin{cases} |x_D - x_E| \geq b + \varepsilon \\ |y_D - y_E| \geq c + \varepsilon, \end{cases} \quad (5)$$

где ε – максимальное значение погрешности при перемещении позиционеров.

Блок-схема программы учёта коллизий приведена на рис. 3.

Рис. 3 – Блок-схема программы учёта коллизий

Таким образом на основании полученного алгоритма была разработана программа в среде MATLAB, позволяющая учитывать коллизии трёх планарных позиционеров на одном статоре для ранее исследованной системы перемещений с шестью степенями свободы [4]. Проведенное моделирование с использованием разработанной программы позволило уточнить границы рабочей области для исследованной системы перемещений.

Список использованных источников:

1. Системы многокоординатных перемещений и исполнительные механизмы для прецизионного технологического оборудования / В.В. Жарский [и др.]; под ред. д-ра техн. наук, проф. С.Е. Карповича. – Минск : Бестпринт, 2013. – 208 с.
2. Карпович, С.Е. Алгоритм генерации опорных точек на пространственной траектории для линейной и сплайновой интерполяции / С.Е. Карпович, В.В. Кузнецов, А.Ю. Войтов // Информационные технологии и системы 2015 : материалы Междунар. науч. конф. – Минск, 2015. – С. 54–55.
3. Карпович, С.Е. Формирование аналитических функций обобщенных координат пространственной системы перемещений с шестью степенями свободы / С.Е. Карпович, В.В. Кузнецов, В.В. Поляковский // Материалы Юбилейной науч.-практ. конф., посвященной 85-летию Гомельского гос. ун-та им. Ф. Скорины. – Гомель, 2015. – Ч. 4. – С. 118–121.
4. Кинематика системы перемещений с шестью степенями свободы / А.Ю. Войтов, В.В. Кузнецов / Научно-практический журнал «Аспирант». 2016. №1. – Ростов-на-Дону, 2016. – С. 74–77.

ОБНАРУЖИТЕЛЬНАЯ СПОСОБНОСТЬ ДЛЯ АВТОМАТИЧЕСКОГО КОНТРОЛЯ ТОПОЛОГИИ

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

Титко Д. С.

Карпович С. Е. – д-р. техн. наук, профессор

Рассматривается методика анализа обнаружительной способности оборудования для автоматического контроля топологии, основанная на вероятностной оценке линейных размеров дефектов СБИС и других изделий электронной техники.

При испытаниях оборудования для автоматического контроля топологии задача экспериментального определения обнаружительной способности является одной из основных. Для этого обычно производится некоторое количество циклов сканирования специально изготовленного и аттестованного тестового шаблона, в результате чего подтверждается вероятность обнаружения дефектов разных типов и размеров. Количество циклов сканирования при этом определяется, как правило, эвристическим путём [1]. Наиболее сложным в этом случае является определение вероятности обнаружения дефектов с линейными размерами, соответствующими границе чувствительности установки. Предлагаемая методика определения вероятности обнаружения таких дефектов, основана на точном расчёте необходимого количества циклов сканирования для подтверждения чувствительности с заданной вероятностью. Следует отметить, что при этом необходимо уточнение определения дефекта заданного размера.

Необходимость такого уточнения обусловлена тем, что обнаружительная способность установки автоматического контроля топологии зависит не только от линейных размеров дефекта, но и от его пространственного распределения, вследствие чего определение размера дефекта и, соответственно, размера минимального обнаруживаемого дефекта только через линейные размеры является неоднозначным. Этот вопрос решается путём стандартизации формы дефектов тестового шаблона и определения обнаружительной способности установки для дефектов фиксированной формы.

Испытания на обнаружительную способность проводятся по следующей методике:

1. Подтверждение обнаружительной способности для минимального дефекта, обнаруживаемого со

100% вероятностью.

2. Подтверждение обнаружительной способности для минимального дефекта, размером “ d ”, обнаруживаемого с вероятностью 95%. При этом d соответствует порогу обнаружения.

Последовательное определение вероятностей обнаружения дефектов с размерами, меньшими порога обнаружения. При этом последовательно рассматриваются дефекты, начиная с дефектов размером $d-\Delta d$, с шагом Δd , заканчивая дефектами, имеющими контраст 10%. Шаг Δd может быть выбран, например, равным 50 нм. Согласно методике вероятность обнаружения дефектов связана с частотой обнаружения дефектов следующим образом:

$$P\left(\left|\frac{m}{n} - p\right| < \varepsilon\right) \geq 1 - pq/n\varepsilon^2, \quad 1)$$

где P – вероятность нахождения обнаружения в интервале, m – число благоприятных исходов, n – общее число циклов сканирования, p – вероятность обнаружения дефектов, q – вероятность необнаружения дефектов, ε – размер доверительного интервала, m/n – частота события: «Обнаружение дефекта».

Отсюда получаем соотношения между вероятностью обнаружения ε и необходимым количеством n циклов сканирования фотошаблонов, которые приведены в таблице.

Вероятность обнаружения	0,005	0,01	0,05	0,1	0,1
Количество циклов n	38000	9500	380	95	24

Нужно отметить, что если воспользоваться теоремой Муавра-Лапласа, то можно получить существенно меньшие приближенные значения для числа циклов испытаний [2].

Точное определение вероятности обнаружения дефектов с размерами, меньшими порога обнаружения установки, позволяет оптимизировать процесс фильтрации ложных дефектов типа прокол и островок при автоматическом контроле топологии, а также повысить воспроизводимость контроля.

Предложенная методика позволяет точно определить количество испытаний, необходимых для подтверждения вероятности обнаружения дефектов при автоматическом контроле топологии и используется при разработке программ и методик испытаний всего спектра отечественного оборудования для автоматического контроля топологии СБИС и других изделий электронной техники.

Список использованных источников:

1. Аваков, С.М. Автоматический контроль топологии планарных структур / С.М. Аваков. – Минск : ФУАинформ, 2007. – 168 с.
2. Alfred, K.W. Resolution Enhancement Techniques in Optical Lithography. SPIE PRESS, USA, 2001. – pp. 1–213.

СПЕЦИАЛИЗИРОВАННЫЙ ПРОЦЕССОР ШИФРОВАНИЯ СТАНДАРТА СТБ 34.11.31-2007

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

Мельников А. М.

Герасимович В. Ю. – ассистент

Рассмотрен один из возможных вариантов реализации специализированного процессора шифрования стандарта СТБ 34.11.31 на базе программируемых логических интегральных схем (ПЛИС) типа FPGA. Приводятся сравнительные характеристики различных реализаций стандарта.

Стандарт СТБ 34.11.31-2007[1] – определяет семейство криптографических алгоритмов шифрования и контроля целостности. Все алгоритмы данного стандарта делятся на 8 групп: шифрования в режимах простой замены, сцепления блоков, гаммирования с обратной связью и счетчика, алгоритмы выработки имитовставки, одновременного шифрования и имитозащиты данных, одновременного шифрования и имитозащиты ключа, а также хеширования. В данной работе будет рассмотрена аппаратная реализация шифрования в режиме гаммирования с обратной связью. Алгоритм рассчитан на шифрование блоков данных длиной 128 бит на 256 битый ключ. Шифрование осуществляется 8-ю раундами преобразований, применяемых к входным данным. При этом для одного блока применяются следующие базовые операции: сложение(80 32-разрядных операций), вычитание(16 32-разрядных операций), сложение по модулю 2(40 32-разрядных операций), циклический сдвиг вправо(56 32-разрядных операций с фиксированным сдвигом на 5,13 или 21 разряд), подстановка(224 8-разрядных операций).

Для реализации данного алгоритма была выбрана последовательная схема с параллелизмом вычислений на уровне такта шифрования, вычисления на t такте приведены на рисунке 2. Выбор последовательной схемы связан с присутствием в режиме гаммирования, обратной связи, не позволяющей реализовать полный конвейер. Параллелизм реализован при помощи конвейерной архитектуры блока

вычислений, который включает в себя повторяющиеся шаги одного такта алгоритма, функциональная схема блока приведена на рисунке 1.

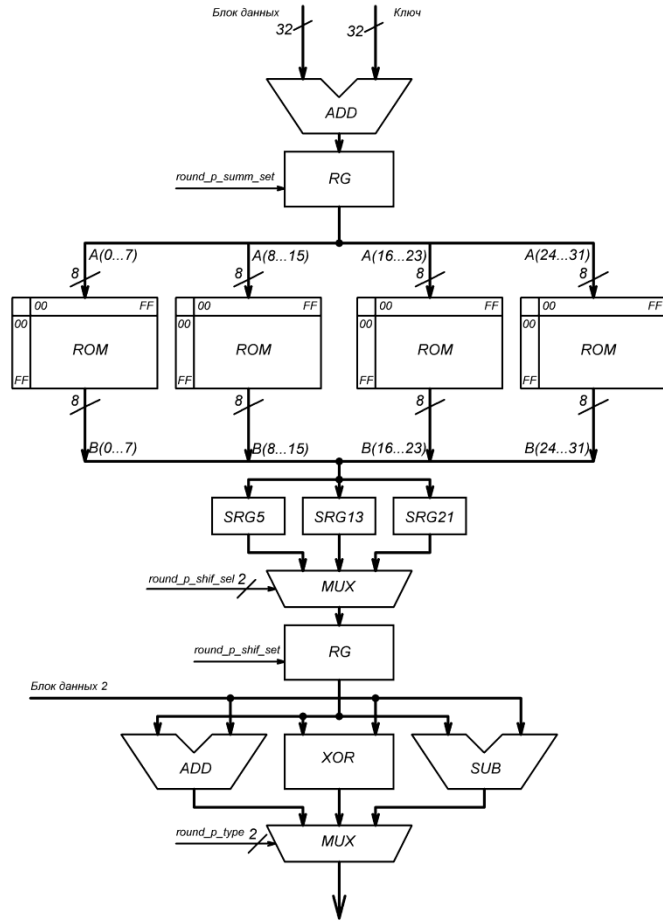


Рис. 1. Функциональная схема блока вычисления

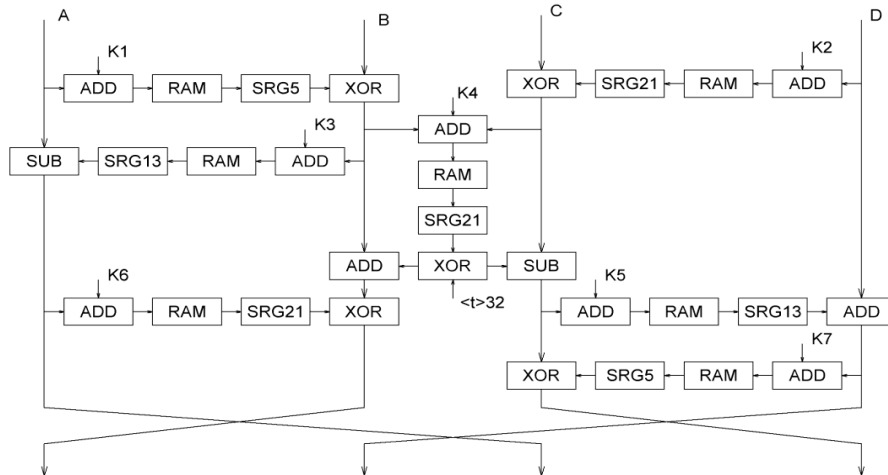


Рис. 2. Вычисления на t-м такте зашифрования

Для размещения проекта был выбран кристалл ПЛИС xc6vix130t, семейства Virtex 6 компании Xilinx, для описания логики работы устройства использовался язык VHDL[2].

В таблицах 1 и 2 приведены результаты проектирования, а также их сравнения с результатами, указанными в материалах [3], [4] и [5]:

Таблица 1. Производительность некоторых реализаций алгоритма

Варианты реализации	Количество тактов	Тактовая частота устройства, МГц.	Производительность, Мбит/сек	Устройство
---------------------	-------------------	-----------------------------------	------------------------------	------------

Belt_seq[3]	336	500	190,47	ПЛИС Spartan xc3s200
Belt_par[3]	211	500	289,59	ПЛИС Spartan xc3s200
Программная[5]	682	1000	187,68	Процессор Pentium III
Последовательная[4]	66	112,4	217,98	ПЛИС Virtex xc6vix130t
Конвейерная[4]	114(1)	217,4	243,65(27 827)	ПЛИС Virtex xc6vix130t
Уменьшенная конвейерная[4]	114(14)	217,4	243,65(1 987,6)	ПЛИС Virtex xc6vix130t
С параллелизмом на уровне такта	225	454.54	258,58	ПЛИС Virtex xc6vix130t

Примечание – Для конвейерных реализаций в скобках указана производительность после заполнения конвейера, а также количество тактов необходимых для формирования каждого следующего результата.

Таблица 2. Аппаратные затраты при размещении на кристалле FPGA

Варианты реализации	Slices	Триггеров	LUTs	RAMs	Устройство
Belt_seq[3]	750	649	1392	9	ПЛИС Spartan xc3s200
Belt_par[3]	1070	302	2050	28	ПЛИС Spartan xc3s200
Последовательная[4]	423	847	1173	-	ПЛИС Virtex xc6vix130t
Конвейерная[4]	9157	10209	14816	112	ПЛИС Virtex xc6vix130t
Уменьшенная конвейерная[4]	4829	5598	7948	28	ПЛИС Virtex xc6vix130t
С параллелизмом на уровне такта	399	1362	937	2	ПЛИС Virtex xc6vix130t

Список использованных источников:

1. СТБ 34.101.31-2007. Государственный стандарт Республики Беларусь. Криптографические алгоритмы шифрования и контроля целостности.
2. Бибило, П. Н. VHDL. Эффективное использование при проектировании цифровых систем / П. Н. Бибило, Н. А. Авдеев. – М.: СОЛОН-Пресс, 2006. – с. 344
3. Поляков, А. С. Характеристики аппаратной реализации некоторых симметричных алгоритмов шифрования / А. С. Поляков, В. Е. Самсонов // Информатика : ежеквартальный научный журнал, 2011. – №1.
4. Ланкевич, Ю. Ю. Процессор алгоритма шифрования «Belt» на базе ПЛИС / Ю. Ю. Ланкевич // Информационные технологии и системы 2013 (ИТС 2013) : материалы международной научной конференции, БГУИР, Минск, Беларусь, 23 октября 2013 г. / редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2013. – с. 190-191.
5. Агиевич, С. В. Алгоритм блочного шифрования BelT / С. В. Агиевич, В. А. Галинский, Ю. С. Харин, Н. Д. Микулич. Управление защитой информации, т.6, №4, 2002. – с.407–412

СПОСОБ ФОРМИРОВАНИЯ ВЫСОКОЧАСТОТНОЙ СОСТАВЛЯЮЩЕЙ СПЕКТРА СИГНАЛА В НИЗКОСКОРОСТНОМ АУДИОКОДЕРЕ

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

Герасимович В. Ю.

Петровский А. А. – д-р. техн. наук, профессор

Работа посвящена описанию варианта повышения качества реконструированного сигнала аудиокодера [1] при низких скоростях битового потока. Рассматривается способ формирования высокочастотной составляющей спектра выходного аудиосигнала путем копирования и масштабирования информации среднечастотной части спектра сигнала.

Принцип сжатия аудиоданных с потерями состоит в выделении из входной последовательности данных, которые позволяют компактно представить и синтезировать на стороне декодера выходной сигнал. Основная задача кодера заключается в поиске наиболее важных для восприятия человеком параметров. Так как количество выбираемых данных ограничено скоростью битового потока (битрейта) аудиокодера, часть информации о входном сигнале теряется. В силу особенностей восприятия человеком аудиосигнала, высокочастотные компоненты спектра менее perceptually важны, нежели низко- и среднечастотные [2]. Поэтому, при низких скоростях битового потока (т.е. малом количестве бит, выделяемых для передачи сигнала), параметры, соответствующие данной полосе могут быть проигнорированы и опущены, что повлечет за собой уменьшение качества реконструированного сигнала. Данный материал посвящен вопросу формирования высокочастотной составляющей спектра в аудиокодеру на основе согласованной подгонки

(СП)[1] при низких скоростях битового потока.

Алгоритм СП представляет любой сигнал $x(t)$ в виде линейной комбинации частотно-временных функций (называемых атомами) $g_{\gamma_n}(t)$, выбираемых из избыточного словаря D . В разрабатываемом кодере словарь атомов строится для каждого фрейма входного сигнала из самого сигнала с помощью пакета дискретного вейвлет преобразования (ПДВП). Индексы вектора $g \in D, (l, n) \in E$ – узлы дерева ПДВП (где l – номер уровня дерева, n – номер узла). Каждый узел дерева декомпозиции соответствует определенной частотной полосе. Алгоритм СП, в данном случае, оптимизирован таким образом, что отбираются только перцептуально важные атомы из словаря (на основе информации о порогах маскирования). Остановка выбора параметров осуществляется по достижению заданного количества, которое определяется целевым битрейтом.

Суть проблемы заключается в том, что при низких скоростях битового потока, ресурса атомов не хватает на выбор параметров из полос, эквивалентных высоким частотам. На рисунке 1 показан пример для 200 атомов отбираемых алгоритмом СП.

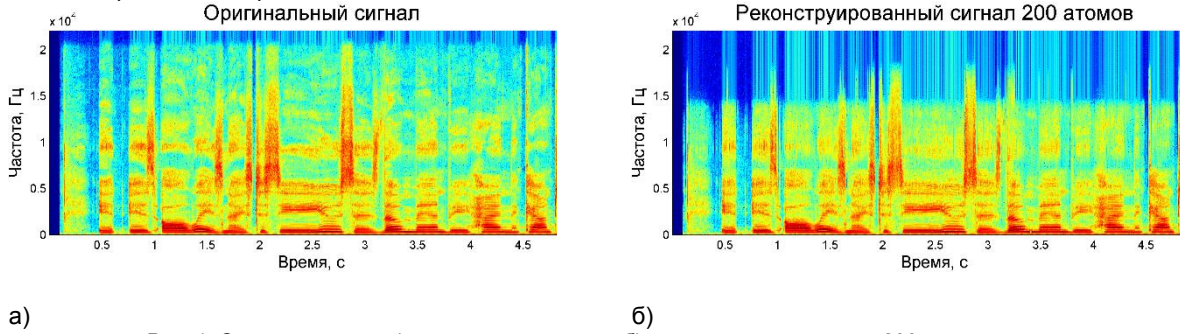


Рис. 1. Спектрограммы: а) – исходного сигнала, б) – реконструированного 200-ми атомами

Вариантом решения является перенос параметров из предыдущих каналов, в которых есть отобранные атомы, и их масштабирование к целевой энергии канала. При этом, вместо того, что бы передавать набор параметров для канала, декодеру для синтеза сигнала необходима только информация об энергии соответствующей частотной полосы. Схожий в своей идее метод применяется в [3], однако в рассматриваемом случае он применен в вейвлет области и масштабирование производится только на энергию канала, а не по спектральной огибающей. Первоисточником идеи является [4], однако применяемый в данной работе метод менее эффективен, нежели *SpectralBandReplication*, но, вместе с тем, менее ресурсоемкий, что критично при работе в реальном масштабе времени.

Для того, что бы оценить эффективность описываемого метода использовалась объективная оценка качества реконструированного сигнала *PEMO-Q*[5]. Данная метрика оценивает перцептуальное подобие сигналов (*PerceptualSimilarityMeasure* – *PSM*). Для того, что бы соотнести данную объективную оценку качества сигнала с субъективной оценкой *SDG* (*Subjective Difference Grade*), *PSM* можно отобразить в шкалу объективного различия (*Objective Difference Grade* – *ODG*), которая коррелирует с *SDG* и определяется следующим образом: 0.0 – не воспринимаемые искажения, -1.0 – воспринимаемые, но не раздражающие, -2.0 – немного раздражающие искажения, -3.0 – раздражающие, -4.0 – очень раздражающие искажения. Для эксперимента было взято три звуковых образца: вокал (*Suzan Vega, es01*), речь на немецком языке (*es02*), речь на английском языке (*es03*). Образцы были закодированы с использованием 200 – 400 атомов, с шагом в 50 атомов, что приблизительно соответствует скорости битового потока от 36,4 до 70,8 кбит/с с шагом в 8,6 кбит/с. Результаты приведены на рисунке 2.

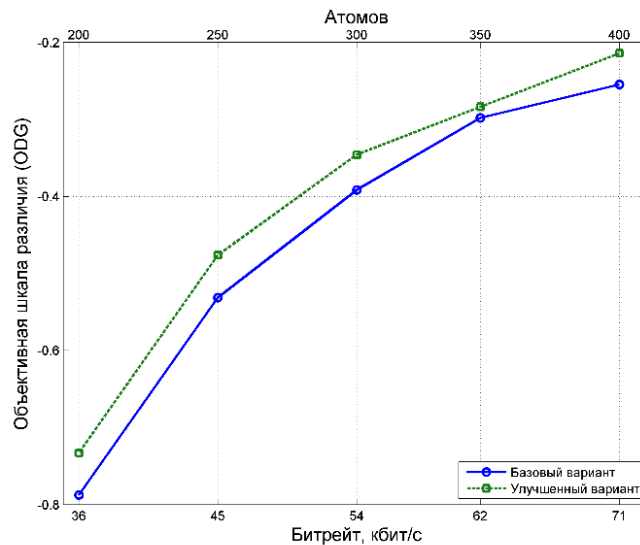


Рис. 2. Экспериментальные результаты

Как видно из экспериментальных результатов, приведенный способ формирования высокочастотной составляющей спектра сигнала даёт прирост качества восстановленного сигнала. При этом, в качестве дополнительных данных, которые необходимо передать декодеру для реконструкции выступает энергия в целевой частотной полосе, т.е. один параметр на полосу. В проведенных экспериментах перенос осуществлялся в три полосы, следовательно на каждый фрейм требовалось только три дополнительных параметра, что незначительно увеличит битрейт, при этом позволит получить более высокое качество выходного сигнала.

Список использованных источников:

1. Petrovsky, Al. Scalable parametric audio coder using sparse approximation with frame-to-frame perceptually optimized wavelet packet based dictionary / Al. Petrovsky, V. Herasimovich, A. Petrovsky // AES 138th Convention, paper 9264 – Warsaw, Poland, 2015, May 7 - 10.
2. Zwicker, E. Psychoacoustics: facts and models / E. Zwicker, H. Fastl. – Berlin: Springer-Verlag, 1990. – 354 p.
3. Valin, J.-M. High-Quality, Low-Delay Music Coding in the Opus Codec / J.-M. Valin, G. Maxwell, T. Terriberry, K. Vos // AES 135th Convention, paper 8942 – New York, USA, 2013, October 17 - 20.
4. Dietz, M. Spectral Band Replication, a novel approach in audio coding / M. Dietz, L. Liljeryd, K. Kjörling, O. Kunz // AES 112th Convention, paper 5553 – Munich, Germany, 2002, May 10 - 13.
5. Huber, R. PEMO-Q – A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception / R. Huber, B. Kollmeier // IEEE Transactions on audio, speech, and language processing, vol. 14. – November, 2006. – pp. 1902-1911.

СИСТЕМА ИДЕНТИФИКАЦИИ ДИКТОРА В АКУСТИЧЕСКИХ ШУМАХ НА ОСНОВЕ АНТРОМОРФИЧЕСКОЙ ОБРАБОТКИ РЕЧЕВОГО СИГНАЛА

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

Кручок Д. Н.

Петровский А. А. – д-р. техн. наук, профессор

Рассматривается система идентификации диктора в акустических шумах с использованием антропоморфической обработки речевого сигнала. Полученный характеристический вектор на базе данного преобразования используется в качестве признаков для системы идентификации диктора. В качестве решающих правил применяются нейронные сети прямого распространения. Приводятся результаты распознавания разработанной системы идентификации диктора.

В работе [1] был предложен алгоритм получения характеристического вектора, который пытается смоделировать процессы обработки речевого сигнала человеческим ухом. Рассматриваемый алгоритм состоит из этапов, представленных на рис. 1, а и частично моделирует процесс обработки звукового сигнала слуховой системой человека.

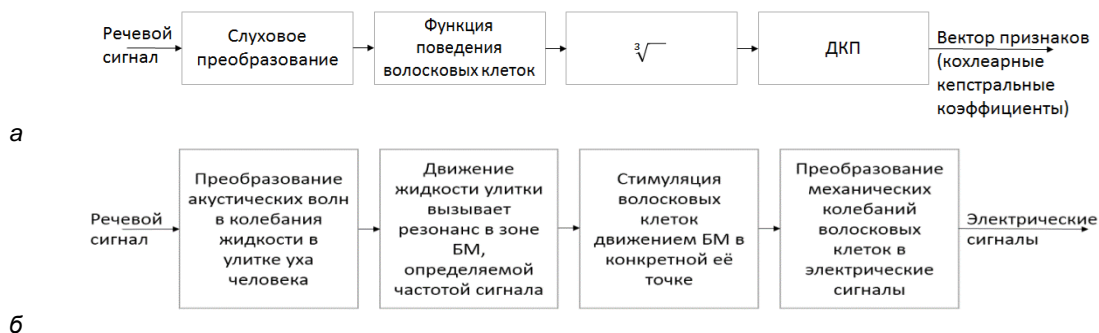


Рис. 1. – Сравнение этапов обработки сигнала: а – схема рассматриваемого алгоритма экстракции; б – процесс обработки звукового сигнала в слуховой системе человека

Полученные кохлеарные кепстральные коэффициенты (ККК) используются в системе идентификации в качестве признаков. В качестве решающих правил системы идентификации диктора наиболее часто используются следующие методы: векторное квантование, гауссовские смеси, нейронные сети и метод опорных векторов. В данном исследовании были выбраны нейронные сети прямого распространения для получения сравнительных результатов распознавания диктора с результатами, представленными в работе [1], в которой были использованы модели гауссовых смесей. Для экспериментальной оценки системы идентификации диктора в акустических шумах, был спроектирован идентификатор дикторов на основе нейронных сетей прямого распространения. Модель системы распознавания представлена на рис. 2 и имеет два режима работы: обучение и идентификация. Архитектура используемой сети выбиралась постепенно от

простого однослойного персептрона к многослойным экспериментально. Используемые типы нейронных сетей: многослойный персептрон (два и три слоя): 2 слоя: 56 входных – 19 выходных. 3 слоя: 56 входных – 30 промежуточных – 19 выходных. Функция активации на всех слоях, кроме выходного: сигмоид, на выходном – функция *softmax*. Процесс обучения происходит с учителем.

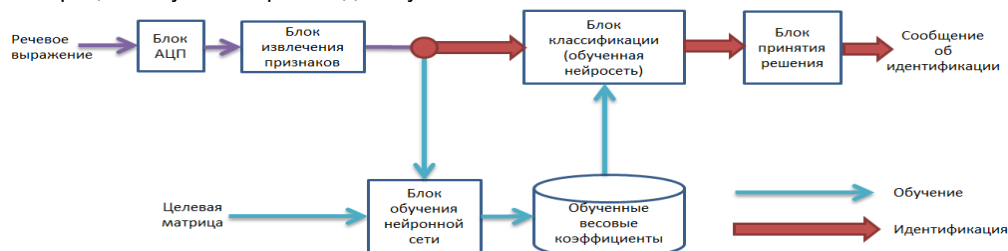


Рис. 2. – Модель системы распознавания

Используемая речевая база была подготовлена на основе базы, сформированной для соревнований по разделению и распознаванию речи и дикторов в условиях с шумами [2]. Было выбрано 4 диктора, у каждого по 18 речевых фраз средней продолжительностью около 2 секунд, не содержащих шумы.

В алгоритме экстракции характеристического вектора использовалось 32 кохлеарных фильтра [1]. Оценкой системы идентификации выступает точность распознавания диктора – отношение числа правильно распознанных речевых выражений к общему числу высказываний, участвовавших в распознавании. Данная оценка выражается в долях единицы или в процентах. Для получения сравнительных результатов был реализован алгоритм получения кепстральных коэффициентов в шкале Мел (мел-частотные кепстральные коэффициенты – МЧКК) и использован в системе идентификации диктора. Система идентификации реализована на языке Matlab. Результаты распознавания на обучающем множестве представлены на рис. 3. Тип используемого шума – лепет, бормотание (англ. «*babble*»).

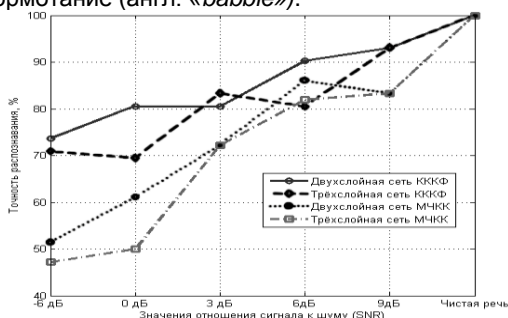


Рис. 3. Сравнение результатов для обучающего множества (тип шума – бормотание)

Как видно из рисунка 4 в условиях без шумов признаки ККК показывают сравнимые результаты с признаками МЧКК и имеют 100% точность распознавания (на ограниченном числе выражений). По мере того, как увеличивается уровень шума, точность системы распознавания падает, однако точность идентификации с использованием признаков ККК значительно лучше, чем МЧКК. Например, при SNR равным 0 дБ, точность распознавания для признаков МЧКК составляет 62 %, а для ККК – 80 %. На тестовом множестве точность распознавания всей системы снижается, однако результаты идентификации с использованием признаков ККК превосходят результаты МЧКК: 77% к 69% при SNR равным 0 дБ (рис. 4). Эффективность используемого алгоритма экстракции характеристического вектора в системе идентификации на нейронных сетях при уровне SNR -6 дБ больше на 20% для МЧКК, и на 5-10% для ККК, чем в системе на основе гауссовых смесей [1].

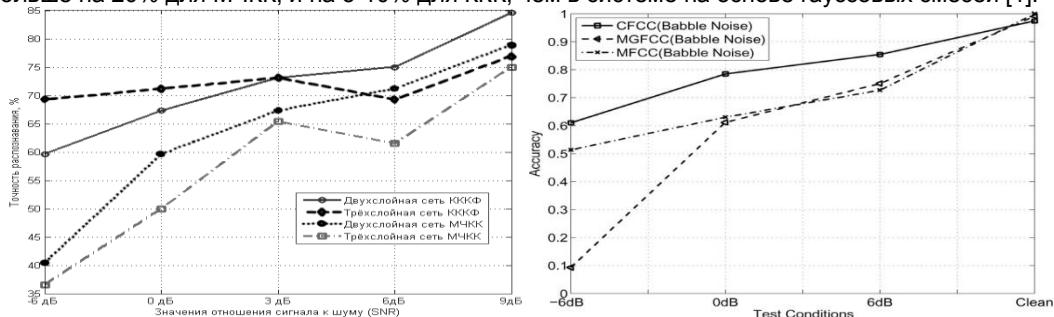


Рис. 4. Результаты распознавания для тестового множества (слева – система идентификации на нейронных сетях, справа – система идентификации на гауссовых смесях [1]) для типа шума – бормотание

Список использованных источников:

1. Q. Li, An auditory-based feature extraction algorithm for robust speaker identification under mismatched conditions / IEEE Transactions Audio, Speech, and Language Processing., – 2011. – Vol. 19, № 6. – P.1791-1801.
2. The PASCAL CHiME speech separation and recognition challenge [Electronic resource]. – 2013. – Mode of access :

СИСТЕМА УЛУЧШЕНИЯ ВОСПРИЯТИЯ И РАЗБОРЧИВОСТИ АУДИО СИГНАЛА В АГРЕССИВНОЙ ШУМОВОЙ СРЕДЕ

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

Белый В. Л.

Петровский А. А. – д-р. техн. наук, профессор

В современных системах улучшения восприятия и разборчивости аудио сигнала в агрессивной шумовой среде и в системах шумоподавления отдельную нишу занимают приложения, предназначенные для мобильных и портативных устройств, предполагающих использования наушников.

Универсальная система улучшения восприятия и разборчивости аудио сигнала требует достижения высокого уровня шумоподавления для шумов различной природы, различного уровня и различного частотного состава. Исходя из данных требований было принято решения разработать систему шумоподавления, представляющую собой двухканальную интеграцию субполосного аудио интегрированного алгоритма активного шумоподавления (ActiveNoiseControl – ANC) с маскированием, основанным на принципах психоакустического анализа.

На рисунках 1 и 2 приведены выбранные структуры субполосного аудио интегрированного алгоритма активного шумоподавления (ANC) и модели расчета коэффициентов усиления маскиера, базирующейся на принципах психоакустики:

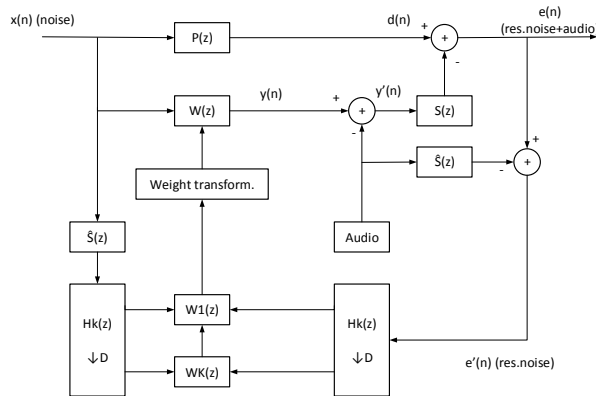


Рис. 1. Структурная схема ANC

Работу субполосного аудио интегрированного алгоритма активного шумоподавления (ANC) можно представить следующими формулами:

$$y(n) = \sum_{n=0}^{N-1} w(n)x(n - N)$$

$$y'(n) = y(n) - a(n)$$

$$e(n) = d(n) - y_s'(n)$$

$$e'(n) = e(n) - a'(n)$$

Где, $a(n)$ – целевой аудио сигнал; $y_s'(n)$ – $y'(n)$ после прохождения вторичного пути; $a'(n)$ – $a(n)$ после прохождения вторичного пути, N – порядок широкополосного адаптивного фильтра;

В качестве алгоритма адаптации коэффициентов субполосных фильтров использовался алгоритм FxLMS:

$$w_k(n + 1) = w_k(n) + \mu \frac{X(n)e(n)}{\epsilon + \|X(n)\|^2}$$

$w_k(n)$ – коэффициенты адаптивного фильтра k – субполосы; K – кол-во субполос; ϵ – небольшая константа для предотвращения деления на нуль, μ – шаг приращения; $X(n) = [x(n) \ x(n-1) \ x(n-2) \ \dots \ x(n-N+1)]$;

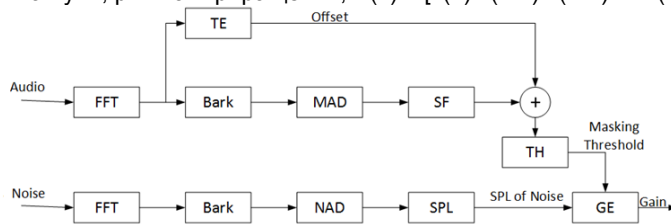


Рис. 2. Структурная схема психоакустической части

Основное преимущество интеграции данных алгоритмов заключается в двухэтапной обработке, которая позволяет достичь более высокой степени подавления шума окружающей среды. Алгоритм ANC является первым этапом обработки, на этап психоакустического маскирования приходится только остаточный и высокочастотный шум, т.к. алгоритм ANC имеет высокую эффективность в полосе частот до 1500 Гц. В качестве маскира на этапе психоакустического маскирования выступает целевой аудио сигнал, поскольку маскируется только остаточный шум, уровень которого предполагается достаточно низким, достигается небольшая вероятность того, что сигнал маскира будет усилен до критически опасного для человеческой слуховой системы уровня.

Структура двухканальной интеграции субполосного аудио интегрированного алгоритма активного шумоподавления с маскированием на основе психоакустического анализа представлена на рисунке 3:

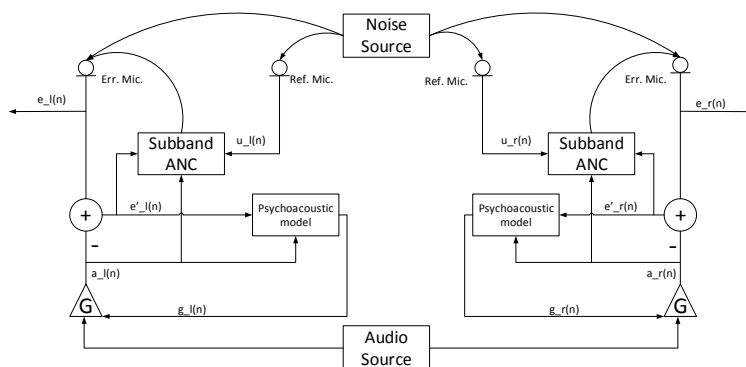


Рис. 3. Структурная схема модели интеграции

Независимая параллельная обработка двух каналов позволяет добиться более точной обработки шумовых сигналов, в независимости от их расположения относительно входов двух каналов. Для проведения тестирования предложенной модели интеграции использовался программный пакет MatLab. В ходе тестирования использовались актуальные первичный и вторичный путь наушников, оценка которых производилась в лабораторных условиях с использованием макета человека и человеческой слуховой системы DummyHead.

Таким образом, была разработана математическая модель улучшения восприимчивости и разборчивости аудио сигналов в агрессивной шумовой среде. Для построения модели использовалась идея интеграции субполосного аудио интегрированного алгоритма активного шумоподавления (ANC) и маскирования, основанного на принципах психоакустики. Предложенный двухэтапный вариант обработки шумового сигнала позволяет достичь более высокого уровня шумоподавления, что является главным преимуществом рассмотренной интеграционной системы.

Список использованных источников:

1. Конг-Аик Ли. Субполосная адаптивная фильтрация, теория и применение/ Конг-Аик Ли, Вун-Сенг Хан, Сен М. Куо. // – JohnWileyandSons,Ltd, 2009. – 324 с.
2. Петровский, А. А. Цифровые банки фильтров: анализ, синтез и применение в мультимедиа системах / А. А. Петровский, М. Парфенюк, А. Борович, М.З. Лившиц // Уч. метод. пособие для студентов учреждений, обеспечивающих получение высшего образования по специальности "Электронные вычислительные средства". – Минск, 2006. –81 с.
3. Bjamason, E. Analysis of filtered-X LMS algorithm. / Bjamason E. // - IEEE Trans. Speech audio Process., 3(6), 1995, 504-514.
4. Davari, P. Designing a new robust on-line secondary path modeling technique for feedforward active noise control systems/ Davari P, Hassanpour H. // - Signal processing, 89(6), 2009, 1195 – 1204.
5. Rees, L. Psychoacoustic modelling for active noise control system/ Rees L., Elliot S.J.// - Proc. Inst. Acoust., 26(pt.2), 2004.

ПРИМЕНЕНИЕ ЧИСЛЕННЫХ МЕТОДОВ ДЛЯ СИНТЕЗА КОЭФФИЦИЕНТОВ ПАРАУНИТАРНОГО БАНКА ФИЛЬТРОВ НА ОСНОВЕ АЛГЕБРЫ КВАТЕРНИОНОВ

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

Рыбенков Е. В.

Петровский Н. А. – ассистент кафедры ЭВС

Предлагается метод синтеза коэффициентов для параунитарного банка фильтров на основе алгебры кватернионов с использованием численных методов. Приведены характеристики полученной системы.

Параунитарный банк фильтров (ПУБФ) – банк фильтров, у которого передаточные функции анализирующих и синтезирующих фильтров и их смещенные версии ортогональны друг другу.

Серьезной практической проблемой, связанной с реализацией ПУБФ на арифметике с фиксированной запятой, является потеря свойства перфективной реконструкции сигнала банком фильтров. Это обусловлено чувствительностью структуры ПУБФ к квантованию коэффициентов [1]. Применение кватернионов позволяет избежать данного недостатка. В работе [2] показано, что для каждой ортогональной матрицы размерностью 4×4 существует уникальная (с точностью до знака) пара единичных кватернионов P и Q для которых выполняется следующее равенство:

$$M^+(P) \cdot M^-(Q) = M^-(Q) \cdot M^+(P).$$

Решетчатые структуры 4-канальных банков анализа и синтеза с линейной ФЧХ на кватернионах приведены в [2].

Поиск оптимальных коэффициентов банка фильтров ведётся с использованием численных методов, относительно критериев качества и ограничений на схемную реализацию. Факторизация такого банка фильтров гарантирует свойства перфективной реконструкции и линейность ФЧХ. Таким образом основной критерий оптимизации – поиск оптимальных коэффициентов в ограничениях арифметики с фиксированной точкой.

Практически все методы оптимизации функции n переменных основаны на многократном повторении следующих действий: выбор некоторого направления; спуск к минимуму вдоль направления спуска. Методы нулевого порядка при выборе направления спуска требуют только вычисления значений функции [3]. В данной работе задача оптимизации решается с использованием программы `fminsearch` из пакета MATLAB. Для задания ограничений использовался метод штрафных функций. Суть метода заключается в сведении задачи поиска условного оптимума к задаче нахождения безусловного оптимума.

Нахождение коэффициентов ПУБФ производится путем оптимизации следующих критериев [2]:
Степень ослабления в полосе задержки каналов, выраженная в терминах энергии:

$$\mathcal{E}_{SBE} = \sum_{k=0}^{M-1} \int_{\omega \in \Omega_k} |H_k(e^{j\omega})|^2 d\omega,$$

где Ω_k – полоса задержки k -ого фильтра банка фильтров.

Коэффициент эффективности кодирования

$$CG = 10 \lg \frac{\frac{1}{M} \sum_{k=0}^{M-1} \sigma_{xk}^2}{\left(\prod_{k=0}^{M-1} \sigma_{xk}^2 \right)^{\frac{1}{M}}}$$

где σ_{xk}^2 – дисперсия сигнала в k -ом канале банка фильтров.

Полярная форма кватерниона позволяет сократить число степеней свободы и легко параметризовать кватернионы с модулями равными единице. Полярная форма кватерниона определяется следующим образом:

$$\begin{cases} q_1 = |Q| \cdot \cos \varphi \\ q_2 = |Q| \cdot \sin \varphi \cdot \cos \psi \\ q_3 = |Q| \cdot \sin \varphi \cdot \sin \psi \cdot \cos \chi \\ q_4 = |Q| \cdot \sin \varphi \cdot \sin \psi \cdot \sin \chi \end{cases},$$

где φ, ψ, χ – произвольные значения углов.

В данной работе рассматриваются только вырожденные кватернионы ввиду более простого поиска оптимума. Вырожденный кватернион – кватернион, мнимые части i и j которого равны нулю, и в полярной форме представлен одним углом φ . Выбрана факторизация множителя в виде лестничной схемы с учётом особенностей кватерниона-константы [4].

Выбор начальной точки производится путем нахождения минимальных значений целевой функции $f(\vec{x}) = (\varepsilon_{SBE} - CG) / 2$, методом перебора всех возможных комбинаций входного вектора с большим параметром шага по каждой степени свободы.

Для того чтобы представить в графическом виде целевую функцию $f(\vec{x})$, изобразим в 3-х мерном виде соотношение углов φ_1 и φ_2 при зафиксированном $\varphi_3 = 1.08$ (рисунок 2). После квантования коэффициентов лестничной параметризации вырожденного кватерниона-константы (длина слова $N=2$) график функции $f(\vec{x})$ показан на рисунке 3.

Результат синтеза коэффициентов четырех канального ПУБФ на кватернионах с длиной фильтра равной 12 и линейной ФЧХ, приведен на рисунке 4. Минимальное ослабление в полосе задержки составляет -20дБ, эффективность кодирования равна 8.26дБ.

Таким образом, был предложен метод численного синтеза коэффициентов параунитарного банка фильтров в алгебре кватернионов. Применение вырожденных кватернионов позволило сократить объем вычислений. Полученный ПУБФ сопоставим с ранее представленными результатами в [2], и обладает свойством перфективной реконструкции для реализации в арифметике с фиксированной точкой.

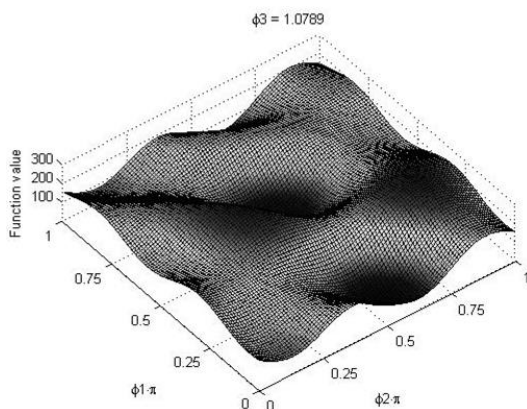


Рис. 1. График функции $f(\vec{x})$ при $\varphi_3 = 1.08$

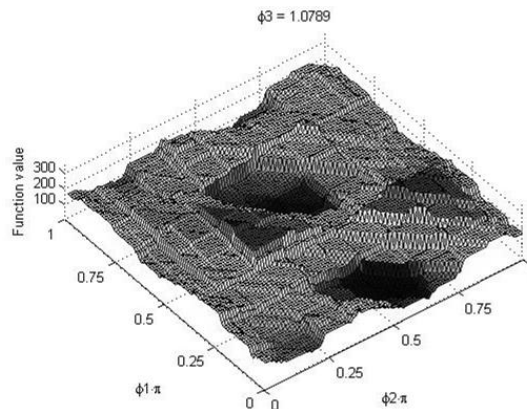


Рис. 2. График функции $f(\vec{x})$ при $N = 2$

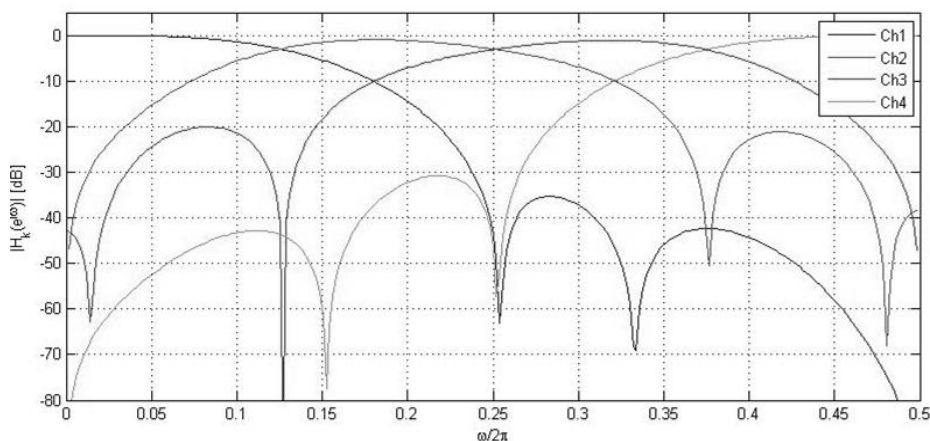


Рис. 3. АЧХ ПУБФ на кватернионах

Список использованных источников:

1. Vaidyanathan P.P. The role of lossless systems in modern digital signal processing: a tutorial / P.P. Vaidyanathan, Z. Doganate // IEEE Trans. on Education, Vol.32, №3, pp.181-197.
2. Парфенюк М. Параунитарные банки фильтров на основе алгебры кватернионов: теория и применение / М. Парфенюк, А.А. Петровский // Журнал "Цифровая обработка сигналов". – 2008 - №1 – С. 22-36.
3. Сеницын А.К. Вычислительная математика : Лекции для студентов 2-ого курса специальностей 1-39 01 03 / А.К. Сеницын, А.А. Навроцкий – изд. БГУИР, 2008. – 50 с.
4. Петровский Н.А. Параунитарные банки фильтров без потерь с линейной ФЧХ на основе DSP48 блоков в FPGA

архитектуре / Н.А. Петровский // Труды международной научной конференции ИТС 2014 (Информационные технологии и системы). – 2014. – С. 190-191.

КВАНТОВАНИЕ И УПАКОВКА КОЭФФИЦИЕНТОВ СУБПОЛОС В КОДЕРЕ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ПАРАУНИТАРНОГО БАНКА ФИЛЬТРОВ

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

Арабей К. В.

Петровский А. А. – д-р. техн. наук, профессор

В последнее время банки фильтров рассматриваются как одна из наиболее эффективных техник компрессии мультимедиа данных. Банки фильтров широко применяется в области кодирования видео и аудио сигналов а также изображений.

Существующие форматы кодирования изображений, например JPEG 2000, могут осуществлять компрессию данных, как с потерями, так и без потерь. При этом для каждого типа компрессии необходимо проводить отдельную процедуру трансформационного преобразования, что требует дополнительных вычислительных затрат. Кодер изображений на основе параунитарного банка фильтров, позволит использовать одно трансформационное преобразование как для компрессии с потерями, так и для компрессии без потерь.

Банк фильтров – цифровая система, состоящая из банка фильтров анализа и банка фильтров синтеза (рис. 1). Входной сигнал $x(n)$, который представляется последовательностью отсчетов, разбивается на M субполосных составляющих при помощи фильтров блока анализа $H_k(z)$ ($k = 0, 1, \dots, M-1$). В идеальном случае эти составляющие в частотной области не перекрываются. Операции, выполняемые блоком синтеза, являются обратными операциями блока анализа. Подобранным соответствующим образом набор фильтров блока синтеза $F_k(z)$ ($k = 0, 1, \dots, M-1$), можно восстановить исходный сигнал $y(n)$ из его субполосных компонент.

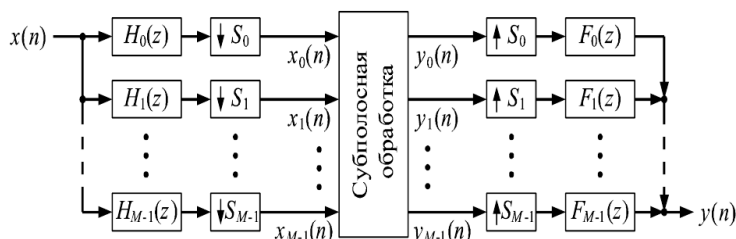


Рис. 1 – Банк фильтров: система анализа/синтеза сигнала

Параунитарный (ортогональный) банк фильтров (ПУБФ) – банк фильтров, у которого передаточные функции анализирующих и синтезирующих фильтров и их соответственно смещенные версии ортогональны друг другу. Фильтры синтеза в параунитарных банках являются транспонированными версиями фильтров анализа:

$$F_k(z) = H_k^T(z^{-1})$$

При соблюдении этого условия обеспечивается возможность перфективной реконструкции банком фильтров входного сигнала $x(n)$.

Преобразование при помощи параунитарного банка фильтров не осуществляет сжатия данных, оно необходимо для подготовки информации к этапам компрессии с потерями упаковки. При работе с изображениями преобразование идет в два этапа: сначала по строкам, потом по столбцам. После преобразования основная часть информации концентрируется в низкочастотных субполосах, высокочастотные полосы представляют матрица, в которой многие коэффициенты либо близки, либо равны нулю. Кроме того, благодаря несовершенству человеческого зрения можно аппроксимировать коэффициенты более грубо без заметной потери качества изображения. Для этого используется квантование коэффициентов. В самом простом случае - это арифметический побитовый сдвиг вправо. При этом преобразовании теряется часть информации, но может достигаться большая степень сжатия.

Обычно в алгоритмах кодирования изображений используется скалярное квантование. Числовая прямая разбивается на отрезки равной длины. Далее, квантуемый элемент, попадающий в какой-то отрезок, заменяется центром этого отрезка. Для каждой субполосы выбирается свой коэффициент квантования, что позволяет регулировать степень сжатия изображения. Как правило, их значения для субполос растут по направлению слева направо и сверху вниз. Операция квантования является единственной фазой, где

происходит потеря информации.

Далее матрицы коэффициентов субполос переводится в вектор при помощи "зигзаг"-сканирования. Полученный вектор свертывается с помощью алгоритма группового кодирования (RLE). Алгоритм RLE заключается в следующем: любой последовательности повторяющихся входных символов ставится в соответствие набор из двух выходных символов: первый байт, определяющий длину входной последовательности, второй — сам входной символ.

На следующем этапе проводится Кодирование по Хаффману. В основе алгоритма кодирования Хаффмана лежит довольно простой принцип: символы заменяются кодовыми последовательностями различной длины. Чем чаще используется символ, тем короче должна быть кодовая последовательность. Кодовая таблица строится на основе статистического анализа имеющейся информации. Код Хаффмана обладает свойствами префиксных кодов и легко может быть свернут обратно в последовательность длин серий.

Сжимая файл по алгоритму Хаффмана, первое, что нужно сделать - подсчитать сколько раз встречается каждый символ. После подсчета частоты вхождения каждого символа, необходимо построить таблицу кодов и сформировать мнимую компоновку между кодами по убыванию. Для восстановления первоначального файла, нужно иметь декодирующую таблицу, так как они будут различны для разных файлов. Следовательно, необходимо сохранять таблицу вместе с файлом.

Для реализации вышеописанных алгоритмов был использован язык технических вычислений MatLab. Таким образом, была разработана модель кодера изображений на основе параунитарного банка фильтров в ряде случаев превосходящая по скорости вычислений и степени сжатия уже существующие кодеры.

Список использованных источников:

1. Петровский, А. А. Цифровые банки фильтров: анализ, синтез и применение в мультимедиа системах : учебно-методическое пособие / А.А. Петровский [и др.]. – Минск, БГУИР, 2006. – 82 с.
2. Парфенюк, М. Параунитарные банки фильтров на основе алгебры кватернионов: теория и применение/ Парфенюк М., Петровский А.А - Минск, БГУИР, 2008. – 14с.
3. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.

ОПИСАНИЕ ПРОЦЕССОРНОГО ЯДРА MOTOROLA MC6800 НА ЯЗЫКЕ ОПИСАНИЯ АППАРАТУРЫ VHDL

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

Винокуров В. В.

Герасимович В. Ю. – аспирант, профессор

Рассмотрен один из методов проектирования процессорного ядра Motorola 6800 с использованием языка описания аппаратуры, показаны основные проблемы на различных этапах проектирования.

Введение

Микропроцессоры представляют собой сложную структуру связанных между собой различных компонентов цифровой техники. Проектирование процессорного ядра является очень трудоемкой задачей, которая занимает колоссальное количество времени и человеческого труда. Без соответствующих САПР, этот процесс занимало бы не один год и увеличивало себестоимость в несколько раз, что в современном темпе развития техники, делает продукт полностью не конкурентоспособным. Одним из главных методов проектирования аппаратуры, является проектирование с использованием языков описания интегральных схем.

Основная цель моей работы – на практике, опробовать VHDL в качестве инструмента для проектирования микропроцессоров, ознакомиться с основными проблемами, с которыми сталкиваются начинающие системотехники. Проектируемая модель в точности повторяет архитектуру известного микропроцессора Motorola 6800. Motorola 6800 — микропроцессор, разработан и выпущен компанией Motorola в 1974 году. Это был первый микропроцессор с индексным регистром.

Основная часть

Перед началом реализации процессора на VHDL, требовалось как можно точнее изучить внутреннюю архитектуру и принципы работы ядра MC6800. В качестве основного источника информации использовалась официальная документация на микропроцессор, в которой были описаны основные режимы работы процессора, а также в общих чертах показана его структура. Вот тут и появляется первая проблема, для молодых специалистов.

Любая крупная IT фирма всегда утаивает ключевые аспекты построения своей архитектуры. Motorola не исключение. В частности, отсутствовала какая-либо информация об организации внутренних шин ядра и взаимодействии между собой основных компонентов. В таком случае проектирование происходит на основе

общего понимания организации ЭВМ.

Непосредственное проектирование VHDL модели выполнялось модульно. В структуру MC6800 входят: АЛУ, 2 8-разрядных аккумулятора, 2 16-разрядных индексных регистра, программный счетчик, регистр команд и устройство управления. Все модули, кроме УУ, описывались по стандартным методам и использовались шаблоны, встроенные в Xilinx ISE.

В конечном счете архитектура MC6800 следующая. Общая шина данных представляет собой 3 8-разрядные шины. Шина А и В связывает выходы регистров с информационными входами АЛУ соответственно, а шина С – выход АЛУ с входами регистров. Передача данных между шинами А(В) и С осуществляется через АЛУ. Так же шины В и С выходят на двунаправленный буфер данных.

Самый сложный компонент – устройство управления. Устройство представляет из себя конечный автомат, состояниями которого является этапы выполнения команд процессора. Вторая проблема для начинающих – разобраться в методах кодирования команд процессора. Как правило, каждая команда – набор элементарных действий, которые повторяются в разной последовательности и в разном количестве от команды к команде. Эти элементарные действия и их последовательность кодируются в команду. Понимание принципов кодирования помогает описать устройство управления очень лаконично, не расписывая каждую команду в отдельности. Реализация устройства управления MC6800 и была основана на конечном автомате, который реализовывал разные стадии выполнения программы. Конечный автомат определяет стадию выполнения команды и в заданные моменты времени подает сигналы управления на все модули (пример: записи данных в аккумулятор, нужную команду АЛУ). По окончании выполнения команды - переходит на состояние выборки следующей команды.

Только к концу проектирования, были выявлены несколько методов оптимизации кода на основе кодирования команд, описанной выше.

В свете выше сказанного, хочется отметить что проектирование любого цифрового устройства является решением комплексной задачи аналитики и проектирования с применением лучших решений в данной области. А применение разнообразных САПР и HDL позволяет оптимизировать работу проектирования, обеспечивает ее гибкость. Данная работа является прямым применением всех наших знаний, полученных за время обучения в нашем ВУЗе.

Список использованных источников:

1. Perry D.L. VHDL: Programming by Example. Fourth Edition. McCraw-Hill, 2002. – 476 p.
2. Бибилко П.Н. Основы языка VHDL. Второе издание. — М.: Солон-Р, 2002. — 224 с

РАЗРАБОТКА ИГРОВОГО ПРОЕКТА НА UNREALENGINE 4

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

Корсаков А. А.

Разработка игр является комплексной и весьма сложной задачей. Основная сложность заключается в том, что игровые проекты включают в себя множество областей: графика, анимация, обработка огромных объемов данных в реальном времени, искусственный интеллект, геймдизайн, социальное программирование, повествование, взаимодействие с игроками и так далее. Над проектом, обычно, трудятся команды из огромного числа специалистов в совершенно разных областях.

Можно утверждать, что на данный момент геймдев как никогда близок к рядовым пользователям. С развитием индустрии начала появляться профильная литература, курсы и специальности в учебных заведениях, а пользователи всё чаще пытаются делать свои проекты, тем самым продвигая такое направление, как Инди-разработка.

Я и моя команда не стала исключением. Нам пришлось пройти долгий путь, впитывая в себя огромные объемы информации, перенимая опыт успешных коллег из-за рубежа и, попутно, экспериментировать и выносить для себя собственные гипотезы и умозаключения.

Для своей работы мы выбрали игровой движок UnrealEngine 4, который, если можно так выразиться, является своеобразным САПРом для создание игр. Его преимуществами является бесплатная модель распространения и, что самое важное, открытый код, который позволяет переписывать любые аспекты системы под свои нужды. UE4 базируется на C++, что даёт ему ощутимый выигрыш в скорости работы относительно конкурентов.

Основными проблемами при разработке игровых проектов являются обеспечение наибольшего быстродействия, организация архитектуры таким образом, чтобы возможно было в кратчайшие сроки вносить изменения и расширять систему, необходимость создания большого количества сопутствующих продуктов, призванных повысить качество и скорость разработки.

Развитие игровой индустрии тесно связано с общим прогрессом в области IT. В частности, на данный

момент именно геймдев сильнее всего стимулирует развитие технологий, связанных с компьютерной графикой и рендером. Именно в разрезе игр впервые обрело законченный вид такое направление, как «виртуальная реальность». Так же, растущие объёмы обработки подталкивают к созданию всё более быстродейственных алгоритмов.

Как уже отмечалось ранее, любой контент, который создаётся для игрового проекта, должен обладать максимальной гибкостью, иметь возможность быстро изменять параметры и расширяться. Как пример можно привести создание искусственного интеллекта. Ограничения по производительности сразу ставят жесткие рамки для написания логики. Так же, крайне важно, чтобы, написав один модуль, можно было получить множество совершенно различных ИИ, регулируя лишь несколько параметров. Благодаря этому, экономится не только время на разработку, но и уменьшается вероятность возникновения ошибок.

Так же, при разработке игрового проекта, большое значение имеет раннее прототипирование. Крайне важно иметь возможность быстро проверить идею на жизнеспособность, затратив на это минимум сил и времени. Данный этап во многом зависит от технологий, которые используются при разработке, и от навыков команды. На данном этапе крайне важно понимать, что качество не имеет никакого значения. Основная идея – посмотреть на то, как будет воспринят итоговый продукт потенциальной аудиторией.

Подводя итог: разработка игр является комплексной задачей, которая за своей кажущейся несерьёзностью скрывает внушительный объем решаемых технических задач и, в общем, вовсе не является тривиальной. Популярность данного направления в данный момент привлекает всё более серьёзные вложения и, как следствие, технологии в геймдев и превращает его в серьёзного игрока на рынке IT.

Список использованных источников:

1. Документация к UnrealEngine 4, URL: <https://docs.unrealengine.com/latest/INT/>
2. Scott Rogers "Levelup! TheGuidetoGreatVideoGameDesign". – Wiley, 2014. – 492 с.

СИСТЕМАУПРАВЛЕНИЯ КОДОВЫМ ЗАМКОМ НА БАЗЕ ПЛАТФОРМЫ ARDUINOMEGA 2560

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

Порхун М. И.

Качинский М. В. – кандидат технических наук, доцент

В современном мире одной из главных задач является обеспечение безопасности. Это обязывает использовать системы контроля доступа к информации. Одним из представителей такой системы является кодовый замок.

Применение подобного способа контроля доступа в помещение подразумевает использование секретной комбинации десятичных цифр (в нашем случае четырёх), позволяющей осуществлять управление кодовым замком. Для реализации системы был выбран микроконтроллер ATmega 2560, как одна из самых популярных платформ для реализации различных устройств. Платформа запрограммирована в соответствии с алгоритмом работы устройства на языке, схожим с C/C++. Для проверки работы устройства выбрана система автоматизированного проектирования Proteus.

Данный кодовый замок обеспечивает:

- 1) Доступ в помещение только после набора секретной кодовой комбинации, представляющей собой четырёхзначное десятичное число;
 - 2) Предоставление ограниченного количества попыток (в нашем случае трёх) набора секретной кодовой комбинации;
 - 3) Звуковую и визуальную сигнализацию, а также блокировку двери на пять минут при попытке несанкционированного доступа;
 - 4) Оперативную смену секретной кодовой комбинации;
 - 5) Подсветку клавиатуры при приближении человека к двери;
 - 6) Оригинальное звуковое сопровождение при открытии двери;
- Структурная схема устройства приведена на рисунке 1.

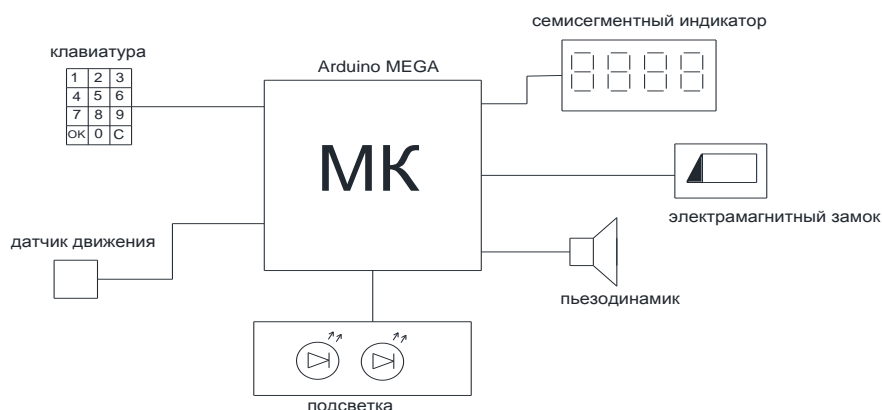


Рис. 1. Структурная схема системы управления кодовым замком

Упрощённый алгоритм работы устройства приведён ниже:

Шаг 1. Начальная инициализация состояния микроконтроллера;

Шаг 2. Анализ датчика движения:

- Если он активен в течение пяти секунд, то перейти на шаг 3;
- Если нет, то вернуться к первому шагу.

Шаг 3. Зажечь подсветку клавиатуры;

Шаг 4. Анализ клавиатуры (ожидание нажатия клавиши (а точнее четырёх клавиш)):

- 1) Для цифровых клавиш:
 - Вывод десятичного эквивалента нажатой цифры на семисегментный индикатор;
 - Запись кода нажатой клавиши в буфер;
- 2) Для функциональных клавиш:

Если нажат сброс:

- Очистить дисплей и буфер кодов введённых клавиш. Перейти к шагу 2.

Если нажата клавиша ОК:

- Проверить соответствие секретной кодовой комбинации введённой комбинации;
- Если коды совпадают, то перейти к шагу 5, если нет, то к шагу 7.

Если введена последовательность четырёх нулей (режим программирования):

- Просьба ввести секретную комбинацию;
- Если нажата клавиша сброса, то очистить семисегментные индикаторы и буфер введённых клавиш, перейти к шагу 2;
- Если она верна, то просьба ввести новый код, если нет, то перейти к шагу 2;
- Сохранение кода, переход к шагу 2.

Шаг 5. Процедура открытия замка:

- Открыть замок;
- Очистить семисегментный индикатор;
- Проиграть звуковой сигнал;
- Подождать десять секунд. Перейти к шагу 6.

Шаг 6. Закрыть замок, выключить подсветку клавиатуры. Перейти к шагу 2.

Шаг 7. Процедура обработки ошибочно введённой секретной кодовой комбинации:

- Инкрементировать «счётчик ошибок»;
- Проверить, третий ли раз был неверно введён код? Если да, то:
 - Перевести клавиатуру в «режим блокировки» (на пять минут);
 - «Поморгать» подсветкой клавиатуры (в течение десяти секунд);
 - Проиграть звуковое предупреждение;
 - Ждать пять минут;
 - После истечения этого времени перейти к шагу 2.
- Если нет, то перейти к шагу 2.

Замечания по данному алгоритму:

- 1) Клавиша ОК сбрасывает только после ввода четырёх цифровых значений;
- 2) Сброс семисегментного индикатора и буфера нажатых клавиш производится нажатием на клавишу С;
- 3) Вход в режим программирования осуществляется путём ввода заранее определённой кодовой комбинации (четыре нуля) и только при сброшенном буфере нажатых клавиш;

Для определения надёжности системы была построена её модель в САПР Proteus и протестирована в различных ситуациях. Симуляция показала, что устройство работает корректно.

Таким образом, была разработана система управления кодовым замком на базе платформы

Arduino MEGA 2560. Рассматриваемая система за счет представленного вышеалгоритма обеспечивает отличную защиту от несанкционированного доступа в помещение, звуковую и визуальную сигнализацию, при попытке несанкционированного доступа, оперативную возможность смены секретной кодовой комбинации и простой, интуитивно понятный для пользователя интерфейс – всё это выделяет ее на фоне других систем.

Список использованных источников:

1. Соммер У. Программирование микроконтроллерных плат Arduino/Freduino / Соммер У // Для радиолюбителей. – СПб.: БХВ – Петербург, 2012 г. – 256 с.
2. Сайт, посвященный программированию плат Arduino [Электронный ресурс]: 2016 г. URL: <http://arduino.ru/>
3. Сайт, посвященный программированию плат Arduino [Электронный ресурс]: 2016 г. URL: <http://arduino.cc/>

СИСТЕМА ДЕТЕКТИРОВАНИЯ КЛЮЧЕВЫХ ФРАЗ В РЕЧЕВОМ СИГНАЛЕ

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

Суша А. В.

Вашкевич М. И. – к-т. техн. наук, доцент

В работе приводится краткая характеристика основных методов детектирования (поиска) ключевых фраз в речевом сигнале. Так же описывается алгоритм работы системы, построенной с применением метода поиска по шаблону. Сравнение шаблона ключевой фразы с фрагментом сигнала осуществляется с помощью DTW-алгоритма.

Поиск ключевых фраз (англ. – keyword spotting) – речевая технология, связанная с решением задачи поиска и локализации заданного слова или целой фразы в речевом потоке. Задача поиска ключевых слов в речевом потоке может быть разделена на несколько направлений:

- непосредственно выявление и определение места ключевого слова или словосочетания (поиск в фонограмме речи, речевых базах, поиск в реальном времени);
- распознавание команд в слитном потоке речи;
- понимание смысла речи посредством поиска ключевых слов или фраз, для диалоговых систем.

Для решения задачи поиска ключевых фраз было разработано множество различных способов, из которых можно выделить основные методы. Ниже приведены широко используемые в настоящее время методы поиска ключевых фраз [1].

Метод *поиска по шаблону* является наиболее распространённым. Суть его заключается в том, что для каждого ключевого слова записывается несколько вариантов его произнесения различными дикторами, на основе чего создаётся шаблон, который используется для организации поиска данного слова в речевом потоке. Главным недостатком такого подхода является то, что создание шаблона требует запись каждого слова микрофоном (слово нельзя просто ввести с клавиатуры).

Скрытые марковские модели (сокращённо – СММ) широко используются для решения задачи поиска ключевых фраз. Для каждого ключевого слова создаётся соответствующая СММ, которая используется при поиске в речевом потоке, и одну «модель мусора» для всех остальных слов. Временная последовательность символов ключевых слов и мусор символов формируется в результате распознавания речевой последовательности. Недостатком метода является то, что для каждого нового ключевого слова нужно не только обучать новую СММ модель, но также нужно заново обучать модель мусора.

Стоит отметить метод *поиска ключевых фраз по фонемной (слоговой) решётке*. Метод основывается на конструировании большой сети возможных звуков (слогов) в различные моменты времени, по которой осуществляется поиск возможных произношений слова или фразы. Основное преимущество этого метода в том, что он обладает большой гибкостью: даже если фонема ключевого слова не является лучшей гипотезой между узлами решётки, она все равно сохраняется в результате распознавания. Но в то же время, построение такой решётки довольно трудоёмкий процесс.

В качестве основы был выбран метод поиска ключевых фраз по шаблону. Данный метод был выбран в связи с его относительной простотой и отсутствием необходимости предварительного обучения и переобучения. Описанный далее алгоритм работы системы детектирования ключевых фраз в речевом сигнале основывается на алгоритмах, описанных в публикациях [2, 3]. Структурная схема алгоритма приведена на рисунке 1. Рассмотрим алгоритм работы системы по шагам.

Подготовительный шаг. Осуществляется фрагментация шаблона ключевого слова на M фреймов, длиной 30 мс. На основании из каждого i -го фрейма формируется вектор характеристических признаков $\bar{a}_i, i = 0, M-1$. В качестве этих признаков были выбраны *мел-частотные кепстральные коэффициенты* (сокращённо – MFCC) [4]. Длина вектора характеристических признаков выбирается произвольно, в работе использовались 13 коэффициентов. В результате этих действий, формируется последовательность векторов характеристических признаков $A_M = \{\bar{a}_1, \bar{a}_2, \dots, \bar{a}_{M-1}\}$.

Шаг 1. Из речевого потока выбирается фрагмент данных, равный по длительности $2M$ фреймов шаблона ключевого слова. Далее осуществляется формирование векторов MFCC $\bar{b}_j, j = \overline{0, 2M-1}$. В результате получаем последовательность векторов характеристических признаков $B_{2M}^n = \{\bar{b}_1, \bar{b}_2, \dots, \bar{b}_{2M-1}\}$ фрагмента речевого сигнала в некоторый момент времени n .

Шаг 2. Из последовательности B_{2M}^n в момент времени n выбираются несколько подпоследовательностей $B_L^i, i = \overline{0, H-1}$, таких, что $L \in [0,5M; 2M]$. Число этих подпоследовательностей выбирается произвольно, но следует отметить, что большое их число существенно замедлит работу алгоритма.

Шаг 3. Каждая i -я подпоследовательность B_L^i сравнивается с последовательностью ключевого слова A_M посредством *DTW-алгоритма* [5], в результате строится таблица минимальных дистанций $d_k, k = \overline{0, K-1}$ между последовательностями $D_K^i = \{d_0, d_1, \dots, d_{K-1}\}$. Стоит отметить, что DTW-алгоритм позволяет сравнивать последовательности разных размеров, а также помимо привычных скалярных значений этих последовательностей может оперировать и векторами, при условии, что имеется функция расчёта расстояния между ними. На основании этой таблицы рассчитывается оценка расстояния между последовательностями по формуле:

$$\bar{D}^i = \frac{1}{N+L} \cdot \sum_{k=0}^{K-1} d_k \quad (1)$$

Шаг 4. На основании полученных значений оценочных расстояний определяется значение локальной минимальной дистанции $LMD(n) = \min(\bar{D}^i), i = \overline{0, H-1}$, где H – число подпоследовательностей, а n – рассматриваемый момент времени.

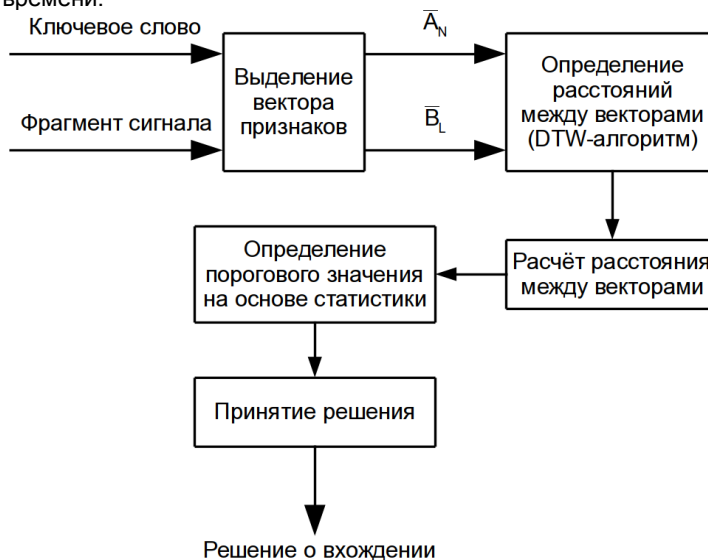


Рис. 2. Структурная схема алгоритма работы системы детектирования ключевых фраз в речевом сигнале

Шаг 5. Действия, описанные в шагах 1-4 выполняются до полной проверки речевого потока. Фрагменты сигнала выбираются с некоторым перекрытием.

Шаг 6. На основании полученных значений локальных минимальных дистанций определяется пороговое значение, которое используется для последующего определения вхождения ключевого слова:

$$TH = M[LMD] - c \cdot \sigma, \quad (2)$$

где M – математическое ожидание;

σ – среднеквадратичное отклонение.

Шаг 7. Используя полученное пороговое значение определяется вхождение ключевого слова в рассмотренные ранее моменты времени n , выбранные с перекрытием. Для этого рассчитывается число «попаданий» ключевого слова в некоторой окрестности момента n , с максимальным удалением S :

$$hit_l = \sum_{l=n-S}^{n+S} TH - LMD(l). \quad (3)$$

Значение hit_l указывает на вхождение ключевого слова в фрагмент речевого сигнала в некоторый момент времени n .

Стоит отметить, что в текущем виде описанный выше алгоритм не пригоден для использования в системах реального времени в связи с необходимостью сбора статистики для определения порогового значения, которое используется при принятии решения. Работы по преодолению данного ограничения производятся.

Список использованных источников:

1. Chandra, Dr.E. Keyword Spotting: An Audio Mining Technique in Speech Processing / Dr.E. Chandra, K.A. Senthildevi // IOSR Journal of VLSI and Signal Processing. – 2015. – №5. – с. 22-27.
2. Barakat, M. S. Keyword spotting based on the analysis of template matching distances / M. S. Barakat, C. H. Ritz, D. A. Stirling // Signal Processing and Communication Systems. – 2011. – с. 1-6.
3. John Sahaya Rani Alex. Spoken Utterance Detection Using Dynamic Time Warping Method Along With a Hashing Technique / John Sahaya Rani Alex, Nithya Venkatesan // International Journal of Engineering and Technology. – 2014. – №2 (6). – с. 1100-1108.
4. Zheng Fang Comparison of Different Implementations of MFCC / Zheng Fang, Zhang Guoliang, Song Zhanjiang // J. Comput. Sci. A Technol. – 2011. – №6 (16). – с. 582-589.
5. Eamonn, J.K. Derivative Dynamic Time Warping / J.K. Eamonn, J.P Michael // SIAM International Conference on Data Mining. – 2001. – с. 11.

СИСТЕМА ИДЕНТИФИКАЦИИ ДИКТОРА ПО ОБРАЗЦУ ГОЛОСА НА БАЗЕ РЕЧЕВЫХ ЗАПИСЕЙ

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

Вашкевич Г. С.

Вашкевич М. И. – канд. техн. наук, доцент

В данной работе описывается общий подход к построению систем идентификации диктора, проводится обзор современных методов и подходов к решению данной задачи, а также представляется пример реализации системы идентификации диктора на основе векторного квантования.

В условиях интенсивного развития систем распознавания речи все чаще перед разработчиками стоит задача не только распознать речь, но и установить личность диктора. Такие системы нашли широкое применение в криминалистике, системах контроля доступа и верификации, а также в обработке речевых баз данных.

На данный момент основными методами идентификации дикторов являются методы, основанные на следующих моделях: модели гауссовых смесей (англ. Gaussian Mixture Models, GMM), модель, основанная на методе опорных векторов (англ. support vector machine, SVM), и модель, основанная на векторном квантовании [1].

В методе, использующем гауссовы смеси, модель голоса диктора представляет собой аппроксимацию распределения используемых речевых признаков смесью гауссовых распределений. Недостатком данного метода является его вычислительная сложность, и сложность реализации [2].

Метод опорных векторов используется для разделения сложных областей в пространстве признаков речевого сигнала на два класса оптимальной (с точки зрения алгоритма) гиперплоскостью. В этом методе отыскиваются образцы, находящиеся на границах между двумя классами и через них строятся две параллельные гиперплоскости, с максимальным расстоянием между ними. После этого на равном удалении от каждой из найденных гиперплоскостей строится разделяющая гиперплоскость. Таким образом все обучающие вектора модели диктора находятся по одну сторону от разделяющей гиперплоскости. Если на вход системы подать тестовый вектор, то система будет сравнивать его положение в пространстве с положением гиперплоскости. Если тестовый вектор находится по одну сторону от гиперплоскости вместе с обучающими векторами модели диктора – то система принимает решение о том, что данный тестовый характеристический вектор принадлежит диктору.

Модель, основанная на векторном квантовании, строит модель диктора на основе кодовой книги, полученной путем векторного квантования обучающей выборки характеристических векторов диктора. Векторное квантование в общем случае осуществляется при помощи алгоритма кластеризации K-средних. Принятие решения о принадлежности тестового характеристического вектора диктору осуществляется при превышении порогового значения меры близости между входным вектором и, хотя бы одним вектором из кодовой книги.

В общем случае системы идентификации дикторов имеют структуру, представленную на рисунке 1:

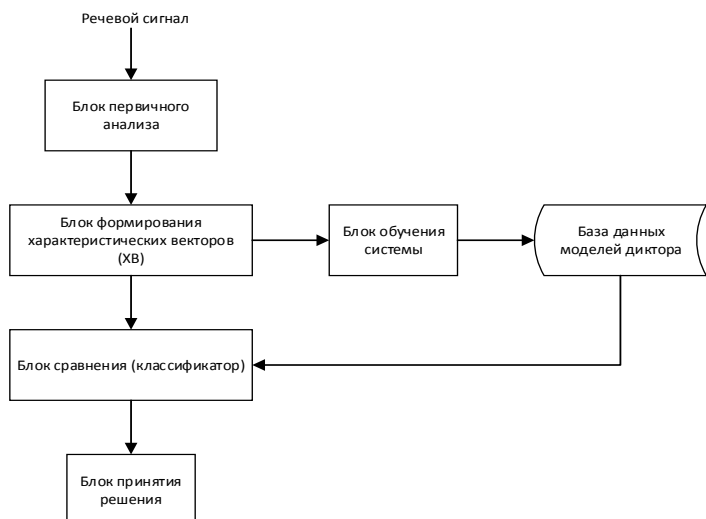


Рис. 1. Структурная схема системы идентификации диктора

Блок первичного анализа предназначен для анализа входного сигнала (разбиения речевого сигнала на фреймы) и выделения участков записи, на которых присутствует речь. Блок формирования характеристических векторов из полученных фреймов формирует мел-частотные кепстральные коэффициенты (MFCC), которые используются для обучения системы в соответствующем блоке. Блок сравнения сравнивает модели дикторов, которые хранятся в базе системы с характеристическими векторами входного сигнала. Результат сравнения передается в блок принятия решения, который сравнивает результаты работы классификатора с пороговым значением и формирует результат идентификации.

Для реализации системы идентификации диктора по образцу голоса был выбран метод, основанный на алгоритме векторного квантования.

Для построения модели диктора используется модифицированный алгоритм К-средних, который состоит из следующих шагов:

1. Ввод обучающих векторов

$$S = \{x_i \in R^d | i = 1, 2, \dots, n\}.$$

2. Определяется начальное состояние вычисляемой кодовой книги

$$C = \{c_i \in R^d | i = 1, 2, \dots, K\}.$$

3. Установить $D_0=0$ и $k = 0$.

4. Классифицировать обучающих векторов в кластеры в соответствии с правилом

$$x_i \in S_q \text{ if } \|x_i - c_q\|_p < \|x_i - c_j\|_p \text{ для } j \neq q.$$

5. Обновить кластеры $c_j, j=1, 2, \dots, K$ как

$$c_o = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i.$$

6. Установить $k = k+1$ и вычислить

$$D_k = \sum_{j=1}^K \sum_{x_i \in S_j} \|x_i - c_j\|_p.$$

7. Если $\frac{D_{k-1} - D_k}{D_k} > \varepsilon$, повторить шаги 4 – 6.

8. Вернуть текущую кодовую книгу

$$y C = \{c_i \in R^d | i = 1, 2, \dots, K\}.$$

Модификация базового алгоритма К-средних выполняется с целью уменьшения влияния выбранных начальных положений центроидов кластеров. Выбор начального состояния центроидов следует проводить по следующему алгоритму:

1. Выбрать первый центроид случайным образом (среди всех точек).
2. Для каждой точки найти значение квадрата расстояния до ближайшего центроида (из тех, которые уже выбраны) dx^2 .
3. Выбрать из этих точек следующий центроид так, чтобы вероятность выбора точки была пропорциональна вычисленному для неё квадрату расстояния. Это можно сделать следующим образом. На шаге 2 нужно параллельно с расчётом dx^2 подсчитывать сумму $Sum(dx^2)$. После накопления суммы найти значение $Rnd = random(0.0, 1.0) * Sum$. Rnd случайным образом укажет на число из интервала $[0; Sum)$, и нам остаётся только определить, какой точке это соответствует. Для этого нужно снова начать подсчитывать сумму $Sum(dx^2)$ до тех пор, пока сумма не превысит Rnd . Как только это случится, суммирование останавливается, и мы можем взять текущую точку в качестве центроида. При выборе каждого следующего центроида специально следить за тем, чтобы он не совпал с одной из уже выбранных в качестве центроидов точек, не нужно, так как вероятность

повторного выбора некоторой точки равна 0.

4. Повторять шаги 2 и 3 до тех пор, пока не будут найдены все необходимые центроиды.

Таким образом полученная кодовая книга представляет собой упрощенную модель диктора, что позволяет сократить количество используемой памяти в сравнении с моделью, состоящей из всех характеристических векторов обучающей выборки, а также позволит повысить скорость вычислений.

Процесс работы системы, реализованной по такому алгоритму сводится к полному перебору всех векторов кодовых книг и их попарному сравнению с входным характеристическим вектором. Если в ходе сравнения обнаружится, что мера близости между векторами превышает определенное пороговое значение – то системой принимается решение о принадлежности входного речевого сигнала к диктору, чья модель участвовала в сравнении.

Важной особенностью работы такой системы является возможность добавления новых моделей дикторов в уже функционирующую систему, не переобучая её. Для того, чтобы система понимала нового диктора достаточно в базу данных, хранящую модели дикторов загрузить новую которую книгу.

Рассмотренный в данной статье метод идентификации дикторов обладает целым рядом преимуществ:

- небольшое потребление памяти для хранения моделей дикторов;
- высокая скорость работы;
- простота добавления новых моделей дикторов в уже функционирующую систему.

Список использованных источников:

1. Первушин, Е. А. Обзор основных методов распознавания дикторов / Е. А. Первушин // Математические структуры и моделирование. – 2011. – №24. – С. 41–54.
2. Садыхов Р. Х., Ракуш В. В. Модели гауссовых смесей для верификации диктора по произвольной речи // Доклады БГУИР. – 2003. – №4. – С. 95–103.

Научное издание

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

МАТЕРИАЛЫ 52-Й НАУЧНОЙ КОНФЕРЕНЦИИ АСПИРАНТОВ, МАГИСТРАНТОВ И СТУДЕНТОВ

(Минск, 25-30 апреля 2016 года)

В авторской редакции

Ответственный за выпуск *Е. А. Сасин*
Компьютерная верстка *С. А. Купраш, С. В. Бирюк, М. С. Марчук*
Дизайн обложки *Н. А. Дубко*

Подписано в печать XX.XX.20XX. Формат 60x84 1/8. Бумага офсетная. Гарнитура «Arial».
Отпечатано на ризографе. Усл. печ. л. 12,78. Уч.-изд. л. 15,4. Тираж XX экз. Заказ XXX.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
2200013, Минск, П. Бровки, 6