

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

**СБОРНИК СТУДЕНЧЕСКИХ
НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ
НАУК ВГУ**

В ы п у с к 11

Под редакцией кандидата технических наук Д.Н. Борисова

Воронеж
Издательский дом ВГУ
2017

УДК 004.65 + 004.438.5

ББК Ч481(2)22

С23

Рекомендовано к печати

Ученым советом факультета компьютерных наук ВГУ

С23 **Сборник студенческих научных работ факультета компьютерных наук ВГУ / под ред. Д. Н. Борисова ; Воронежский государственный университет. – Выпуск 11. – Воронеж : Издательский дом ВГУ, 2017. – 389 с.**

ISBN 978-5-9273-2483-5

В сборник включены научные работы студентов факультета компьютерных наук ВГУ, выполненные в 2016–2017 гг. под руководством преподавателей факультета, представленные в виде докладов и рекомендованные к опубликованию оргкомитетами студенческих научных конференций.

УДК 004.65 + 004.438.5

ББК Ч481(2)22

ISBN 978-5-9273-2483-5

© Воронежский государственный университет, 2017

© Оформление. Издательский дом ВГУ, 2017

СРАВНЕНИЕ МЕТОДОВ СУПЕРПИКСЕЛЬНОЙ СЕГМЕНТАЦИИ

Е.В. Акимова

Студент

Е.Ю. Митрофанова

Доцент

Введение

Задача сегментации тесно связана с процессом распознавания и является предварительным этапом в практически любой системе обработки изображений, так как позволяет существенно упростить дальнейший анализ однородных областей изображения.

Процесс сегментации рассматривается как базовый этап построения сцены, от качества выполнения которого зависит успех решения задачи распознавания, идентификации и интерпретации объектов на изображении. Сегментация осуществляется путем разбиения изображения на непересекающиеся области – суперпиксели, другими словами, относительно однородные группы расположенных рядом пикселей объединенных на основе яркостных, цветовых, текстурных и других характеристик.

Алгоритмы сегментации цветных изображений имеют большую область применения на практике [1], например, в системах защиты – для аутентификации личности, на производстве – для индикации дефектов сборки деталей, в медицине – для первичной обработки снимков, для анализа местности по снимкам со спутников – определения площади наводнения, распознавания лесных пожаров и т.д.

На сегодняшний день известно большое количество алгоритмов сегментации изображений [2], использующих различные признаки и подходы. Существующие решения не всегда могут обеспечить достаточно качественный результат, таким образом возникает необходимость поиска новых решений с использованием базовых подходов сегментации и их комбинирования. В качестве базового подхода для дальнейших исследований был выбран алгоритм суперпиксельной сегментации SLIC.

1. Алгоритм SLIC (Simple Linear Iterative Clustering)

На текущий момент алгоритм SLIC [3] является одним из самых результативных представителей алгоритмов суперпиксельной

сегментации, он превосходит своих конкурентов по времени работы и компактности получаемых суперпикселей.

По способу реализации алгоритм SLIC является дискриминационным, т.е. суперпиксели получаются путем предварительного разбиения пикселей на кластеры заданного размера с последующим итеративным уточнением.

Алгоритм SLIC достаточно прост в реализации и разделяется на шесть основных этапов [4]:

1. Разбиение изображения на K фрагментов размера $a \times a$, которые задают начальное приближение кластеров-суперпикселей. Начальные центры – геометрические центры фрагментов C_k ;

2. Корректировка координат центров фрагментов из условия минимального значения цветового градиента в окрестности геометрического центра;

3. Формирование локальных кластеров в $2a \times 2a$ окрестности центров C_k посредством усреднения вектора интенсивностей в кластере;

4. Нахождение расстояния D между центром и точками фрагмента, как комбинации евклидовых расстояний по цветовой d_c и пространственной d_s составляющим:

$$d_c = \sqrt{(c_{j1} - c_{i1})^2 + (c_{j2} - c_{i2})^2 + (c_{j3} - c_{i3})^2}, \quad (1)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \quad (2)$$

$$D = \sqrt{d_c^2 + m^2 \times \left(\frac{d_s}{a}\right)^2}, \quad (3)$$

m – соотношение вкладов двух составляющих описания изображения в величину расстояния D ;

i и j – номера точек, между которыми вычисляется расстояние;

1. Определение новых центров кластеров и вычисление смещений центров;

2. Повтор шагов 3 – 5 до тех пор, пока смещение центров между итерациями не станет меньше заданного значения.

По своей сути SLIC является пространственно-ориентированной реализацией алгоритма кластеризации k -средних. Основное отличие от классического алгоритма k -средних заключается в ограничении области поиска на начальном этапе: пиксели для каждого сегмента ищутся не во всём изображении, а в небольшой области, пропорциональной среднему размеру сегмента, с последующей минимизацией функции ошибки кластеризации.

Однако, несмотря на хорошие общие показатели, применение базовой реализации алгоритма SLIC на практике выявило появление некоторых недостатков в процессе кластеризации отдельных пикселей, а именно возникновение тонких ответвлений и изолированных областей (рис. 1).

Данное обстоятельство хорошо наблюдается на изображениях, у которых потенциальные суперпиксели имеют примерно одинаковую яркость и цветность, но отличаются по текстуре.

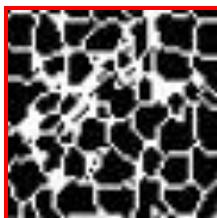


Рис. 1. Возникновение тонких ответвлений и изолированных областей

2. Модификация алгоритма SLIC

На текущий момент существует большой спектр подходов по модификации и развитию методов суперпиксельной сегментации изображений.

Рассмотрим классический алгоритм SLIC. Как уже отмечалось, при его применении в стандартном варианте в процессе кластеризации отдельных пикселей возникают тонкие ответвления и изолированные области. Целью нашего исследования является минимизация появления данных недостатков.

Один из возможных подходов к решению данной проблемы – совершенствование алгоритма SLIC путем добавления постобработки результата сегментации изображения (рис. 2).



Рис. 2. Блок-схема модифицированного алгоритма SLIC

В представляемой работе для сегментации изображений предлагается применять алгоритм SLIC с последующей трехступенчатой обработкой.

Перейдем к поэтапному описанию модифицированной части алгоритма суперпиксельной сегментации – постобработке изображения.

Первый этап, удаление тонких ответвлений, состоит из двух частей:

1. Перевод исследуемого суперпикселя в бинарный вид (где 1 – признак принадлежности пикселя к исследуемому суперпикселю);
2. Итеративный поиск и осуществление удаления ответвлений от исследуемого суперпикселя толщиной в один пиксель (рис. 3).

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Рис. 3. Удаление обособленных пикселей в бинарном представлении

Второй этап, присоединение групп внутренних пикселей к исследуемому суперпикселю, можно разделить на три части:

1. Итеративный поиск изолированных областей в блоках $a \times a$ пикселей, т.е. поиск групп единиц среди нулей в бинарном представлении суперпикселя;
2. Осуществление проверки размера обособленных областей, найденных на предыдущем этапе. Изолированной считается область, состоящая менее чем из $a+1$ пикселей;
3. Перевод изолированной группы пикселей к исследуемому суперпикселю (рис. 4).

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Рис. 4. Присоединение группы внутренних пикселей к исследуемому суперпикселю в бинарном представлении

Третий этап, присоединение групп изолированных пикселей одного кластера к соседним наиболее подходящим кластерам, является наиболее сложным и состоит из шести частей:

1. Нахождение изолированных областей в блоках $a \times a$ пикселей;
2. Вычисление площади бинарного объекта и проверка соответствия найденного размера обособленной области (менее чем $a + 1$ пикселей);
3. Нарастивание границ объекта;
4. Нахождение номеров окружающих пикселей, приращенных на предыдущем этапе;
5. Вычисление преобладающего индекса среди окружающих суперпикселей;
6. Присоединение группы пикселей к наиболее подходящему соседнему кластеру, имеющему преобладающий индекс (рис. 5-6).

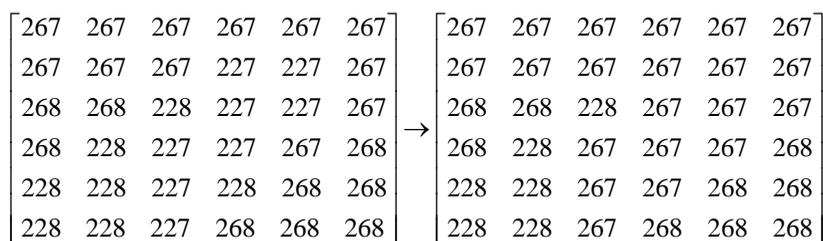


Рис. 5. Присоединение группы пикселей исследуемого суперпикселя (227) к другому суперпикселю (267) с преобладающим индексом

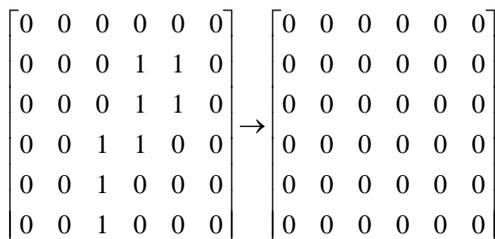


Рис. 6. Присоединение группы пикселей исследуемого суперпикселя (227) к другому суперпикселю (268) в бинарном представлении

3. Сравнение методов сегментации

С помощью высокоуровневого интерпретируемого языка программирования MATLAB был разработан алгоритм, способный выполнять все этапы предложенного решения нашей задачи.

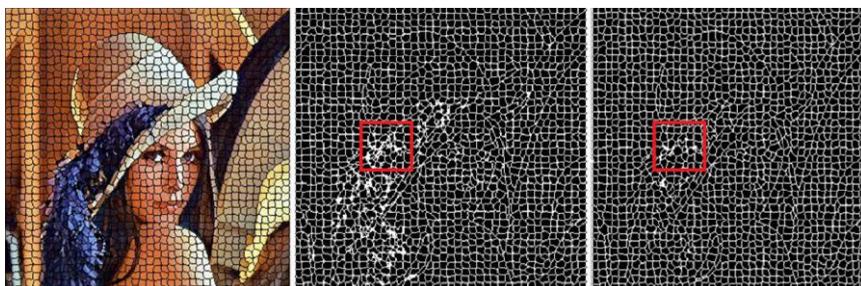
Новый модифицированный алгоритм суперпиксельной сегментации SLIC Postprocessing (SLICP), основан на базовом алгоритме SLIC с последующей трехступенчатой постобработкой каждого суперпикселя.

В качестве материала для тестирования разработанного алгоритма было выбрано одно из базовых изображений тестового набора для

исследования эффективности методов цифровой обработки изображений [5]– Lena.bmp. При выборе данного кандидата основным преимуществом был тот факт, что потенциальные суперпиксели изображения Lena.bmp имеют похожие яркостные и цветовые характеристики, но достаточно сильно отличаются по своей текстуре.

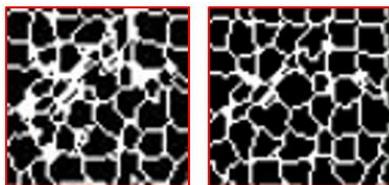
Рассмотрим результаты применения базового SLIC и модифицированного SLICP методов к выбранному ранее изображению (рис. 7). Достаточно только взглянуть на результаты сегментации, чтобы увидеть явное преимущество метода SLICP (рис. 8):

- Исчезли тонкие ответвления;
- Значительно уменьшилось количество изолированных областей;
- Общие границы суперпикселей сохранили свою форму и стали более гладкими.



слева – отображение границ областей на исходном изображении, по центру – результат сегментации SLIC, справа – результат сегментации SLICP

Рис. 7. Пример суперпиксельной сегментации изображения Lena.bmp



слева – алгоритм SLIC, справа – алгоритм SLICP

Рис. 8. Фрагмент сегментированного изображения Lena.bmp

В качестве мер для количественной оценки эффективности предложенного решения рассматривались ошибки первого (отношение неправильно выделенных граничных пикселей к общему числу пикселей)

и второго (отношение правильно выделенных граничных пикселей к общему числу граничных пикселей) рода [6]. Оба критерия показали незначительное преимущество модифицированного алгоритма.

Заключение

Рассмотрена задача совершенствования существующих методов сегментации. В качестве базового подхода для исследований был выбран алгоритм суперпиксельной сегментации SLIC.

В соответствии с поставленной задачей разработана программа, реализованная в системе компьютерной математики MATLAB. В качестве решения для осуществления модификации алгоритма SLIC была предложена трехшаговая процедура постобработки.

Показано, что предложенный способ сегментации дает явное уменьшение тонких ответвлений и сокращение изолированных областей в процессе кластеризации отдельных пикселей.

Дальнейшие исследования будут направлены на более глубокое изучение метода и его сравнение с другими подходами повышения качества сегментации.

Список литературы

1. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман; пер. с англ. – М. : БИНОМ; Лаборатория знаний, 2006. – 752 с.
2. Саввин, С. В. Методы суперпиксельной сегментации и их применение для анализа изображения с разнородной текстурой / С. В. Саввин, А. А. Сирота // Вестник ВГУ, серия : системный анализ и информационные технологии. – Воронеж, 2016. – № 4. – С. 165-173.
3. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods / R. Achanta [и др.] // IEEE transactions on pattern analysis and machine intelligence. – 2012. – Т. 34. – № 11. – С. 2247-2281.
4. Мурашов, Д. М. Применение теоретико-информационного подхода для сегментации изображений / Д. М. Мурашов // Machine Learning and Data Analysis. – 2016. – № 2. – С. 231-243.
5. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
6. Некоторые методы сравнительного исследования детекторов границ / А. В. Захаров [и др.] // Труды НИИСИ РАН. – 2012. – № 1. – С. 4-13.

СИСТЕМА УЧЕТА ПОСЕЩАЕМОСТИ НА ОСНОВЕ RFID

Р.П. Аксенов

Студент

А.С. Коваль

Старший преподаватель

Введение

Во всех учебных заведениях отмечают посещение занятий студентами. В зависимости от принятой процедуры, это может отнимать часть времени занятия, либо, например, при самостоятельной записи, иметь низкую достоверность. В качестве иллюстрации проблемы, достаточно представить себе, как будет проходить стандартная переключка на лекционном занятии, где присутствует полный курс факультета. Целью данной работы является разработка и внедрение системы учета посещаемости, которая при приемлемом уровне достоверности, не требовала бы никакого дополнительного времени и усилий от преподавателя.

У каждого из студентов нашего ВУЗа есть специальный электронный пропуск, содержащий RFID метку, по которому осуществляется вход в здание университета. Эта RFID-метка всегда находится при студенте, поддерживается службой охраны (восстанавливается, приобретается) и однозначно соответствует студенту. Таким образом, затраты на идентификатор студента в системе учета посещаемости – отсутствуют. Сокращение затрат на хранение и обработку данных посещаемости можно достичь использованием уже существующего в ВГУ "образовательного портала "Электронный университет ВГУ", построенного на ПО Moodle, который является частью информационной системы поддержки учебного процесса и работает с учебными планами, курсами и базой контингента.

Электронные системы учета посещаемости используются на некоторых предприятиях для учета рабочего времени сотрудников. Разрабатываемая система учета предназначена для автоматизации учета посещаемости занятий студентами в учебных заведениях, в частности, в ВГУ.

1. RFID (Radio Frequency Identification)

RFID – способ автоматической идентификации объектов, в котором посредством радиосигналов считываются или записываются, хранящиеся в RFID-метках данные [1]. Полная система состоит из таких компонентов, как метки и считыватели. Метки (tag) RFID – устройства, способные

хранить и передавать данные. В памяти меток содержится их уникальный ID (идентификатор). Считыватели (reader) – приборы, которые читают информацию с меток и/или записывают в них данные. Эти устройства могут быть как постоянно подключенными к учетной системе, так и работать автономно.

RFID-метка представляет собой миниатюрное запоминающее устройство. Она состоит из чипа, где хранится информация о карте, и антенны, с помощью которой метка передает информацию о карте. По источнику питания выделяют 2 типа меток: пассивные и активные. Пассивные RFID-метки не имеют встроенного источника энергии. Электрический ток, индуцированный в антенне электромагнитным сигналом от считывателя, обеспечивает достаточную мощность для функционирования чипа.

Активные RFID-метки обладают собственным источником питания и не зависят от энергии считывателя, вследствие чего они читаются на дальнем расстоянии, имеют большие размеры. Однако, такие метки наиболее дороги, а у батарей ограничено время работы. Так же метки делятся по рабочей частоте на:

- Метки диапазона LF (125 – 134кГц);
- Метки диапазона HF (13,56 МГц);
- Метки диапазона UHF (860 – 960 МГц).

Для данной работы был выбран первый тип меток, так как его функционала достаточно для требуемой задачи, а так же имеют низкую цену.

2. Moodle

Moodle (англ. Modular Object-Oriented Dynamic Learning Environment) – система управления курсами, также известная, как система управления обучением. Представляет собой свободное веб-приложение, предоставляющее возможность веб-обучения. Moodle устанавливается на веб-сервер, пользователи получают доступ к системе по сети через браузер. Для работы с Moodle подойдет любой современный компьютер, имеющий выход в сеть Интернет[2].

Структурная схема системы дистанционного обучения Moodle представлена на рис.1.

Система предоставляет возможность инсталляции образовательных ресурсов и обеспечивает средствами доступа к ресурсам и управления ими. Система также обеспечивает коммуникационное взаимодействие участников образовательного процесса.

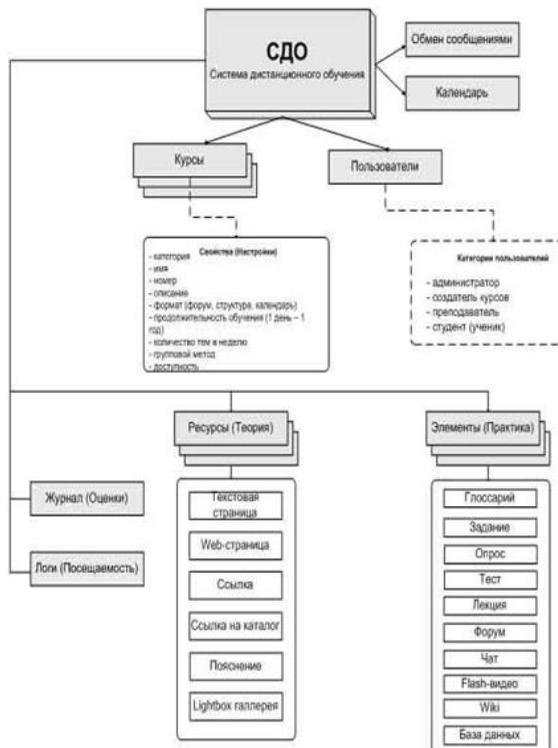


Рис. 1. Структурная схема СДО Moodle

3. Постановка задачи

Спроектировать считывающее устройство; создать базу данных с информацией о студентах, которые пришли на занятие; создать программу для обработки информации о посещениях с целью хранения и дальнейшей передачи полученной информации на сервис Moodle; создать процесс передачи полученных данных на сервис Moodle с целью автоматического отмечания посещаемости студентами занятий в университете. В дальнейшем для реализации поставленных задач потребуется установка считывателей, которые будут осуществлять передачу данных на сервер, на входах в учебные аудитории университета.

4. Используемые технологии

Аппаратные средства:

- Плата – Arduino Leonardo R3;
- Считывающее устройство с антенной;

Программные средства:

- Среда разработки – Arduino для написания скетчей, Visual Studio 2013 для локального хранения данных;
- БД – MySQL;
- Языки – C#, C, SQL.

5. Построение проекта системы

После определения выполняемых функций, было решено приступить к созданию аппаратного средства для считывания информации с электронных карт студентов. Был создан макет работы считывателя RFID-меток на основе платы Arduino Leonardo R3. Затем было спроектировано несколько сценариев работы данной электронной системы:

- Сбор информации на входе в учебную аудиторию. Установка считывателя у двери аудитории, т.е. студент обязан прикладывать свою электронную карту, чтобы отметить свою посещаемость на занятии;
- Установка более мощной антенны считывателя, например в дверной проем, чтобы студенту не требовалось доставать и самостоятельно применять личную электронную карту;
- Установка считывателя, способного собирать информацию с личных карт во время проведения учебных занятий в аудитории.

Предложенные выше сценарии имеют как свои преимущества, так и недостатки. На данном этапе ведется разработка 1-го вида сценария работы данной системы.

Структурная схема электронной системы учета посещаемости показана на рис. 2. Диаграмма последовательности системы представлена на рис. 3.

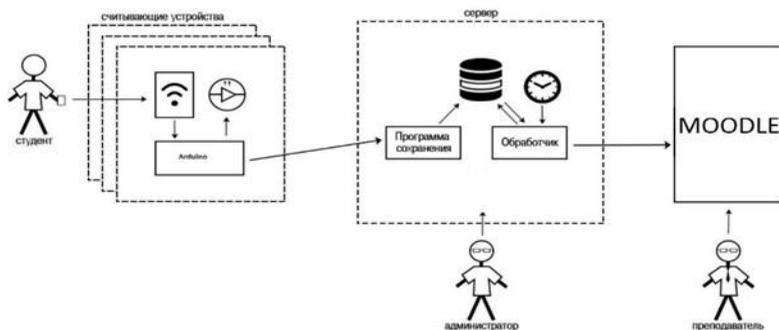


Рис. 2. Схема передачи данных в системе

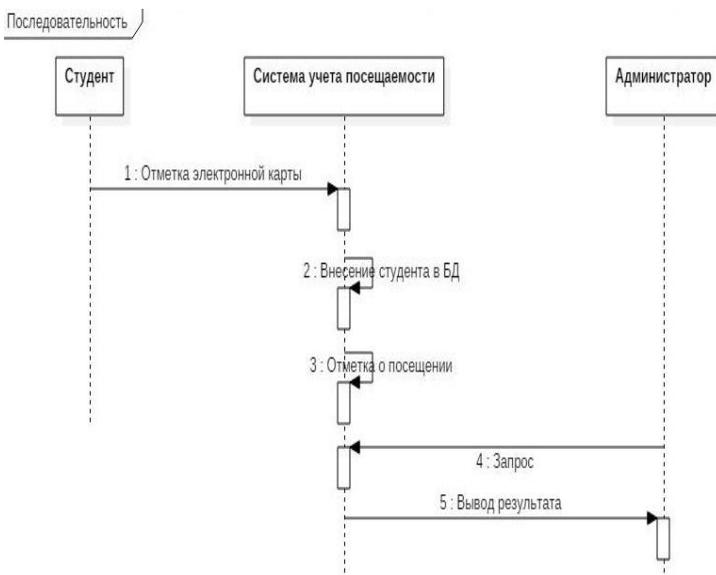


Рис. 3. Диаграмма последовательности системы

6. Описание разработки системы

После сборки считывателя была произведена программная настройка для корректной его работы.

Затем ведется заполнение базы данных, содержащей информацию о картах, с целью дальнейшего отmarkания посещаемости студентами занятий по этой информации. Основной этап – передача данных в систему Moodle. Данная система имеет огромное количество плагинов, облегчающих ведение электронных курсов. Существует и плагин «Посещаемость», но он не включает в себя функционал для автоматического учета посещаемости. На данном этапе ведется разработка плагина, способного забирать информацию с локального приложения и передавать её на сервис Moodle.

Локальное приложение представляет собой программу, написанную на языке программирования C#, которая способна получать данные с COM – порта и записывать их в файл. Здесь используется простой текстовый файл формата CSV (Comma-Separated Values — значения, разделённые запятыми), так как с ним проще работать. Для удобства отображения можно использовать стандартные табличные файлы типа Excel.

Система Moodle предоставляет огромный выбор API для разработки или доработки собственных решений. Для разработки собственного плагина, который автоматически отmarkает посещаемость в системе

Moodle, были использованы следующие API ,помогающие реализовать конкретно поставленную задачу:

- Data manipulation API – для работы с базами данных;
- Form API – для работ с формами (все формы базируются на основе moodleform);
- File API – непосредственно для работы с файлами.

Разработана система для локального хранения данных о присутствующих студентах. Решение о создании такой системы было принято после того, как в процессе работы несколько раз пришлось столкнуться с тем, что сервис Moodle не был работоспособен.

7. Тестирование стенда

На данном этапе считыватель менее чем за 1 секунду способен записывать данные с электронной карты студента, что говорит нам о том, что при внедрении 1 сценария данной системы, не возникнет проблем с очередью на входе в аудиторию. Кроме того были выполнены и другие виды тестов использования собранного считывателя: тестирование расстояния срабатывания считывания RFID-карты, возможность считывателя получать информацию с электронной карты студента, которая находится за преградой (обложка студенческого билета, кошелек, сумка), так как студенты часто не используют их электронную карту, извлекая её из своих вещей. Проведенные тесты показали, что стенд, собранный в настоящий момент времени способен производить считывание RFID – карт на расстоянии не более 4 см. Так же было найдено, что устройство может получать информацию с карты студента через плотную ткань сумки или кошелька, а также через обложку студенческого билета. Но в этих случаях расстояние считывания заметно сокращается. Были проведены тесты считывания с написанной программой, которая заносит данные с COM-порта в текстовый файл для дальнейшей отправки их в Moodle. В таблице «Результаты тестирования считывателя и программы» показаны полученные результаты после проведения опытов, демонстрирующих наиболее вероятные сценарии работы системы.

Таблица

Результаты тестирования считывателя и программы

Номер опыта	1	2	3
Препятствие	Отсутствует	Студенческий	Кошелек/сумка
Расстояние, см	<=4	<=2	0
Время записи, мс	500	500	500

Заключение

Анализ разработанной системы показал, что она работоспособна в сценариях, описанных в постановке задачи и способна, таким образом, экономить время преподавателя на учебных занятиях. В дальнейшем возможно улучшение данной системы учета посещаемости в направлении увеличения расстояния устойчивого приема RFID карт и, таким образом, сокращения времени на регистрацию студентов в начале занятия.

Данная статья посвящена разработке системы контроля посещаемости на основе технологии RFID, а также возможности её применения в учебных заведениях. Описано создание данного продукта, проанализированы различные варианты использования. Решены проблемы, с которыми можно было столкнуться после внедрения данной системы. Использование данной системы учета посещаемости занятий студентами университета позволит в дальнейшем сократить временные затраты на стандартный опрос присутствующих на лекции.

Список литературы

1. Документация о технологии RFID от компании-производителя RFID-меток «РСТ-Инвент» [Электронный ресурс] : база данных. – Режим доступа : <http://www.rst-invent.ru/about/technology/>
2. Документация сервиса Moodle [Электронный ресурс] : база данных. – Режим доступа : <https://moodle.org/?lang=ru>

РЕАЛИЗАЦИЯ МОДУЛЯ РАСЧЕТА СТИПЕНДИЙ ДЛЯ УЧЕБНЫХ ЗАВЕДЕНИЙ НА ПЛАТФОРМЕ 1С

К.Г. Антипов

Студент

А.И. Другалев

Ассистент

Введение

В условиях развития рыночной экономики все большее число учебных заведений осознают преимущества использования в своей деятельности информационных систем. Информационные системы предлагают информацию в понятном, удобном для пользователя виде, поэтому их разработка и внедрение является важным звеном в развитии информационных технологий.

В наше время, трудно представить себе, что управление таким громоздким процессом, как расчет стипендий студентов даже в сравнительно небольшом учебном заведении, осуществляется вручную или на морально и технически устаревшей технике, практически невозможно.

Поэтому, проблема создания автоматизированной системы для расчета стипендий в учебных заведениях весьма актуальна, поскольку объем обрабатываемой информации значителен, и вручную справляться с таким количеством данных не только очень сложно и трудоемко, но и экономически невыгодно.

В целях повышения эффективности функционирования, учебные заведения сегодня активно используют информационные технологии, позволяющие автоматизировать их деятельность путем использования информационных систем.

Многие учебные заведения уже осознали необходимость внедрения автоматизированных средств для расчета стипендий и некоторые из них даже приняли попытки самостоятельной разработки и внедрения подобных систем. Решение обозначенной задачи не такое простое, каким может показаться на первый взгляд – сложность расчета стипендий студентов заключается в учете наличия различных видов расчета, разных вариантов выплат, различных вариантов налогообложений.

В данной дипломной работе предлагается модуль реализующий функционал по расчету стипендий студентов. Программный продукт предоставляет информацию о студентах, назначенных и произведённых им выплатах. На основе представленной системы можно анализировать информацию о выплатах, делать соответствующие выводы.

1. 1С: Предприятие 8

Повышение результатов деятельности, уменьшение ошибок, вследствие влияния человеческого фактора, автоматизация рутинной работы являются видными преимуществами внедрения автоматизированных систем в различных сферах, учебные заведения не являются исключением. Говоря о программных продуктах, позволяющих автоматизировать деятельность, нельзя не говорить об отечественном комплексе 1С: Предприятие, которое занимает первое место на российском рынке.

1С: Предприятие – программный продукт компании «1С», предназначенный для автоматизации деятельности на предприятии.

Первоначально «1С: Предприятие» было предназначено для автоматизации бухгалтерского и управленческого учётов (включая начисление зарплаты и управление кадрами), но сегодня этот продукт

находит своё применение в областях, далёких от собственно бухгалтерских задач [1].

Система программ 1С: Предприятие 8 [2] включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц [3]. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу [4, 5].

2. Требования к функционалу

Целью данной работы является разработка самостоятельного модуля, который позволяет автоматизировать процесс расчета стипендий в учебных учреждениях, и обладающего следующей функциональностью:

- иметь возможность ведения контингента студентов;
- иметь возможность назначения и расчета различных видов начислений;
- иметь возможность предоставлять актуальную отчетность;
- иметь возможность осуществлять депонирование выплат;
- иметь возможность отражать выплату стипендий, как через кассу, так и через банк.

Более детально и наглядно со стороны пользователя, отразить требования можно посредством использования диаграммы вариантов использования (рис. 1).



Рис. 1. Диаграмма вариантов использования

3. Принцип расчета и выплаты стипендий

После проведения анализа предметной области, аналогов и определения списка требований к продукту, можно начинать разрабатывать логическую структуру модуля. В результате была построена схема, показывающая процесс расчета стипендий (рис. 2).

4. Реализация

Реализация логики работы модуля проводилась в соответствии с ранее разработанной схемой расчета стипендий (рис. 2).

Для ввода информации о назначении/удержании стипендии создан документ «Приказ». В системе существуют следующие типы приказов:

– назначение стипендии – начисления по данному приказу осуществляются ежемесячно в сумме, указанной в графе «Сумма (в месяц)» с помощью документа «Начисление стипендии»;

– разовые выплаты – начисления по данному приказу осуществляются при вводе данного приказа в сумме, указанной в графе «Сумма».

Расчет стипендии осуществляется с помощью документа «Начисление стипендии». Данный документ вводится ежемесячно.

Выплата стипендии может осуществляться как через кассу, так и с расчетного счета организации, открытого в банке.

Для выплаты стипендии через кассу необходимо создать документ «Ведомость на выплату через кассу». Из данного документа можно распечатать платежную ведомость. На основании данной ведомости вводится документ «Расходный кассовый ордер» с видом операции «Выплата стипендии», который отражает фактическое выбытие денежных средств из кассы учреждения.

В случае если стипендия была не выплачена, то на основании «Ведомости на выплату через кассу» создается документ «Депонирование стипендии». В данном документе вводится список обучающихся, не получивших стипендию, которая затем будет перечислена на специальный счет до востребования.

Для выплаты стипендии через банк предназначен документ «Ведомость на выплату через банк». Для выплаты необходимо выбрать зарплатный проект (договор с банком). Из данного документа можно распечатать список перечисляемой в банк стипендии. На основании данной ведомости вводится документ «Списание безналичных денежных средств» с видом операции «Выплата стипендии», который отражает фактическое выбытие денежных средств с лицевого счета учреждения. Также ведомость можно выгрузить в банк в формате определенным банком.

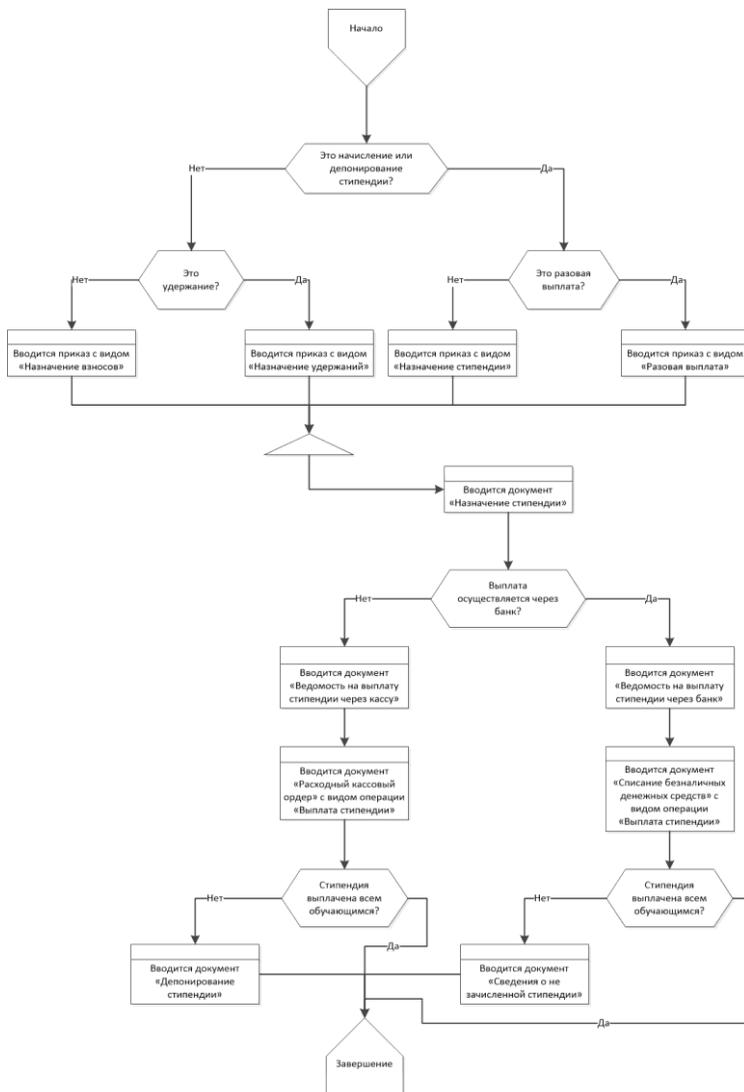


Рис. 2. Общая схема расчета и выплаты стипендий

Для выплаты депонированной стипендии предназначен документ «Выплата депонированной стипендии».

Для списания депонированной стипендии предназначен документ «Списание депонированной стипендии». При заполнении автоматически

табличная часть заполнится обучающимися, которым была не выплачена стипендия. На основании документа можно распечатать справку о списании депонированной стипендии.

На основании данных о начисленной стипендии можно будет сформировать следующие отчеты:

- ведомость начислений и удержаний;
- свод по начисленной и не выплаченной стипендии;
- справка о начисленной стипендии.

5. Интерфейс модуля

Поскольку модуль создавался для платформы 1С: Предприятие 8, то и стиль интерфейса регламентируется соответственно самой платформой [6].

В качестве примера рассмотрим реализацию интерфейса документа «Начисление стипендии» (рис. 3).

Рис. 3. Интерфейс документа «Начисление стипендии»

Данный документ состоит из следующих частей:

– шапка документа, в которой указывается месяц начисления, дата и номер документа (заполняются автоматически), группа студентов (если не указана заполнение будет происходить по всем группам), кнопка «Заполнить» при нажатии на данную кнопку будет происходить заполнение документа на основании введенных приказов о назначении/удержании стипендии;

– в табличной части «Начисления» отображаются все начисления, рассчитанные в данном месяце, с указанием вида начисления, периода, суммы к выплате, а также за счет каких средств будет происходить выплата;

- в табличной части «Удержания» отображаются все удержания, рассчитанные в данном месяце, с указанием вида удержания, периода, суммы удержания, а также статья, по которой будет проходить данное удержание;
- в табличной части «НДФЛ» отражаются суммы налога на доходы физических лиц, удержанные в данном месяце;
- в табличной части «Взносы» отражается наименование и сумма взносов.

Заключение

Данная статья посвящена разработке модуля расчета стипендий для учебных заведений на базе платформы 1С: Предприятие 8. В итоге было разработано приложение, предоставляющее следующий функционал:

- имеет возможность ведения контингента студентов;
- имеет возможность назначения и расчета различных видов начислений;
- имеет возможность предоставлять актуальную отчетность;
- имеет возможность осуществлять депонирование выплат;
- имеет возможность отражать выплату стипендий, как через кассу, так и через банк.

С помощью разработанного модуля, пользователь может вести автоматизированный учет стипендий в учебном заведении.

Список литературы

1. Википедия [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/1С:Предприятие>
2. 1С: Предприятие 8 [Электронный ресурс]. – Режим доступа : <http://v8.1c.ru/overview/>
3. Харитонов С.А. Бухгалтерский и налоговый учет в 1С: Бухгалтерии 8 / С.А. Харитонов. – М. : ООО «1С-Пабблишинг», 2014. – 735 с.
4. Михайлов С.Е. 1С – программирование как дважды два / С.Е. Михайлов. – М. : ООО «1С-Пабблишинг», 2005. – 398 с.
5. Митичкин С.А. Практика программирования в среде 1С: Предприятие 8 / С.А. Митичкин. – М. : «КомБук», 2004. – 272 с.
6. Радченко М.Г. Предприятие 8.3 практическое пособие разработчика / М.Г. Радченко, Е.Ю. Хрусталева. – М. : ООО «1С-Пабблишинг», 2013. – 964 с.

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА СО SPRING CLOUD

И.С. Баранников

Студент

А.И. Другалев

Ассистент

Введение

Микросервисная архитектура – архитектура приложения в виде набора независимо развертываемых сервисов [1]. При использовании данного подхода единое приложение строится из небольших сервисов, взаимодействующих между собой.

Для более глубокого рассмотрения микросервисного приложения его стоит сравнить с монолитом – приложением, построенным как единое целое. Для приложения, построенного в виде микросервисов, характерны такие преимущества как:

- Частичное развертывание – возможность обновления частей приложения, в то время как монолитное приложение придется заново развертывать целиком;

- Мультиплатформенность – возможность использовать разные языки программирования и технологии благодаря распределению модулей по разным серверным узлам;

- Сохранение модульности и независимая масштабируемость;

- Децентрализованное хранение (каждый сервис имеет свою базу);

В то же время у монолитного подхода есть свои преимущества:

- Простота – монолитная архитектура проще в реализации и управлении;

- Рефакторинг модулей – в монолите легче проводить изменения между модулями, т. к. микросервисная архитектура подразумевает четкое установление границ модулей (но данную особенность в то же время справедливо будет отнести и к преимуществам микросервисного подхода в архитектуре);

В микросервисном приложении существует 2 типа модулей:

- Функциональные – отвечающие за определенную бизнес-логику;

- Инфраструктурные – обеспечивают совместную работу функциональных сервисов;

Ниже рассмотрим некоторые инфраструктурные сервисы, которые реализованы в Spring Cloud [2].

1. Spring Boot

Spring Boot представляет собой инструмент для быстрого и удобного создания каркаса приложения с необходимым набором зависимостей. Spring Boot включает в себя инструменты командной строки, а также веб-конструктор, чтобы можно было быстро создать прототип приложения, импортировать его себе в IDE и модифицировать под свои нужды. Со Spring Boot возможно создание полноценных Spring-приложений с минимальным количеством конфигурационных файлов. Кроме того, Spring Boot имеет встроенный сервер Jetty, что так же ускоряет процесс разработки и тестирования приложения. А также сервисы Netflix OSS интегрированы со Spring Boot посредством Spring Cloud Netflix [3], что позволяет с помощью нескольких аннотаций включить и использовать в своем приложении проверенные временем компоненты Netflix такие как: API Gateway(Zuul), Service Discovery(Eureka), Circuit Breaker(Hystrix) и т.д.

2. Spring Cloud Config

Spring Cloud Config Server представляет собой хранилище конфигураций для распределенной системы [4]. В качестве источника конфигураций могут поддерживаться как системы контроля версий, так и локальные файлы. Spring Cloud Config легко встраивается в Spring Boot приложение посредством добавления соответствующей зависимости и аннотации `@EnableConfigServer`. По умолчанию он запускается на порту 8080, как и любое Spring Boot приложение, но его можно изменить в `application.properties`, здесь же указывается и репозиторий с конфигурационными `yaml` и `property` файлами.

Для клиентского приложения(микросервиса) достаточно будет указать в конфигурационном файле (`application.properties`) название сервиса и адрес `config`-сервера. И когда клиентское приложение запустится, оно заберет конфигурацию с `config`-сервера.

3. API Gateway (Zuul)

API Gateway – единая точка входа в приложение, она принимает внешние запросы и направляет их на нужные сервисы.

У Netflix существует реализация под названием Zuul, которая интегрирована со Spring Cloud. Включить Zuul в Spring Boot приложении можно с помощью аннотации `@EnableZuulProxy`.

В самом простом виде в конфигурации указывается `uri` запроса и название сервиса, на который его нужно отправить. Например:

```
zuul:
  routes:
    example-microservice-name:
      path: /example-uri/**
      serviceId: test_service
```

В данном случае все запросы, url которых начинается с example-url будут направлены на сервис с id = test_service. Стоит заметить, что в качестве расположения соответствующего сервиса указывается либо serviceId (когда используется Service Discovery), либо url для указания физического нахождения микросервиса.

4. Service Discovery(Eureka)

Service discovery является одним из ключевых паттернов микросервисной архитектуры. Он позволяет автоматически определять адреса для экземпляров сервисов.

Компонент Netflix Eureka интегрирован со Spring, поэтому добавив соответствующую зависимость и аннотацию @EnableEurekaServer можно превратить Spring Boot приложение в Registry server.

В клиентском приложении (микросервисе) нужно будет добавить аннотацию @EnableEurekaClient и указать расположение Eureka сервера, например:

```
eureka:
  client:
    serviceUrl:
      defaultZone: http://localhost:8761/eureka/
```

Имя клиентского сервиса(serviceId) берется из \${spring.application.name}, поэтому его необходимо задать. Далее при старте приложения оно будет зарегистрировано в Eureka с соответствующим именем. Вся необходимую информацию о сервисах можно посмотреть на дашборде.

5. Circuite Breaker(Hystrix)

Hystrix является имплементацией паттерна предохранитель. Он предназначен для контроля задержек и ошибок в распределенной системе. Для предотвращения каскадных отказов, причиной которых является сбой на более низком уровне и которые могут дойти вплоть до пользователя, используются fallback-методы, которые будут вызваны при неуспешном обращении (Рисунок).

Для подключения Hystrix к своему клиентскому приложению(сервису) нужно добавить соответствующую зависимость и аннотацию @EnableCircuitBreaker. А аннотацию @HystrixCommand(fallbackMethod = "doSomething") повесить на методы, в которых может произойти ошибка. Параметр fallbackMethod служит для указания названия метода, который будет выполнен в случае возникновения ошибки.

Большим достоинством Hystrix является наличие дашборда, на котором отображаются метрики, собранные по каждой HystrixCommand.

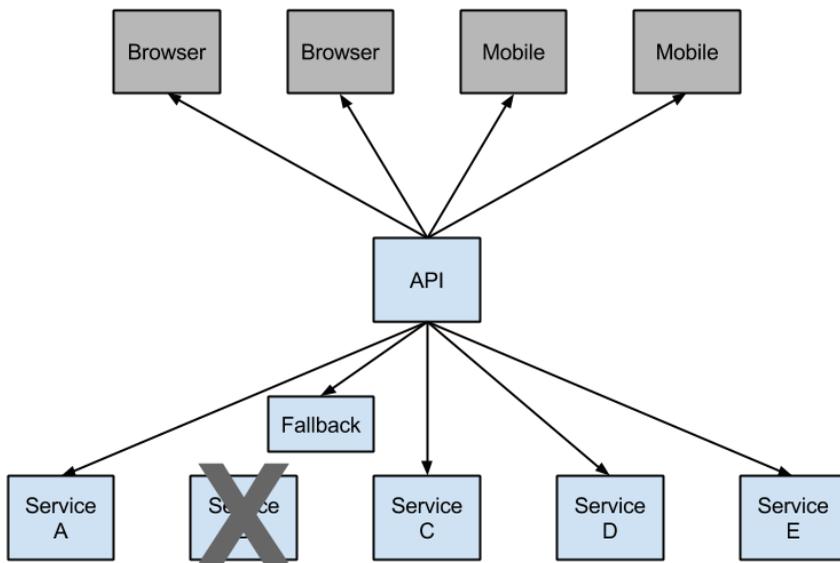


Рисунок. Предотвращение каскадного отказа

Заключение

Данная статья посвящена микросервисной архитектуре Enterprise-приложений, основным ее идеям и сервисам, с помощью которых можно организовать инфраструктуру своего приложения при выборе данного подхода. Микросервисный подход со своими преимуществами будет полезен в крупных приложениях, в которых можно выделить десятки/сотни функциональных сервисов с четкими границами. А огромное количество инструментов/сервисов, интегрированных со Spring Cloud, сделает процесс создания инфраструктуры достаточно простым, как было показано в данной статье. В то же время важно понимать, что большинству проектов (особенно небольших) микросервисная архитектура не нужна [5], достаточно продуманной структуры и четкого отображения ее в коде. К тому же в будущем хорошо структурированные и организованные модули можно без больших усилий перевести на микросервисы.

Список литературы

1. Архитектура микросервисов [Электронный ресурс] : статья. – Режим доступа : <https://habrahabr.ru/company/mailru/blog/320962/#2>

2. Микросервисная архитектура, Spring Cloud и Docker [Электронный ресурс]: статья. – Режим доступа : <https://habrahabr.ru/post/280786/>

3. Spring Cloud Netflix [Электронный ресурс]: документация. – Режим доступа : <http://cloud.spring.io/spring-cloud-netflix/spring-cloud-netflix.html>

4. Spring Cloud Config [Электронный ресурс]: документация. – Режим доступа : http://cloud.spring.io/spring-cloud-config/spring-cloud-config.html#_client_side_usage

5. Преимущества и недостатки микросервисной архитектуры [Электронный ресурс]: статья. – Режим доступа : <http://eax.me/micro-service-architecture/>

ГЕОПОЗИЦИОНИРОВАНИЕ СОТРУДНИКОВ ПОСРЕДСТВОМ МОБИЛЬНОГО ПРИЛОЖЕНИЯ 1С НА ПЛАТФОРМЕ ANDROID

М.А. Безруков

Студент

А.И. Другалев

Ассистент

Введение

В наши дни, в эпоху высоких технологий и постоянного технологического прогресса, одним из невосполнимых ресурсов является время. В предпринимательской деятельности каждая упущенная минута и не вовремя принятое решение прямо пропорционально упущенной прибыли. Предприниматели сталкиваются с такой проблемой, как нехватка времени и иррациональное его использование. Например, владельцу предприятия нужно ехать в офис, либо звонить секретарю просто для того, чтобы узнать о финансовом положении фирмы, ее задолженностях. Или нужно контролировать процесс производства на предприятии. Но, пожалуй, самое главное – контролировать работу сотрудников своей организации.

Речь идет о том, что большинство сотрудников, как правило, неполноценно осуществляют свою деятельность в рабочее время, не с полной отдачей. Раньше уходят на перерыв, позже с него возвращаются. И это еще не говоря о, так называемом, человеческом факторе.

Курьерская служба, служба такси, логистика – руководителям организаций, специализирующихся по всем этим видам деятельности необходимо каким-либо образом контролировать работу своих сотрудников во избежание недобросовестного исполнения своих обязанностей и, как следствие, уменьшения прибыли.

В связи с этим было принято решение о создании информационной системы, которая позволит устранить проблему иррационального использования сотрудниками времени в предпринимательской деятельности.

Актуальность решения данной проблемы обусловлена тем, что разработанная система предоставит пользователю возможность оперативного получения интересующей его информации удаленно, имея под рукой лишь смартфон с приложением IC [1, 2] и функцией доступа в сеть Интернет. Сам пользователь сможет в этот момент находиться в любой точке земного шара, в самолете или автомобиле, поскольку система не будет требовать наличия выгруженной базы данных IC, а все данные будут периодически передаваться через веб-сервис в виде сообщения с координатами местонахождения устройства сотрудника. После получения данных пользователь сможет отразить их на карте и построить маршрут с целью выявления отклонений и нарушений.

1. Цель работы

Цель работы в данной статье – реализация информационной системы для обеспечения удаленного доступа к информации о местонахождении сотрудников посредством мобильного приложения с включенным GPS, периодически посылающем координаты, после преобразования которых можно будет построить маршрут для анализа.

Таким образом, предприниматель сможет экономить своё время, получая при этом достоверную и актуальную на текущий момент информацию о геопозиционировании сотрудников, введя при этом какие-либо штрафные санкции в случае выявления нарушений. Сотрудник при этом перестанет отвлекаться на посторонние вещи и станет использовать сэкономленное время для выполнения других задач и, как следствие, получения дополнительной прибыли.

В процессе анализа разрабатываемой информационной системы было выявлено, что она должна иметь следующие варианты использования (рис. 1):



Рис. 1. Диаграмма use-case

2. Реализация

Для реализации требуемой системы необходимо разработать алгоритм обмена данными между приложением 1С на мобильном устройстве и десктопной частью информационной системы.

Десктопные приложения – это программы, логика работы которых требует наличия оператора (человека работающего с программой), содержащие в себе всю полную функциональность и способные работать отдельно на любой машине изолированно от других приложений.

Для осуществления этого алгоритма можно использовать «пустую» конфигурацию 1С, в которой в свойствах центральной базы в пункте «Назначение использования» нужно указать и «Мобильное устройство», и «Персональный компьютер» для того, чтобы все инструменты при разработке и настройке были доступны.

Также для реализации обмена данными понадобятся следующие инструменты:

- web-сервисы;
- стандартные планы обмена;
- XDTO-пакеты.

Web-сервисы – один из механизмов платформы, используемых для интеграции с другими информационными системами. Он является средством поддержки SOA [3] (англ. Service-Oriented Architecture) – сервис-ориентированной архитектуры, которая является современным стандартом интеграции приложений и информационных систем.

В web-сервисе «WebОбмен» создадим операцию «ВыполнитьОбмен», в которую добавим параметры «КодУстройства», необходимый для идентификации устройства, с которым будет проходить обмен, и «ДанныеУстройства», необходимый для хранения координат

местоположения сотрудника, которому принадлежит устройство и другую информацию, требуемую для контроля добросовестного выполнения обязанностей.

Web-сервисы в данном случае выступают как точка обмена информацией. Именно при вызове web-сервиса с мобильного приложения будет запускаться web-сервис и далее будет осуществлен обмен.

Планы обмена содержат информацию об узлах, которые могут участвовать в обмене данными, определяют состав данных, которыми будет производиться обмен. В одном прикладном решении может существовать несколько планов обмена, каждый из которых может описывать свой порядок обмена данными.

В частности, при написании данного приложения планы обмена нужны, чтобы определить объекты метаданных, которые должны принимать участие в обмене, чтобы корректно передавать информацию о сотруднике и его местоположении.

XDTO-пакеты позволяют описать в конфигурации систему типов и значений для взаимодействия с различными внешними источниками данных, например с web-сервисами. В конфигурацию может быть добавлено произвольное количество XDTO-пакетов.

В свойствах XDTO-пакета «WebОбмен» в поле URI пространства имен укажем адрес web-сервиса, в данном случае «<http://localhost/ws1>».

Таким образом, план обмена формирует пакеты, и посредством web-сервиса будет производиться обмен этими пакетами с мобильным приложением.

Для операции «ВыполнитьОбмен» создадим одноименную процедуру, в которой реализуем получение и выгрузку данных в мобильное устройство.

Далее необходимо создать объект IC «Общая команда».

Общие команды – объекты конфигурации, с помощью которых разработчик может описать действия, предназначенные для выполнения пользователем.

В общей команде «Обмен» создадим процедуру «ОбработкаКоманды», с помощью которой происходит обмен данных с центральной базой и обработка полученных данных.

После выполнения перечисленных выше действий будет реализован механизм двустороннего обмена данными между мобильным приложением и центральной базой.

Чтобы осуществить первый обмен, необходимо использовать так называемые push-оповещения [4].

С помощью push-уведомлений можно реализовывать сценарии взаимодействия основного и мобильного приложений по инициативе

основного приложения. Когда в основном приложении возникает некоторое событие, приложение рассылает push-уведомления получателям – мобильным приложениям.

Это оповещение необходимо использовать для того, чтобы активировать мобильное приложение, если оно ещё не было запущено.

Например, предприниматель хочет узнать местоположение конкретного сотрудника, но на мобильном устройстве последнего приложение 1С запущено не было. В таком случае с центральной базы посылается push-оповещение на устройство, соответствующее сотруднику и связанное с ним по уникальному коду устройства.

Если в этот момент сотрудник будет находиться вне зоны действия сети Интернет, то ничего не произойдёт. Но как только сотрудник появится в сети, ему на телефон придёт уведомление, при нажатии на которое будет осуществлен запуск приложения, а также активирован GPS для установления координат и отправки их в центральную базу.

После того, как необходимые данные поступят в центральную базу, у пользователя будет возможность увидеть текущее местонахождение на карте. Но так как данные с мобильного приложения присылаются в центральную базу периодически (рис. 2), то пользователь сможет увидеть соединенные точки, представляющие собой маршрут сотрудника за определенный период времени (рис. 3) и при необходимости провести анализ маршрута для выявления отклонений и принятий соответствующих санкций.

Точка	
Широта	Долгота
Точка1	
50,471937713865	30,508396428467
Точка2	
50,461090696983	30,523159306885
Точка3	
50,452761996659	30,609848302246
Точка4	
50,419761475085	30,544788640381
Точка5	
50,424806241567	30,460331242920

Рис. 2. Периодически загруженные данные о местоположении

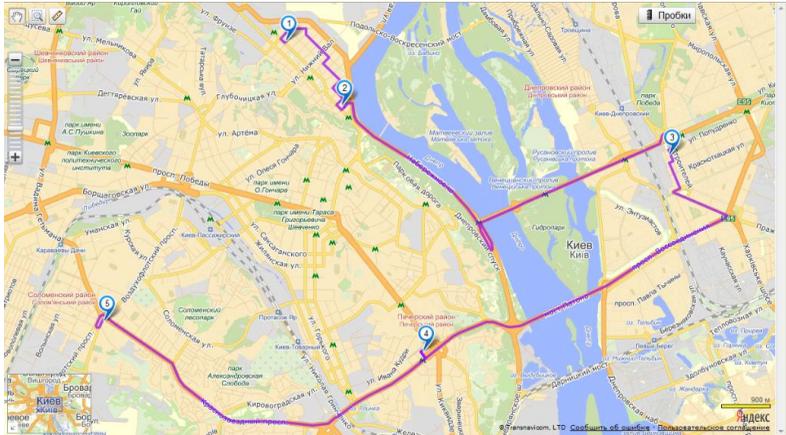


Рис. 3. Маршрут, построенный по точкам на карте

Заключение

Данная статья посвящена реализации механизма обмена данными между типовыми конфигурациями 1С или уникальными решениями и мобильным приложением 1С на платформе Android. Такой механизм необходим для реализации модуля «Геопозиционирование сотрудников», содержащего функционал, позволяющий периодически выгружать данные о местоположении сотрудника через мобильное приложение для дальнейшего анализа.

Список литературы

1. Радченко М.Г. 1С: Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / М.Г. Радченко, Е.Ю. Хрусталева. – М. : 1С-Паблишинг, 2013. – 964 с.
2. Ф. Насипов, Е. Гилев. Разработка мобильных приложений на 1С: Предприятие 8.3 [Электронный ресурс]. – Режим доступа : <http://курсы-по-1с.рф/1с-v8/1с-mobile-2016/>
3. Сервис-ориентированная архитектура [Электронный ресурс] : Материал из Википедии. – Режим доступа : https://ru.wikipedia.org/wiki/Сервис-ориентированная_архитектура
4. 1С: Предприятие 8. Заметки из Зазеркалья [Электронный ресурс]. – Режим доступа : <http://v8.1с.ru/o7/201502push/index.htm>

АНАЛИЗ МОДЕЛЕЙ КАНАЛОВ МНОГОЛУЧЕВОГО РАСПРОСТРАНЕНИЯ СИГНАЛА

Г.А. Белевский

Студент

Д.Н. Борисов

Доцент

Ведение

Необходимость в высокоскоростных соединениях, предоставляющих высокие показатели качества обслуживания с высокой отказоустойчивостью растет от года в год. Однако большинство беспроводных технологий не позволяют предоставить абонентам высокое качество обслуживания на краю зоны покрытия. В сотовых и других беспроводных системах связи качество соединения, также как и доступная скорость передачи данных стремительно падает с удалением от базовой станции [1]. Вместе с этим падает и качество услуг, что в итоге приводит к невозможности предоставления услуг реального времени с высоким качеством на всей территории радио покрытия сети. В настоящее время наиболее эффективным способом решения данной проблемы является использования технологий основанных на многолучевом распространении сигналов [2]. Для радиосигналов эффект многолучевого распространения может становиться положительным или отрицательным. Из-за разницы в фазах множества путей копий сигналов комбинированный сигнал на приемнике чаще будет затухать или будет повреждаться. В связи с чем в настоящее время данная область исследования является крайне актуальной.

1. Space-Time Block Code (STBC)

Самым первым и известным STBC (временной блок-код) является код Аламоути, который является сложным ортогональным пространственно-временным кодом, специализированным для случая двух передающих антенн [3].

Данный комплексный ортогональный пространственно-временной блок-код для двух передающих антенн был разработан Аламоути [3]. В кодере Аламоути два последовательных символа x_1 и x_2 кодируются матрицей кодового слова пространства-времени:

$$X = \begin{pmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{pmatrix}$$

Как показано на рис. 1, закодированный сигнал передается от двух передающих антенн в течение двух периодов. В течение первого периода, два символа x_1 и x_2 одновременно передаются от двух передающих антенн. Во время второго периода эти символы передаются повторно, причем с первой передающей антенны передается x_2^* , а со второй – x_1^* .



Рис. 1. Кодер Аламоути

2. MRC

Технология оптимального весового сложения (MRC) позволяет улучшить значение соотношение сигнал/шум (SNR) для принимаемого сигнала [2].

Алгоритм работы MRC подразумевает сбор на нескольких антеннах всех прямых и переотраженных сигналов в модели многолучевого распространения сигналов. Далее специальный процессор (DSP) отбирает лучший сигнал с каждого приемника и выполняет комбинирование. Фактически математическая обработка реализует виртуальный фазовый сдвиг для создания положительной интерференции сигналов со сложением. В результате, суммарный (результатирующий) сигнал значительно лучше по характеристикам, чем все исходные.

3. Моделирование каналов

Для исследования энергетической эффективности моделей каналов MRC и STBC рассмотрим частоту появления ошибочных битов (BER) при изменении соотношению сигнал/шум (E_b/N_0).

С использованием пакета прикладных программ Matlab для модели MRC была построена зависимость частоты ошибочных битов от отношения сигнал/шум для системы SIMO (одна передающая антенна, несколько принимающих, 1x2, 1x4), а так же для эталонной модели – системы SISO (одна передающая антенна, одна принимающая, 1x1). В работе исследовалась энергетическая эффективность для модуляции BPSK (рис. 2) (binary phase shift key, двоичная фазовая манипуляция) и QPSK (рис. 3) (quadrature phase shift keying, квадратурная фазовая манипуляция).

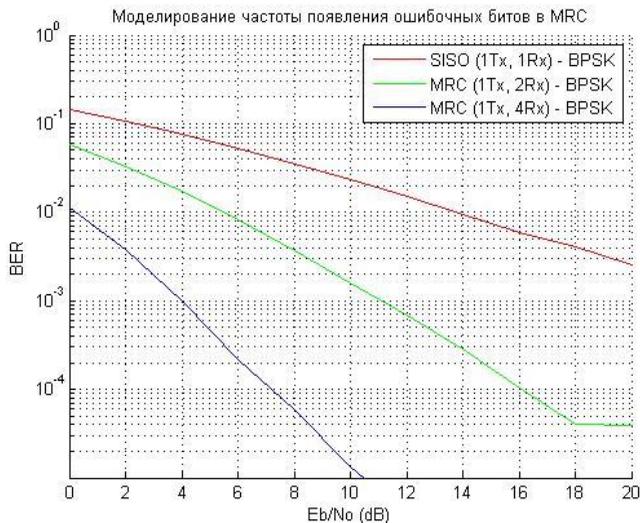


Рис. 2. Частота появления ошибок при BPSK модуляции

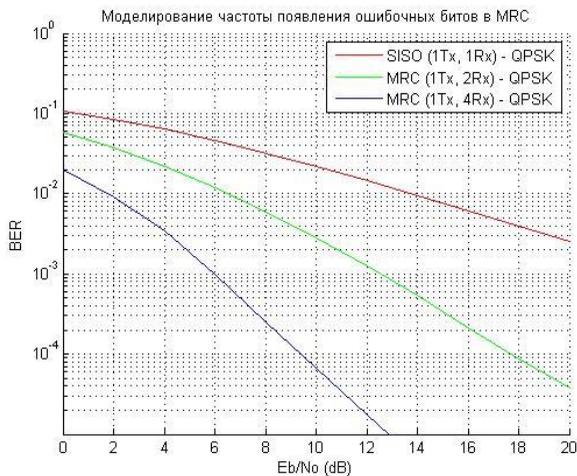


Рис. 3. Частота появления ошибок при QPSK модуляции

Для модели STBC была исследована зависимость частоты ошибочных битов по отношению сигнал/шум для системы MISO (несколько передающих антенн, несколько принимающих). В данной работе рассматривались случаи двух и четырех передающих и одной

принимающей антенны т. е. случаи 2×1 , 4×1 , так же в качестве эталонной модели, использовалась система SISO. Для данной модели в качестве наиболее востребованных на практике случаев исследовались сигналы с BPSK (рис. 4) и QPSK (рис. 5) модуляцией.

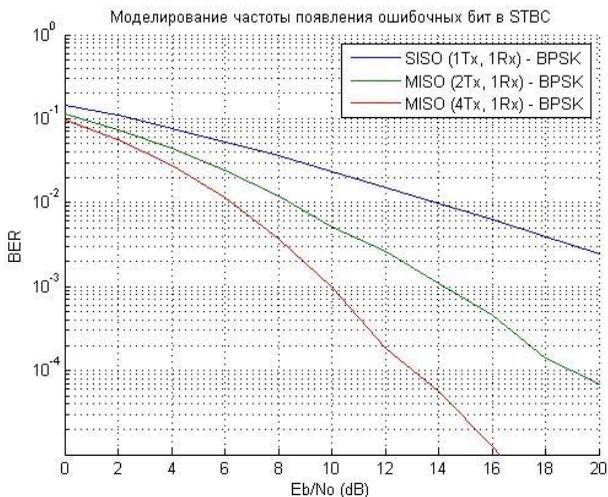


Рис. 4. Частота появления ошибок при BPSK модуляции

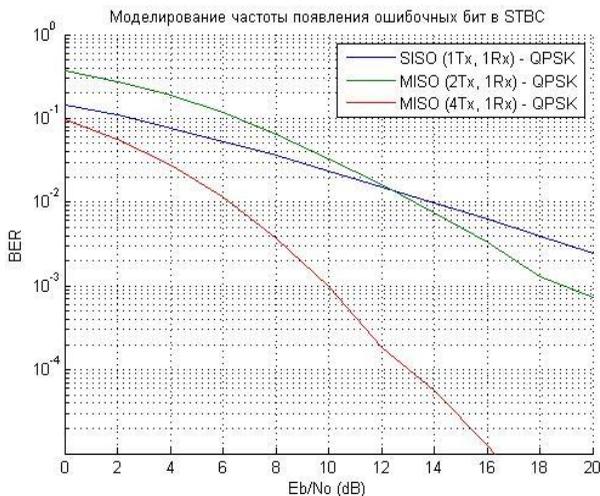


Рис. 5. Частота появления ошибок при QPSK модуляции

Заключение

Из результатов исследования можно сделать вывод о том, что для моделей каналов MRC и STBC с увеличением количества антенн появляется возможность получения наилучшей энергетической эффективности, однако вместе с этим повышается сложность аппаратного исполнения компонентов системы, возникает необходимость увеличения количества излучающих и принимающих антенн. Наилучшее соотношение BER к отношению сигнал/шум возможно при модуляции BPSK.

Результаты исследования показывают, что частота появления ошибочных бит при сравнении моделей STBC и MRC, различается на порядок в пользу MRC.

Список литературы

1. Сотовая связь [Электронный ресурс]. – Режим доступа : <http://celnet.ru/mimo.php>
2. MIMO [Электронный ресурс]. – Режим доступа : <http://www.mforum.ru/analit/MIMO.htm>
3. Alamouti, S.M. A simple transmit diversity scheme for wireless communications / S.M. Alamouti // IEEE Select. Areas Commun.. – 1998. – № 16(8). – P. 1451-1458.
4. Bottomley, G.E. Adaptive arrays and MLSE equalization / G.E. Bottomley // IEEE VTC'95. – 1995. – Vol. 1. – P. 50-54.

РАЗРАБОТКА ИНТЕРФЕЙСА ДЛЯ СЕМЕЙСТВА ОС LINUX АДАПТИРОВАННОГО ДЛЯ ЛЮДЕЙ ПОЖИЛОГО ВОЗРАСТА

Р.И. Бобков
Студент
С.А. Зуев
Ассистент

Введение

В наше время люди все больше и больше приобщаются к миру IT. Каждый день мы пользуемся компьютером для того, чтобы узнать актуальные новости, прогноз погоды, пообщаться в Интернет со своими друзьями и просто знакомыми. Мы, молодые люди, делаем это не задумываясь, как само собой разумеющееся. Для нас не составляет ничего сложного освоить интерфейс и воспользоваться основными функциями только что установленного приложения. Абсолютно противоположная

тенденция наблюдается при работе людей пожилого возраста с различными рода устройствами, будь то смартфон, планшет или персональный компьютер. Именно о решении данного вопроса и пойдет речь.

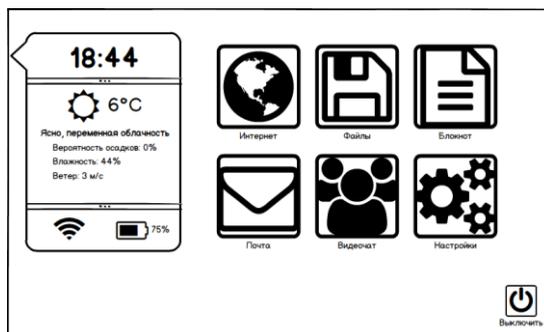
1. Требования к интерфейсу операционной системы

Для комфортной работы пользователя пожилого возраста необходимо обеспечить следующие условия [1]:

- максимальная простота интерфейса, нет лишних функций
- отсутствие всплывающих окон и меню, которые могут напугать пользователя
- интерфейс должен быть выполнен строго, без каких-либо анимаций и в одном цветовом решении
- должна быть возможность использования компьютерной мыши, так и сенсорного экрана
- разграничение прав на пользователя и суперпользователя
- доступ к суперпользователю по паролю, который имеют кураторы(родные пользователя/техподдержка)
- невозможность установки/удаления/настройки/скачивания приложений без прав суперпользователя
- одно рабочее приложение на экране, без возможности свернуть его
- невозможно открыть другое приложение, не закрыв запущенное
- работа интерфейса на слабых компьютерах/компьютерах без видеокарты

2. Концепция интерфейса

На рис. 1 можно видеть концепцию интерфейса, все построено в минималистичном стиле. Планируется использование модифицированного LXDE (Lightweight X11 Desktop Environment) [2, 3].



Главный экран пользователя после запуска системы

Рис. 1. Интерфейс пользователя

На главном экране расположены основные необходимые пользователю приложения (Таблица).

Таблица

Приложения пользователя

Интернет	Интернет-браузер, с функцией безопасного поиска и серфинга интернет-страниц. Включена фильтрация Adult-контента и мошеннических сайтов. Трафик идет через специальный сервер, который сжимает контент на сайте и передает его в сжатом виде пользователю. Тем самым мы экономим интернет-трафик пользователя и ускоряем загрузку страниц. Основан на WebKit (Chromium).
Файлы	Тривиальный менеджер файлов. Под режимом пользователя исключает загрузку и установку исполняющихся файлов. Запрещает загрузку файлов .exe. Не предоставляет доступ к системным директориям. Автоматически рассортировывает файлы по папкам: «Музыка», «Картинки», «Документы», «Прочее» основываясь на расширениях файлов. Основан на PCManFM (PCMan File Manager).
Блокнот	Обычный текстовый редактор, с минималистичным интерфейсом, основанный на abiWord.
Почта	Минималистичный почтовый клиент, который основан на Geary.
Видеочат	Текстовый и видео чаты, аналог Skype с собственной системой регистрации. Будет создан веб-клиент, для возможности общения без установки приложения. Так же будет разработан клиент для ОС Windows. Используемый протокол – Jingle (дополнение XMPP [4]) – поддерживает передачу голоса, видео и файлов по сети. Клиент основан на Pidgin.

Настройки	Самое важное приложение в системе. Обычный пользователь будет иметь доступ к малозначительным настройкам системы: фоновое изображение рабочего стола и цветовой гаммы, настройка времени, переключение языка. Подключение к сети Wi-Fi. Запуск обновления системы. Выход и авторизация пользователя. Суперпользователь будет иметь расширенные права – установка/настройка/удаление приложений. Откат обновления системы (в случае возникновения ошибок или несовместимости системы с оборудованием). Установка и настройка новых подключенных устройств. Изменение прочих системных настроек. Приложение «Настройки» планируется разработать с нуля.
-----------	---

На рис. 2 представлен экран пользователя после запуска браузера.

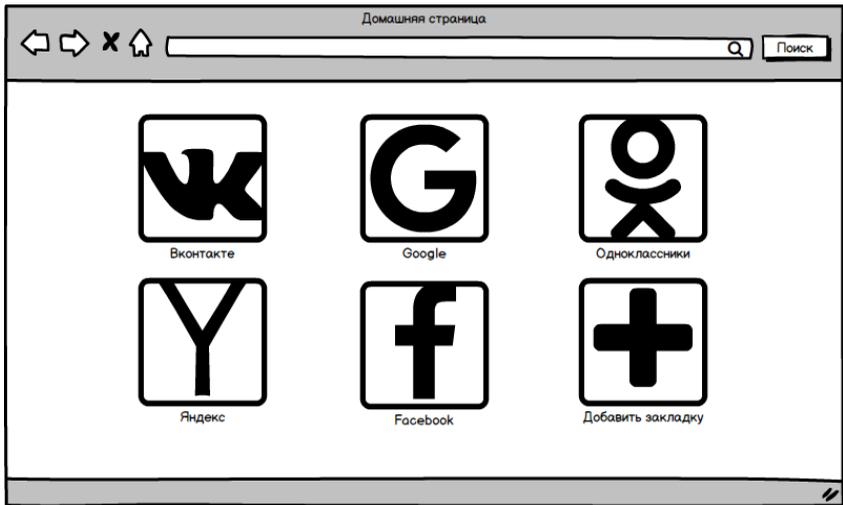


Рис. 2. Панель закладок в браузере

3. Концепция системы

В целом должен быть создан огромный стек программного и аппаратного обеспечения (рис. 3). Будет реализован центральный сервер (серверы), через который (которые) и будет проходить интернет-трафик всех пользователей. Серверы должны выполнять следующие функции:

- Контроль удаленного подключения к компьютеру пользователя. Подключение будет происходить с помощью протокола SSH.

- Фильтрация и сжатие интернет-трафика пользователя (исключение загрузки сайтов мошенников и adult-сайтов)
- Хранение учетных записей пользователя
- Ведение логов по каждому пользователю
- Раздача обновлений с сервера
- Установка приложений с сервера

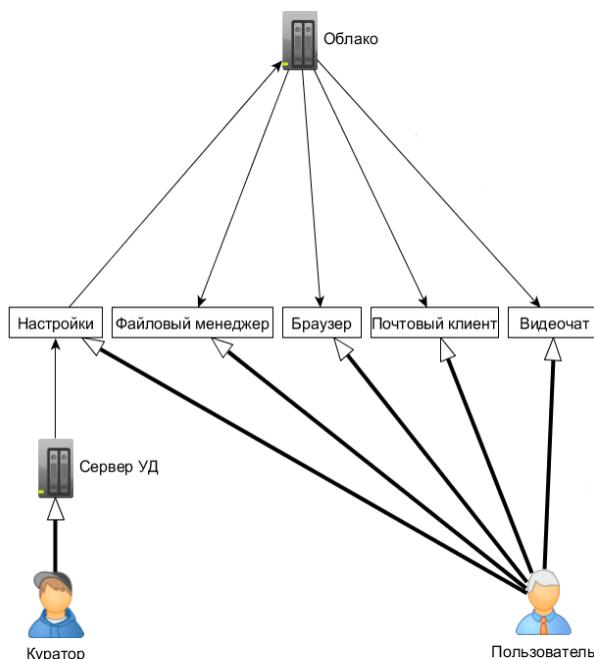


Рис. 3. Общий вид системы

Все это позволит контролировать систему и выявлять ошибки пользователя/операционной системы/приложение в оперативном режиме. Так же это позволит безопасно подключаться к компьютеру пользователя и производить манипуляции с настройками, доступ же к файлам пользователя будет ограничен.

Заключение

Естественно, такая операционная система и ее интерфейс в частности, не могут гарантировать быстрого и беспрепятственного освоения пользователем компьютера. Однако, возможность удаленного доступа для

исправления ошибок и помощи пользователю существенно экономит время и денежные средства куратору. Кроме того, данная совокупность программного обеспечения будет распространяться под лицензией GNU GPL, то есть open source, а значит, за нее не нужно платить ни рубля! Учитывая финансовое состояние большинства людей пожилого возраста – этот факт является неоспоримым преимуществом.

Список литературы

1. Linux from scratch [Электронный ресурс]. – Режим доступа : <http://www.linuxfromscratch.org/lfs/>
2. Russel J. LXDE / J. Russel, R. Cohn. – New York : Wiley, 2012. – 712 p.
3. Керниган Б. Язык программирования СИ / Б.Керниган, Д. Ритчи. – М. : Вильямс, 2016. – 288 с.
4. XMPP [Электронный ресурс]. – Режим доступа : <https://xmpp.org/>

РЕАЛИЗАЦИЯ АУДИО- И ВИДЕОКОНФЕРЕНЦИЙ НА МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ

Д. Ю. Богомолов

Студент

П. С. Лысачев

Старший преподаватель

Введение

В наше время все больше и больше становится компаний, имеющих представительство в нескольких городах или даже странах. Для таких компаний на первое место выходит проблема эффективной коммуникации между персоналом в разных частях света. Конечно, ничто не может быть эффективнее, чем живое общение, но аудио- и видеоконференции могут послужить неплохой заменой. С их помощью решаются следующие проблемы:

- Снижение расходов на командировки;
- Быстрое принятие решений в чрезвычайных ситуациях;
- Увеличение производительности персонала;
- Широкий и эффективный обмен опытом.

Аудио- и видеоконференции различают по типу взаимодействия людей. Первым типом является презентация, в которой выступает один

докладчик, а все остальные участники принимают информацию от него. Вторым типом является обсуждение, когда каждый из участников обменивается аудио- и видеoinформацией [1].

Также аудио- и видеоконференции можно разделить по количеству участников, а именно, на двусторонние и многосторонние.

В данной статье рассматривается способ организации аудио- и видеоконференции типа многосторонних обсуждений посредством WebRTC.

1. Принципы работы WebRTC

WebRTC – это набор протоколов и API, которые позволяют установить связь между пользователями через peer-to-peer соединение. Наиболее распространенным является случай, когда браузеры пользователей подключаются к одному веб-серверу и запускают одно и то же веб-приложение [2]. Такая конфигурация порождает треугольную схему подключения. В данной схеме ключевую роль играет сигнальный сервер. Он может быть как объединен с веб-сервером, так и выделен в самостоятельную часть приложения. Сигнальный сервер отвечает за пересылку служебной информации, такой как описание сессии и кандидаты, от одного браузера к другому до того, как наладилось непосредственное соединение между ними. После успешного подключения, обмен аудио и видео происходит непосредственно между браузерами, не используя сигнальный сервер.

Для того, чтобы установить связь между браузерами, необходимо проделать следующую последовательность действий [3]:

1. Пользователь А создает `RTCPeerConnection` – объект peer-to-peer соединения

2. Пользователь А с помощью протокола SDP, создает offer, содержащий описание сессии передачи потоковых данных и отправляет его Пользователю В через сигнальный сервер.

3. Пользователь В, в свою очередь, принимает offer от пользователя А и отправляет answer, содержащий описание сессии Пользователя В.

4. Оба пользователя регистрируют callback для отправки кандидатов друг другу.

Кандидат представляет из себя информацию о подключении пользователя к интернету, которую использует фреймворк ICE.

ICE пытается найти лучший способ соединения между клиентами. Сначала предпринимается попытка соединения напрямую с помощью адреса, полученного от другого устройства. Если такое соединение установить не получается (устройство находится за NAT), то используется STUN сервер [4].

STUN сервер используется для того, чтобы получить внешний адрес устройства, с которым смогут работать другие клиенты. Запрос на STUN сервер отсылается перед пересылкой кандидата другому клиенту через сигнальный сервер.

Если и с помощью STUN сервера установить соединение не представляется возможным, то используются TURN сервера. Их основной целью является ретрансляция мультимедиа потока, т.е. данные в таком случае идут не напрямую между клиентами, а через TURN сервер.

2. Реализация сигнального сервера

Так как сигнальный сервер явно не входит в спецификацию WebRTC, то его реализация целиком и полностью ложится на плечи разработчика. В данной работе основными инструментами реализации сигнального сервера являются язык программирования Java и фреймворк Spring.

Как следует из предыдущего пункта, сигнальный сервер должен обрабатывать следующие специфичные для WebRTC типы сообщений: offer, answer, ice candidate. Также были добавлены вспомогательные типы сообщений:

- Тип сообщения callrequest, который используется для изначальной рассылки запроса на звонок всем участникам конференции;
- Тип сообщения leave, который используется для фиксации выхода одного из пользователей из конференции.

Основными компонентами сигнального сервера являются два контроллера, работающих с клиентом и сервис, предоставляющий контроллерам информацию об активных пользователях (рис. 1).

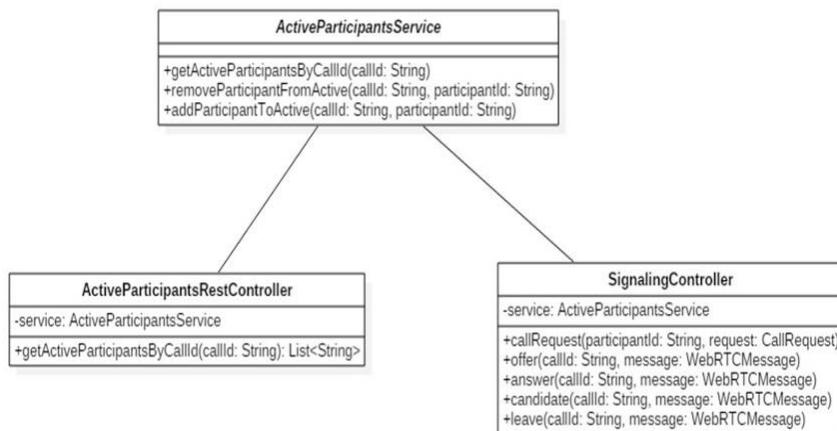


Рис. 1. Диаграмма классов сигнального сервера

Сервис позволяет получать активных пользователей по идентификатору звонка, добавлять пользователя к активным и удалять их из активных. Активными считаются пользователи, находящиеся в данный момент в конференции.

SignalingController – контроллер, ответственный за обработку основной массы сообщений, идущей от клиента. Обмен сообщениями между сигнальным сервером и клиентом происходит через протокол STOMP. STOMP является протоколом более высокого уровня над WebSockets и определяет механизм подписок на интересующие клиента темы, для получения сообщений от сервера. Также в данной реализации используется протокол SockJS поверх Websockets, который эмулирует HTML5 Websockets API. Это нужно для того, чтобы можно было перейти на резервный протокол, в случаях, когда браузер не поддерживает WebSockets или в ситуациях, когда прокси препятствует их использованию.

ActiveParticipantsRestController – контроллер, ответственный за обработку запросов на получение активных пользователей в звонке. Данный контроллер работает по традиционному механизму запрос-ответ по протоколу HTTP.

Так как сигнальный сервер является отдельным микросервисом, то ему достаточно хранить только идентификаторы звонков и активных пользователей. Задача хранения полных описаний этих сущностей перекладывается на другой микросервис внутри инфраструктуры приложения. Кроме передачи ответственности, выделение сигнального сервера в отдельный сервис позволяет обновлять функциональность аудио- и видеоконференций без перекомпиляции всего приложения, а также сохранять работоспособность этого микросервиса даже в случае сбоя остальных микросервисов.

3. Реализация клиентского приложения

Клиентское приложение было написано на языке Javascript стандарта ES6. Это позволило вынести логику по работе с Websockets в отдельные сервисы со статическими методами (рис. 2).

WebSocketService – сервис, предоставляющий основные функции для работы с Websockets, такие как: подключение к указанному адресу, отключение от него, подписка на определенную тему по указанному адресу, отписка от нее и отправка сообщения.

SignalingService – сервис, расширяющий WebSocketService и позволяющий подписаться на все необходимые для WebRTC темы (типы сообщений), отписаться от получения сообщений в случае выхода из конференции, а также отправлять сообщения на сигнальный сервер.

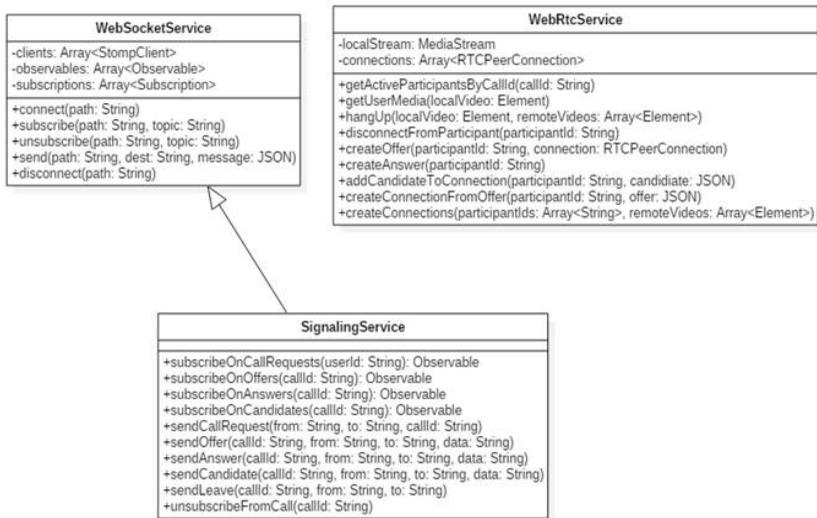


Рис. 2. Диаграмма классов клиентского приложения

WebRTCSERVICE – сервис, предоставляющий основные функции для работы с WebRTC. Например, получение локального потока мультимедиа, создание offer и answer, создание RTCPeerConnection, выход из конференции и т.д.

Для того, чтобы организовать конференцию на несколько человек, необходимо найти способ подключения нескольких членов конференции друг к другу. В данном случае была выбрана схема подключения “full mesh” (рис. 3).

Данная схема подразумевает подключение каждого браузера с каждым и по подключению нового пользователя, каждый активный клиент должен установить соединение с вновь подключившимся. Схема “full mesh” имеет несколько преимуществ над схемой с выделенным медиасервером [2]:

- Отсутствие выделенного медиасервера;
- Наименьшие задержки при передаче;
- Наилучшее качество;

Однако, такая схема является неприменимой для большого количества участников в конференции, так как каждое подключение требует выделенной полосы.

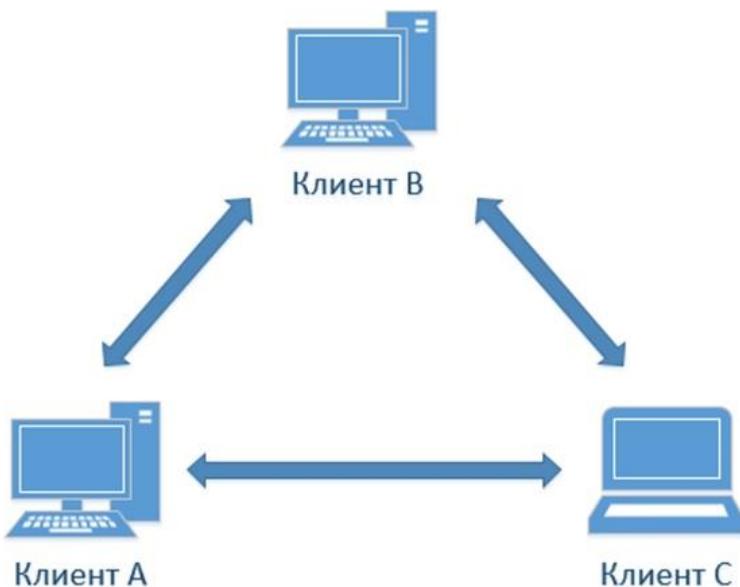


Рис. 3. «full mesh» схема

4. Принцип работы конференции

Соединение нескольких клиентов в конференции происходит по следующему принципу:

- При входе в систему клиент устанавливает WebSocket соединение с сигнальным сервером и подписывается на тему с типом сообщения callrequest и с идентификатором пользователя;

- После нажатия на кнопку “Создать конференцию” всем предполагаемым участникам рассылаются сообщения с типом callrequest на адрес с их идентификатором;

- В случае принятия звонка, подключившийся клиент делает HTTP запрос на сигнальный сервер с целью получения всех активных пользователей в данной конференции;

- После получения списка активных пользователей, клиент создает для каждого из них RTCPeerConnection и рассылает offer;

- Активный клиент, получивший offer, создает из него RTCPeerConnection и отправляет answer;

- Во время создания объекта RTCPeerConnection создается callback позволяющий отсылать кандидаты другой стороне соединения;

Цепочка событий для отсоединения клиента от конференции выглядит следующим образом:

- Пользователь нажимает на кнопку “Отсоединиться от конференции”;
- Клиент отправляет сообщение на сигнальный сервер с типом leave.
- Сигнальный сервер рассылает его всем активным клиентам в конференции;
- Активный клиент, получив сообщение с типом leave, закрывает соединение с отправителем сообщения;

Заключение

В рамках данной статьи был рассмотрен один из способов реализации аудио- и видео конференций по типу многосторонних обсуждений на основе WebRTC.

Рассмотренный подход базируется на основе микросервисной архитектуры и позволяет легко встроить функциональность аудио- и видеоконференций в другое приложение.

Список литературы

1. Types of Videoconferences [Электронный ресурс]. – Режим доступа : <https://si.ua.es/en/videoconferencia/types-of-videoconferences.html>
2. Johnston, A. WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web / A. Johnston, D. Burnett. – St. Louis : Digital Codex LLC, 2013. – 350 p.
3. WebRTC in the real world: STUN, TURN and signaling [Электронный ресурс] / S. Dutton. – Режим доступа : <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure>
4. Manson, R. Getting Started with WebRTC / R. Manson. – Birmingham : Packt Publishing, 2013. – 114 p.

DLP СИСТЕМА НА ОСНОВЕ СТАТИСТИЧЕСКИХ И СТЕГАНОГРАФИЧЕСКИХ МЕТОДОВ

А.В. Васильев

Студент

М.А. Дрюченко

Доцент

Введение

Системы защиты от утечек конфиденциальной информации (Data Loss Prevention – DLP) предназначены для отслеживания и блокирования попыток несанкционированной передачи данных за пределы корпоративной сети [1].

Помимо предотвращения утечек информации DLP система может выполнять функции по отслеживанию действий пользователей, записи и анализу их коммуникаций через e-mail, социальные сети, чаты и т.д. Основная задача систем DLP – обеспечение выполнения принятой в организации политики конфиденциальности (защита информации от утечки) [1].

Использование DLP системы наиболее актуально для организаций, где риск утечки конфиденциальной информации повлечет серьезный финансовый или репутационный ущерб, а также для организаций, которые настроенно относятся к лояльности своих сотрудников. Решения класса DLP по предотвращению утечек информации обеспечивают защиту такой конфиденциальной информации, как условия тендеров, заказы на услуги и решения, номера пластиковых карт, сведения о счетах клиентов, персональные данные сотрудников и клиентов, финансовые данные и т.д.

1. Основные функции DLP-систем

В первую очередь необходимо обозначить основные функции DLP-системы:

- контроль передачи информации через Интернет с использованием E-Mail, HTTP, HTTPS, FTP, Skype, ICQ и других приложений и протоколов;
- контроль сохранения информации на внешние носители – CD, DVD, flash, мобильные телефоны и т.п.;
- защита информации от утечки путем контроля вывода данных на печать;

- блокирование попыток пересылки/сохранения конфиденциальных данных, информирование администраторов ИБ об инцидентах, создание теневого копий, использование карантинной папки;
- поиск конфиденциальной информации на рабочих станциях и файловых серверах по ключевым словам, меткам документов, атрибутам файлов и цифровым отпечаткам;
- предотвращение утечек информации путем контроля жизненного цикла и движения конфиденциальных сведений.

2. Схема DLP системы и реализация

Предложенная схема DLP системы представлена на рисунке. В первую очередь стоит отметить центр управления и мониторинга. В крупной системе их может быть несколько. Данное программное обеспечение доступно администраторам и начальникам организации. Функционал:

- мониторинг активности;
- маркирование файлов по уровням доступа;
- настройка уровней доступа для всей сети или каждого ПК отдельно.

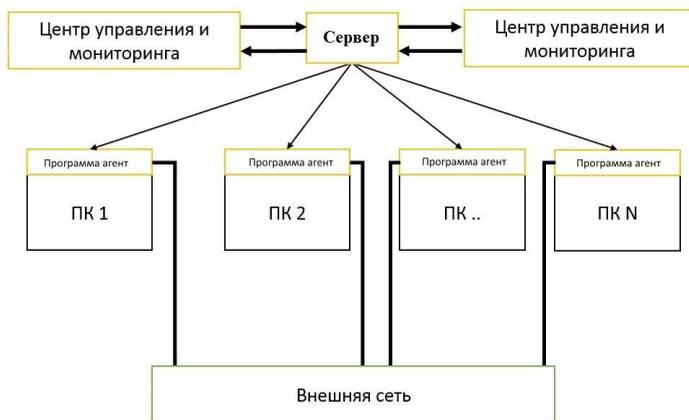


Рисунок. Схема DLP системы

На стороне ПК пользователей предполагается системное ПО. Назовем данную программу – агентом. Функционал программы-агента:

- Мониторинг папок, файлов;
- Мониторинг внешних носителей;
- Контроль внешних устройств;
- Контроль трафика.

3. Стеганографические методы

При реализации DLP системы возникает задача подписи файлов – маркирование по уровням доступа. Для того, чтобы конечный пользователь, которому предоставляется файл не мог изменить метку и передать файл третьим лицам, будут использоваться различные методы стеганографии. К базовым отображениям стеганографического преобразования данных относят [2]:

– скрывание информации $F_1 : Z \times D \times K \rightarrow \tilde{Z}$;

– восстановление информации $F_2 : \tilde{Z} \times K \rightarrow \tilde{D}$, где Z, \tilde{Z} – множество исходных пустых и заполненных – «защищенных» контейнеров; D, \tilde{D} – множество исходных и восстановленных сообщений, представляющих собой идентифицирующие конкретного пользователя цифровые метки; K – множество стеганографических ключей, используемых для кодирования и декодирования идентифицирующих меток.

Принципиально важным моментом, обуславливающим работоспособность разрабатываемой DLP-системы, является выбор стеганографических алгоритмов для скрывания меток. На начальном этапе при разработке пилотного варианта системы рассматривался классических метод метод замены наименьших значащих бит. Данный метод основан на возможности замены младших разрядов цифровых отсчетов аудио-, видео- и графических файлов, не несущих существенной информации на биты скрываемого сообщения. Из-за погрешностей в работе устройств аналого-цифрового преобразования наименее значимые биты (НЗБ) контейнера зачастую имеют случайный характер и описываются как шумы квантования. На этом же принципе основано и сжатие с потерями (JPEG, MP3, MP4 и пр.). Достоинством метода являются его простота и возможность встраивания достаточно больших объемов данных (пропускная способность скрытого канала связи составляет 12,5-30%). Недостатком является низкая стеганографическая стойкость, высокая чувствительность к малейшим искажениям контейнера [2, 3].

4. Проблема искажений маркированных контейнера

Одной из возникших проблем является искажение стеганографического контейнера при изменении и перезаписи файла пользователем. Чтобы уйти от неизбежного изменения контейнера необходимо отлеживать открытие любых файлов пользователем. Для этого событие открытия файла перехватывается программой-агентом, далее осуществляется проверка файла на наличие ранее встроенной метки и если таковая имеется, программа переходит в активный режим. Активный режим предусматривает проверку уровня доступа пользователя. Если прав на просмотр файла у пользователя нет, должен последовать отказ в

открытии/печати/отправке. Если права на открытие файла успешно проверены, программа-агент в отдельном потоке, дожидается закрытия файла. После этого в фоновом режиме (незаметно для пользователя) производится процедура стеганографической «перемаркировки» файла соответствующими метками, идентифицирующими данного пользователя.

Важным аспектом является промежуток между открытием и закрытием файла. Пользователь может сохранить файл с другим названием, сделать снимок экрана стандартными средствами Windows. Активный режим обязан пометить создаваемые файлы как потенциально опасные – содержащие тем или иным образом данные из маркированного файла. Новые метки, стеганографически записываемые в потенциально опасные файлы, должны наследовать уровень доступа исходной метки, а также дополнительно включать в себя данные пользователя: данные о ПК, данные учетной записи пользователя, время работы с фалом и т.п.

Разрабатываемый и реализуемый в настоящий момент пилотный вариант DLP-системы включает следующие функциональные возможности:

- маркирование документов различных форматов;
- отслеживание открытия/закрытия файлов, работа с файловой системой, процессами;
- блокировка действий пользователя на ПК в зависимости от действий с файлами;
- инструменты администрирования.

Заключение

В рамках данной работы была описана основная концепция DLP-системы, использующей стеганографические методы в интересах контроля несанкционированной передачи данных за пределы корпоративной сети. Рассмотрены основные функциональные особенности разрабатываемой DLP-системы.

Список использованной литературы

1. Выбираем DLP-систему для средней организации [Электронный ресурс]. – режим доступа : <https://habrahabr.ru/post/141000/>
2. Грибунин, В.Г. Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев.– М. : СОЛОН-ПРЕСС, 2009. – 384 с.
3. Fridrich J. Steganography in digital media / J. Fridrich. – Cambridge: Cambridge university press, 2011. – 437 p.

ПОСТРОЕНИЕ И АНАЛИЗ СОЦИАЛЬНЫХ ГРАФОВ ПОЛЬЗОВАТЕЛЕЙ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»

Т.С. Гаршин

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Анализ социальных сетей является новым направлением в изучении социального взаимодействия, коммуникации в различных социальных группах, которое получило широкое распространение в бизнесе, коммуникативных исследованиях, экономике, информатике, политологии, социальной психологии. При регистрации и заполнении своего профиля в сети пользователь вносит информацию различного рода о себе, своих увлечениях, взглядах, поэтому профили пользователей представляют собой богатые ресурсы для аналитической, исследовательской, коммерческой деятельности. На основе данной информации разрабатываются подходы к решению следующих задач:

- обнаружение профилей, принадлежащих одному человеку, в нескольких социальных сетях;
- выявление настоящих друзей, родственников, коллег;
- определение скрытых друзей и т.д.;
- решение маркетинговых задач (реклама продукции, информирование о выгодных предложениях, акциях, новинках и т.д.);
- помощь в рекрутинге персонала;
- помощь в разработке и создании различных сервисов и приложений для пользователей;
- анализ электорального поведения и настроений;
- генерация рекомендаций друзей (Mail.ru, «ВКонтакте») и рекомендаций медиа-контента, сообществ, новостей и т. п. [1].

1. Социальный граф и граф интересов пользователей

Социальный граф – это граф, узлы которого могут быть представлены людьми, организациями, предприятиями, городами, странами и т.д., а связи между ними – обмен ресурсами, сотрудничество, дружба, родство. Рассматривая Интернет-сообщества, узлами веб-графов будут выступать профили пользователей, веб-страницы, блоги; ребрами – взаимоотношения между ними (цитирование, дружба, веб-ссылки, подписки). Узлы сети (акторы) могут быть описаны атрибутами (имя, пол, место проживания,

дата рождения и т.д.), связи между которыми образует сетевую структуру, и могут иметь в разных местах различную плотность. Наиболее важными с точки зрения изучения, как отдельных элементов, так и такой системы в целом, являются структура отношений между акторами и местоположение отдельных акторов в сети [2].

Интересным объектом для анализа интернет-сообщества является эго-сеть. Эго-сеть (личная сеть) – совокупность социальных связей отдельного индивида (список «друзей» пользователя). Анализ эго-сети, позволяет вычислить родственников, коллег, скрытых друзей, а также получить интересующие данные о недостающих (неуказанных) атрибутах индивида. Полная эго-сеть представляет собой связи всех узлов списка (друзей пользователя) друг с другом. В данной работе рассматривается задача построения и изучения личных сетей контактов пользователей социальной сети «ВКонтакте».

2. API «ВКонтакте»

Целью работы является разработка программного инструментария для сбора данных о дружественных связях пользователей в социальной сети «ВКонтакте» с целью построения графа контактов и исследования его.

Для доступа к данным профилей пользователей социальных сетей существуют специальные API-интерфейсы («Инстаграм» – api.instagram.com, «ВКонтакте» – api.vk.com), позволяющие получать данные с помощью соответствующих методов (см. табл. 1). Обращение к различным методам осуществляется путем POST или GET запросов, которые состоят из:

- `method_name` – название метода API, к которому происходит обращение;
- `parameters` – входные параметры соответствующего метода API;
- `access_token` – ключ доступа;
- `v` – используемая версия API.

Таблица 1

Методы API VK для работы с пользователем и друзьями

<code>users.get</code>	Возвращает расширенную информацию о пользователях
<code>users.search</code>	Возвращает список пользователей в соответствии с заданным критерием поиска
<code>users.getFollowers</code>	Возвращает список идентификаторов пользователей, которые являются подписчиками пользователя

users.getSubscriptions	Возвращает список идентификаторов пользователей и сообществ, которые входят в список подписок пользователя
friends.get	Возвращает список идентификаторов друзей пользователя или расширенную информацию о друзьях пользователя (при использовании параметра fields)
friends.getMutual	Возвращает список идентификаторов общих друзей между парой пользователей
friends.search	Позволяет искать по списку друзей пользователей. Для расширенного поиска по списку друзей можно использовать метод users.search с параметром from_list = friends
friends.getList	Возвращает список меток друзей пользователя

Также, существует класс специализированных программ, ориентированных на анализ социальных сетей, примеры некоторых из них:

- UCINET (www.analytictech.com/ucinet/);
- Pajek (pajek.imfm.si) для анализа больших сетей;
- STOCNET (stat.gamma.rug.nl/stocnet).

Также можно выделить класс программ для отображения графов (визуализаторов), позволяющих также рассчитать характеристики и провести анализ графа:

- Cytoscape (<http://www.cytoscape.org/>);
- GraphInsight;
- Social Networks Visualizer;
- netminer (<http://www.netminer.com/index.php>).

В данной работе решается учебная задача, в которой интегрируется решение 3-х подзадач:

- построение графа связей конкретного пользователя социальной сети «ВКонтакте»;
- визуализация графа;
- расчет параметров, характеризующих структуру графа.

3. Построение графа социальных контактов

Для реализации приложения использовались:

- язык программирования Python (версия 3);
- библиотека NetworkX (<http://networkx.github.io/>) для создания и изучения структуры, динамики и функционирования сложных сетевых структур;

– библиотека `matplotlib` (<https://matplotlib.org/>) для визуализации графа.

Для построения графа дружеских связей использовались API-методы `users.get`, `friends.get` и `friends.getMutual`. `VkFriends` – основной класс приложения, в котором реализована основная функциональность приложения. Поиск друзей пользователя с заданным `id` осуществляется методом `friends`, приведенным на листинге 1:

Листинг 1

Метод friends класс VkFriends

```
def friends(self, id):
    r = requests.get(self.request_url(
        'friends.get',
        'user_id=%s&fields=uid,first_name,last_name, '+
        'photo,country,city,sex,bdate' % id
    )).json()['response']
    return {item['id']: item for item in r['items']],
            r['count']}
```

Метод `deep_friends`, приведенный на листинг 2, возвращает словарь с `id` пользователей, которые являются друзьями или друзьями друзей (и т.д. в зависимости от `deep` – глубины поиска). Учитывая, что для построения графа требуется выполнить множество `http`-запросов, для быстрого действия в реализации метода задействована многопоточность.

Листинг 2

Метод deep_friends класс VkFriends

```
def deep_friends(self, deep):
    result = {}

    @force
    def worker(i):
        r = requests.get(self.request_url(
            'execute.deepFriends',
            'targets=%s' % VkFriends.make_targets(i),
            access_token=True
        )).json()['response']
        for x, id in enumerate(i):
            result[id] =
                tuple(r[x]["items"]) if r[x] else None

    def fill_result(friends):
        with ThreadPoolExecutor(
            max_workers=self.max_workers
        ) as pool:
            [
                pool.submit(worker, i)
                for i in friends
            ]
```

```

        for i in VkFriends.parts(friends)
    ]

for i in range(deep):
    if result:
        fill_result(list(
            set([item for sublist in result.values()
                if sublist for item in sublist]) -
            set(result.keys())
        ))
    else:
        fill_result(requests.get(self.request_url(
            'friends.get',
            'user_id=%s' % self.my_id,
            access_token=True
        )).json()["response"]["items"])

return result

```

На рисунке ниже приведен результат визуализации графа связей конкретного пользователя «ВКонтакте». Визуализация графа позволяет среди друзей конкретного пользователя увидеть нескольких кластеров из тесно связанных пользователей.

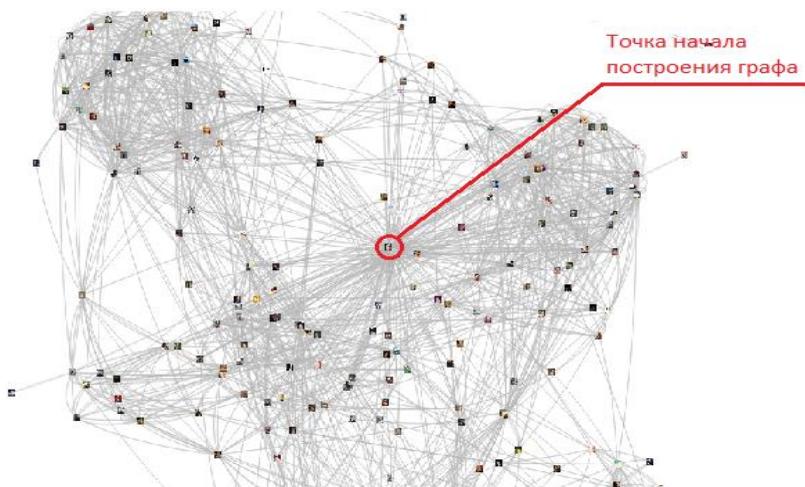


Рисунок. Граф социальных контактов пользователя социальной сети «ВКонтакте» (Id145466999)

Параметры построенного графа социальных контактов для данного пользователя приведены в таблицу 2.

Таблица 2

Параметры графа социальных контактов пользователя социальной сети «ВКонтакте» (Id145466999)

Параметр графа	Описание	Значение
Количество узлов		108
Количество связей		891
Плотность	Отношение числа имеющихся рёбер графа к максимально возможному количеству рёбер данного графа	0,09
Диаметр	Максимальный эксцентриситет	3
Среднее геодезическое расстояние	Расстоянием между двумя вершинами графа называется число рёбер в кратчайшем пути	1,92
Средняя степень узла	Среднее число связей q_i , с которыми этот узел связан с другими узлами	9

Заключение

В результате проделанной работы реализовано приложение для построения и анализа графа социальных контрактов для пользователя социальной сети «ВКонтакте». Полученные наработки могут быть использованы для осуществления более глубокого анализа социальных связей пользователей данной социальной сети.

Список литературы

1. Бартуков С. Идентификация пользователей социальных сетей в Интернет на основе социальных связей / С. Бартуков, А. Коршунов [Электронный ресурс]. – Режим доступа : http://web.iis.nsk.su/files/identifikatciia_polzovatelei_sotcialnykh_setei.pdf
2. Евин И.А. Введение в теорию сложных сетей / И.А. Евин // Компьютерные исследования и моделирование. – 2010. – Т. 2. – № 2. – С. 121-141.

РЕШЕНИЕ СИСТЕМ ПОЛИНОМИАЛЬНЫХ УРАВНЕНИЙ В ПАКЕТЕ MAPLE

К.Г. Гетманова,
Магистрант
В.Е. Павлов
Магистрант
А.В. Лобода
Профессор

Введение

В данной работе представлено решение одной системы полиномиальных уравнений. Поставленная задача связана с исследованием вопроса однородности геометрических объектов в многомерных пространствах [1].

Известно, что универсальных методов решения произвольных систем полиномиальных уравнений не существует. Достаточно сильные методы решения некоторых типов таких систем предоставляет техника базисов Гребнера. В изучаемой в докладе задаче была предпринята попытка прямого использования пакета Groebner. В итоге (что типично для алгоритма Гребнера [2]) были получены уравнения высоких степеней с целочисленными коэффициентами, порядок которых оценивается числами типа e^{118} .

Работать с такими уравнениями неудобно, потому в качестве основного средства при решении нашей задачи использовались преобразования отдельных уравнений системы. В отличие от линейных систем, результаты преобразования отдельных полиномиальных уравнений могут приводить к системам, не эквивалентным исходным. Чтобы не задаваться вопросом эквивалентности преобразований, выполним в конце проверку всех полученных в ходе исследования «решений».

1. Постановка задачи

Требуется решить систему четырех полиномиальных уравнений относительно трех неизвестных, принимающих действительные значения. Уравнения имеют вид:

$$\begin{aligned} -18\omega_3^2\omega_{12} - 18\omega_{10}^2\omega_{12} - 27\omega_{10}^3 - 27\omega_{10}\omega_3^2 + 8\omega_3 &= 0, \\ -84\omega_{10}^5\omega_{12}^2 - 24\omega_{10}^4\omega_{12}^3 + 128\omega_3\omega_{10}^3\omega_{12} - 90\omega_{10}^6\omega_{12} - \end{aligned} \quad (1)$$

$$\begin{aligned}
& -456\omega_3^2\omega_{10}^4\omega_{12} - 240\omega_3^2\omega_{10}^3\omega_{12}^2 - 156\omega_{10}\omega_3^4\omega_{12}^2 + \\
& + 32\omega_3\omega_{10}^2\omega_{12}^2 - 32\omega_3^3\omega_{12}^2 - 168\omega_3^6\omega_{12} + 200\omega_3^3\omega_{10}^2 - \quad (2) \\
& -156\omega_3^6\omega_{10} - 369\omega_3^4\omega_{10}^3 - 534\omega_3^4\omega_{12}\omega_{10}^2 + 64\omega_{10}\omega_3^3\omega_{12} + \\
& + 120\omega_3\omega_{10}^4 - 240\omega_3^2\omega_{10}^5 + 24\omega_3^4\omega_{12}^3 - 27\omega_{10}^7 + 96\omega_3^5 = 0, \\
& 261\omega_3^4\omega_{10}^2 + 24\omega_3^6\omega_{12} - 40\omega_3^3\omega_{10}^2 + 516\omega_3^6\omega_{10} + 204\omega_3^2\omega_{10}^5 + \\
& + 459\omega_{10}^7 - 704\omega_3\omega_{10}^3\omega_{12} - 160\omega_3\omega_{10}^2\omega_{12}^2 + 1248\omega_3^2\omega_{10}^4\omega_{12} + \\
& + 1026\omega_{10}^6\omega_{12} - 32\omega_3^5 - 24\omega_3^4\omega_{12}^3 + 246\omega_3^4\omega_{12}\omega_{10}^2 + 128\omega_{10}\omega_3^3\omega_{12} + \quad (3) \\
& + 660\omega_{10}^5\omega_{12}^2 + 672\omega_3^2\omega_{10}^3\omega_{12}^2 + 12\omega_{10}\omega_3^4\omega_{12}^2 + 120\omega_{10}^4\omega_{12}^3 + \\
& + 96\omega_{10}^2 + 96\omega_3^5\omega_{12}^3\omega_3^2 + 32\omega_3^3\omega_{12}^2 - 632\omega_3\omega_{10}^4 = 0, \\
& 32\omega_3^5\omega_{12} - 32\omega_3^3\omega_{10}\omega_{12}^2 - 96\omega_3^8 + 24\omega_3^6\omega_{12}^2 + 16\omega_3^5\omega_{10} + \\
& + 72\omega_3^3\omega_{10}^3 - 342\omega_3^6\omega_{10}^2 - 507\omega_{10}^4\omega_3^4 - 342\omega_{10}^6\omega_3^2 - 81\omega_{10}^8 + \\
& + 72\omega_3\omega_{10}^5 + 24\omega_3^4\omega_{10}\omega_{12}^3 - 654\omega_{10}^3\omega_{12}\omega_3^4 - 192\omega_3^6\omega_{10}\omega_{12} - \quad (4) \\
& - 624\omega_{10}^5\omega_{12}\omega_3^2 - 264\omega_{10}^4\omega_{12}^2\omega_3^2 - 132\omega_{10}^2\omega_{12}^2\omega_3^4 - 162\omega_{10}^7\omega_{12} - \\
& - 108\omega_{10}^6\omega_{12}^2 - 24\omega_{10}^5\omega_{12}^3 + 96\omega_3\omega_{10}^4\omega_{12} + 64\omega_3^3\omega_{10}^2\omega_{12} + 32\omega_3\omega_{10}^3\omega_{12}^2 = 0.
\end{aligned}$$

Кроме того, считается, что выполняется ограничение:

$$\omega_3 \neq 0. \quad (5)$$

2. Практическая часть

Перепишем уравнение (1), обращая внимание на его линейность относительно переменной ω_{12} :

$$-18(\omega_3^2 + \omega_{10}^2)\omega_{12} - 27\omega_3^3 - 27\omega_{10}\omega_3^2 + 8\omega_3 = 0. \quad (6)$$

Из условия (5) следует, что $\omega_3^2 + \omega_{10}^2 \neq 0$, тогда решим уравнение (1) относительно ω_{12} :

$$\omega_{12} = -\frac{1}{18} \cdot \frac{-8\omega_3 + 27\omega_{10}^3 + 27\omega_{10}\omega_3^2}{\omega_3^2 + \omega_{10}^2}. \quad (7)$$

Подставим выражение (7) в начальные уравнения (1-4). После подстановки уравнения усложняются: например, уравнение (2), которое содержит 20 слагаемых степени, не превосходящей 7, переходит в полиномиальное уравнение 13 степени, хотя число слагаемых уменьшается до 17.

Средствами математического пакета Maple удобно упрощать подобные выражения, так функция *factor* осуществляет разложение на

множители [3]. Тогда уравнение (1) обнуляется, а остальные уравнения после разложения на множители принимают вид:

$$\begin{aligned} \frac{32\omega_3^2}{243} \cdot \frac{Q_1 Q_2 Q_3}{(\omega_3^2 + \omega_{10}^2)^2} &= 0, \\ \frac{32}{243} \cdot \frac{Q_4}{(\omega_3^2 + \omega_{10}^2)^2} &= 0, \\ -\frac{32\omega_3^3}{243} \cdot \frac{Q_5 Q_2 Q_3}{(\omega_3^2 + \omega_{10}^2)^2} &= 0, \end{aligned} \quad (8)$$

где

$$Q_1 = 9\omega_{10}\omega_3^2 + 2\omega_3 + 9\omega_{10}^3, \quad (9)$$

$$Q_2 = 9\omega_3^3 - 4\omega_3 + 9\omega_{10}\omega_3^2 + 4\omega_{10} + 9\omega_3\omega_{10}^2 + 9\omega_{10}^3, \quad (10)$$

$$Q_3 = 9\omega_3^3 + 4\omega_3 - 9\omega_{10}\omega_3^2 + 4\omega_{10} + 9\omega_3\omega_{10}^2 - 9\omega_{10}^3, \quad (11)$$

$$\begin{aligned} Q_4 = & 3645\omega_{10}\omega_3^8 - 162\omega_3^7 + 7290\omega_{10}^3\omega_3^6 - 1458\omega_{10}^2\omega_3^5 + 288\omega_{10}\omega_3^4 - \\ & - 2430\omega_3^3\omega_{10}^4 + 32\omega_3^3 - 7290\omega_{10}^7\omega_3^2 - 288\omega_3^2\omega_{10}^3 - 160\omega_3\omega_{10}^2 - \\ & - 1134\omega_3\omega_{10}^6 - 576\omega_{10}^5 - 3645\omega_{10}^9, \end{aligned} \quad (12)$$

$$Q_5 = 9\omega_3\omega_{10}^2 - 2\omega_{10} + 9\omega_3^3. \quad (13)$$

Если исходная система четырех уравнений имеет решения, то все они одновременно являются и решениями системы (8).

Ясно также, что после тривиальных преобразований система (8) равносильна более простой системе:

$$\begin{aligned} Q_1 Q_2 Q_3 &= 0, \\ Q_4 &= 0, \\ Q_5 Q_2 Q_3 &= 0. \end{aligned} \quad (14)$$

В свою очередь решение системы (14) сводится к рассмотрению следующих трех случаев:

$$Q_2 = 0, Q_4 = 0, \quad (15)$$

$$Q_3 = 0, Q_4 = 0, Q_2 \neq 0, \quad (16)$$

$$Q_1 = 0, Q_4 = 0, Q_5 = 0, Q_2 \neq 0, Q_3 \neq 0. \quad (17)$$

Исследование следует начать с обнуления Q_2 , так как этот множитель содержится в двух уравнениях системы (14).

3. Рассмотрение трех случаев

Рассмотрим сначала первый из выделенных случаев, т.е. систему (15). Построим базис Гребнера для многочленов Q_2, Q_4 :

$$36\omega_3^3 + 919\omega_3^5 - 10386\omega_3^7 + 810\omega_3^9 + 30618\omega_3^{11} + 19683\omega_3^{13} = 0, \quad (18)$$

$$-55721196\omega_3 + 1643620511\omega_3^3 - 1902842711\omega_3^5 - 692245845\omega_3^7 +$$

$$+ 34359631137\omega_3^9 + 22865111244\omega_3^{11} + 55721196\omega_{10} = 0. \quad (19)$$

Уравнение (19) линейно относительно переменной ω_{10} , а уравнение (18) зависит только от ω_3 и раскладывается на множители:

$$\omega_3^3(3\omega_3 + 2)(3\omega_3 - 2)(2187\omega_3^8 + 4374\omega_3^6 + 2034\omega_3^4 - 250\omega_3^2 - 9) = 0. \quad (20)$$

Линейные скобки $(3\omega_3 + 2)$ и $(3\omega_3 - 2)$ дают два решения системы (13):

$$\left(\omega_3 = -\frac{2}{3}, \omega_{10} = 0, \omega_{12} = -\frac{2}{3} \right), \quad (21)$$

$$\left(\omega_3 = \frac{2}{3}, \omega_{10} = 0, \omega_{12} = \frac{2}{3} \right). \quad (22)$$

Еще один множитель $(2187\omega_3^8 + 4374\omega_3^6 + 2034\omega_3^4 - 250\omega_3^2 - 9)$ из уравнения (20) дает после введения замены $t = \omega_3^2$ уравнение четвертой степени:

$$2187t^4 + 4374t^3 + 2034t^2 - 250t - 9 = 0. \quad (23)$$

Уравнение (23) разрешимо в радикалах [4].

Согласно вычислениям в Maple, уравнение (23) имеет единственный положительный корень $t = -\frac{1}{2} + \frac{1}{18}\sqrt{97} + \frac{1}{54}\sqrt{-42 + 6\sqrt{97}}$, по которому восстанавливаются значения переменной $\omega_3 = \pm\sqrt{t}$ и соответствующие им значения ω_{10} и ω_{12} .

Таким образом, система (15) имеет решения (21-22) и еще два решения:

$$\omega_3 = \frac{1}{18}\sqrt{-162 + 18\sqrt{97} + 6\sqrt{-42 + 6\sqrt{97}}},$$

$$\omega_{10} = -\frac{(\sqrt{97} + 31)\sqrt{-42 + 6\sqrt{97}} - 12\sqrt{97} - 156}{3888} \cdot \sqrt{-162 + 18\sqrt{97} + 6\sqrt{-42 + 6\sqrt{97}}}, \quad (24)$$

$$\omega_{12} = -\frac{(\sqrt{97} + 13)\sqrt{-42 + 6\sqrt{97}} - 12\sqrt{97} - 84}{216\sqrt{-42 + 6\sqrt{97}}}.$$

$$\begin{aligned}
& \cdot \sqrt{-162+18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} . \\
\omega_3 &= -\frac{1}{18}\sqrt{-162+18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} , \\
\omega_{10} &= \frac{(\sqrt{97}+31)\sqrt{-42+6\sqrt{97}}-12\sqrt{97}-156}{3888} \cdot \\
& \cdot \sqrt{-162+18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} , \\
\omega_{12} &= \frac{(\sqrt{97}+13)\sqrt{-42+6\sqrt{97}}-12\sqrt{97}-84}{216\sqrt{-42+6\sqrt{97}}} \cdot \\
& \cdot \sqrt{-162+18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} .
\end{aligned} \tag{25}$$

Рассмотрим теперь второй выделенный случай, описываемый системой (16). Выполняя действия, подобные описанным выше при исследовании случая (15), получим:

$$\omega_3^3(9\omega_3^2+4)(2187\omega_3^8-4374\omega_3^6+2034\omega_3^4+250\omega_3^2-9)=0. \tag{26}$$

Согласно условию (5), все действительные решения уравнения (26) возникают из более простого уравнения:

$$2187\omega_3^8-4374\omega_3^6+2034\omega_3^4+250\omega_3^2-9=0. \tag{27}$$

По аналогии с системой (15) уравнение (27) имеет единственный положительный корень $t_2 = \frac{1}{2} - \frac{1}{18}\sqrt{97} + \frac{1}{54}\sqrt{-42+6\sqrt{97}}$.

Для системы (16) получены следующие решения:

$$\begin{aligned}
\omega_3 &= \frac{1}{18}\sqrt{162-18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} , \\
\omega_{10} &= -\frac{(\sqrt{97}+31)\sqrt{-42+6\sqrt{97}}+12\sqrt{97}-156}{3888} \cdot \\
& \cdot \sqrt{162-18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} , \\
\omega_{12} &= \frac{(\sqrt{97}+13)\sqrt{-42+6\sqrt{97}}+12\sqrt{97}+84}{216\sqrt{-42+6\sqrt{97}}} \cdot \\
& \cdot \sqrt{162-18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} , \\
\omega_3 &= -\frac{1}{18}\sqrt{162-18\sqrt{97}+6\sqrt{-42+6\sqrt{97}}} ,
\end{aligned} \tag{28}$$

$$\omega_{10} = \frac{(\sqrt{97} + 31)\sqrt{-42 + 6\sqrt{97}} + 12\sqrt{97} + 156}{3888} \cdot \sqrt{162 - 18\sqrt{97} + 6\sqrt{-42 + 6\sqrt{97}}}, \quad (29)$$

$$\omega_{12} = -\frac{(\sqrt{97} + 13)\sqrt{-42 + 6\sqrt{97}} + 12\sqrt{97} + 84}{216\sqrt{-42 + 6\sqrt{97}}} \cdot \sqrt{162 - 18\sqrt{97} + 6\sqrt{-42 + 6\sqrt{97}}}.$$

Рассматривая, наконец, третий случай, т.е. систему (17), заметим, что

$$-\omega_{10}Q_5 + \omega_3Q_1 = 2(\omega_3^2 + \omega_{10}^2) = 0. \quad (30)$$

Такое равенство противоречит начальному условию задачи (5). Это означает, что система уравнений (17) не имеет решений.

Заключение

В результате исследования системы полиномиальных уравнений (1-4) было установлено, что она имеет действительные решения.

Проверка подтвердила, что все шесть выписанных «решений» удовлетворяют исходной системе четырех уравнений. Других решений, кроме найденных, система (8), а с ней и исходная система, не имеют. Таким образом, множество найденных решений полно.

Список литературы

1. Лобода, А.В. Использование компьютерных алгоритмов в задаче коэффициентного описания однородных поверхностей / А.В. Лобода, В.И. Суковых // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2015. – № 1. – С. 14-22.
2. Аржанцев, И.В. Базисы Грёбнера и системы алгебраических уравнений / И.В. Аржанцев. – Москва: МЦНМО, 2003. – 68 с.
3. Манзон, Б.М. Maple V. Power Edition / Б.М. Манзон. – М.: Информационно-издательский дом Филинь, 1998. – 240 с.
4. Курош, А.Г. Курс высшей алгебры / А.Г. Курош. – М.: Наука, 1968. – 427 с.

МОДЕЛИРОВАНИЕ РАБОТЫ ФУНКЦИОНАЛЬНЫХ БЛОКОВ СОЦИАЛЬНОЙ СЕТИ С ЦЕЛЬЮ ИЗУЧЕНИЯ ИХ ЭФФЕКТИВНОСТИ

Д.А. Гончарова

Студент

М.Г. Матвеев

Профессор

Введение

В последнее время все большую и большую распространенность находят различные социальные сетевые сервисы. Под понятием сетевого сервиса обычно подразумевают какой-либо виртуальный продукт, связывающий пользователей с помощью разнообразного программного обеспечения и сети Интернет. В настоящее время социальные сетевые сервисы используются для:

- общения и поддержки социальных контактов
- хранения, поиска, обмена и редактирования различной информации, в т.ч. медиафайлов
- различной творческой деятельности
- определенных задач индивидуального характера

В рамках данной работы проводилось исследование такого сетевого сервиса, как социальная сеть. Однако исследовательский интерес был направлен не на изучение поведения пользователей в системе или какие-либо подобные социальные исследования, а на техническую составляющую самого процесса функционирования.

Работа посвящена анализу функциональных блоков социальной сети на основе их моделирования с целью определения требований к их эффективности.

Результаты данной работы могут быть использованы как начальные значения при проведении нагрузочного тестирования социальной сети как сложной системы [1], а именно: стандартные сценарии использования системы, частота использования тех или иных сценариев, а также, что самое важное, начальные требования к эффективности и ресурсам памяти всех обслуживающих серверов системы.

Средой для разработки модели выбрана SimEvents – библиотека MATLAB /Simulink [2] для моделирования систем с дискретными состояниями, использующая теорию очередей и систем массового обслуживания. Такой выбор оказался не случайным: работа социальной

сети – классический пример системы массового обслуживания. Система массового обслуживания – это система, состоящая из множества обслуживающих приборов (от одного до бесконечности), на вход которой поступают определенные требования, которые в дальнейшем этой системой и реализуются [3].

1. Сбор статистики

Для получения данных об активности пользователей был проведен опрос, в котором участников просили ответить, как часто они совершают определенное действие в социальных сетях. В нем приняли участие 140 человек.

После проведения опроса его результаты были проанализированы и были вычислены средние значения ответов участников, которые приведены ниже (табл. 1).

Таблица 1

Результаты опроса

Логин	0.11656511904761913 раз в час
Изменение информации о себе	6.45077380952381E-4 раз в час
Общее время в соц. сети	4.246428571428571 часов в день
Время в новостной ленте	1.46910714285714 часов в день
Лайки	1.8569642857142854 раз в час
Репосты	0.1579761904761905 раз в час
Просмотр фото	10.969642857142858 раз в час
Просмотр видео	1.7663690476190474 раз в час
Создание постов	0.025858809523809517 раз в час
Редактирование постов	0.00386035085034014 раз в час
Удаление постов	0.00388059523809523 раз в час
Комментирование	0.06619398809523812 раз в час
Редактирование комментариев	0.00609930318877552 раз в час
Удаление комментариев	0.003005595238095238 раз в час
Загрузка фото на сайт	0.060803035714285705 раз в час
Удаление фото	0.00134785714285714 раз в час
Загрузка видео на сайт	0.002211190476190476 раз в час
Удаление видео	4.464285714285714E-5 раз в час
Отправка заявки в друзья	0.010001369047619038 раз в час
Удаление из друзей	0.012749702380952377 раз в час
Принятие заявки в друзья	0.04831970238095239 раз в час
Отклонение заявки в друзья	0.11136863095238099 раз в час

Окончание таблицы 1

Просмотр списка друзей	0.23386071428571437 раз в час
Посещение страниц друзей	0.4527050000000002 раз в час
Поиск по пользователям	0.19089464285714283 раз в час
Отправка сообщений в ЛС	55.95238095238095 раз в час
Отправка сообщений в групповые чаты	27.610714285714284 раз в час
Удаление сообщений	0.003164166666666667 раз в час
Редактирование описания фото	0.00145476190476190 раз в час
Редактирование описания видео	0.007169642857142856 раз в час

Высокая точность обусловлена тем, что в дальнейших расчетах порядок этих чисел значительно возрастет.

2. Состав системы

Система состоит из шести серверов:

– App Services – сервер всех сервисов социальной сети, через него проходят все запросы пользователей.

– Auth Server – сервер аутентификации пользователей. Используется только для входа в систему.

– DB Server – сервер базы данных, в которой расположена вся информация о пользователях и материалах на их страницах, в том числе описания фото и видео. Задействован в большинстве запросов пользователей.

– Chat Server – сервер обмена сообщениями, задействован во всех запросах, связанных с ними.

– Media Server – сервер, на котором хранятся сами фото и видео пользователей. Задействован во всех запросах, в которых требуется загрузить медиа-материалы.

– Search Server – сервер, предназначенный для поиска по базе пользователей.

Для каждого типа запросов был написан список серверов, которые задействованы в его выполнении.

3. Моделирование системы

Алгоритм работы модели:

1. Создается сущность в блоке Time-Based Entity Generator. Период создания рассчитан на основе данных, приведенных в табл. 1.

2. С помощью блока Set Attribute сущности присваивается атрибут с номером типа запроса с определенной вероятностью, которая также рассчитана на основе табл. 1.

3. В зависимости от типа запроса сущности присваивается очередь серверов, которые задействованы в выполнении запроса. Это сделано в блоке Attribute Function.

4. Сущность проходит по всем серверам из очереди (Single Server), на каждом из которых собирается статистика прибывших запросов, потом сущность уничтожается в блоке Entity Sink.

В процессе работы модели симулируется действие системы в течение одной секунды при активности 20 миллионов пользователей для выяснения необходимой эффективности серверов.

4. Результат симуляции

В результате симуляции были получены данные о необходимой эффективности серверов, которые представлены ниже (табл. 2–7).

Таблица 2

Сервер App Services

Тип запроса	Количество в секунду
Логин	650
Изменение информации о себе	6
Подгрузка новостной ленты	54502
Лайк	10539
Репост	942
Просмотр фото	61014
Просмотр видео	9690
Создание постов	154
Редактирование постов	23
Удаление постов	28
Комментирование	379
Редактирование комментариев	41
Удаление комментариев	20
Загрузка фото на сайт	287
Удаление фото	10
Загрузка видео на сайт	11
Отправка заявки в друзья	67
Удаление из друзей	54

Окончание таблицы 2

Тип запроса	Количество в секунду
Принятие заявки в друзья	266
Отклонение заявки в друзья	594
Просмотр списка друзей	1350
Посещение страниц друзей	2509
Поиск по пользователям	1123
Отправка сообщений в ЛС	311635
Отправка сообщений в групповые чаты	153353
Удаление сообщений	16
Редактирование описания фото	9
Редактирование описания видео	40

Таблица 3

Auth Server

Тип запроса	Количество в секунду
Логин	650

Таблица 4

DB Server

Тип запроса	Количество в секунду
Изменение информации о себе	6
Подгрузка новостной ленты	54502
Лайк	10539
Репост	942
Просмотр фото	61014
Просмотр видео	9690
Создание постов	154
Редактирование постов	23
Удаление постов	28
Комментирование	379
Редактирование комментариев	41
Удаление комментариев	20

Окончание таблицы 4

Тип запроса	Количество в секунду
Загрузка фото на сайт	287
Удаление фото	10
Загрузка видео на сайт	11
Отправка заявки в друзья	67
Удаление из друзей	54
Принятие заявки в друзья	266
Отклонение заявки в друзья	594
Просмотр списка друзей	1350
Посещение страниц друзей	2509
Поиск по пользователям	1123
Редактирование описания фото	9
Редактирование описания видео	40

Таблица 5

Chat Server

Тип запроса	Количество в секунду
Отправка сообщений в ЛС	311635
Отправка сообщений в групповые чаты	153353
Удаление сообщений	16

Таблица 6

Media Server

Тип запроса	Количество в секунду
Подгрузка новостной ленты	54502
Просмотр фото	61014
Просмотр видео	9690
Загрузка фото на сайт	287
Удаление фото	10
Загрузка видео на сайт	11
Посещение страниц друзей	2509

Search server

Тип запроса	Количество в секунду
Поиск по пользователям	1123

Заключение

Смоделировав работу системы, мы смогли узнать примерное количество и тип запросов, которые каждый сервер будет обрабатывать в секунду при активности 20 миллионов пользователей. В совокупности получилось:

- App Services – 609312 запросов;
- Auth Server – 650 запросов;
- DB Server – 143658 запросов;
- Chat Server – 465004 запросов;
- Media Server – 128023 запросов;
- Search Server – 1123 запросов.

Эти данные помогут нам предположить, какими техническими характеристиками должны обладать сервера в реальной работе.

Список литературы

1. Файзутдинов Р. Н. Математическое моделирование сложных систем. Лабораторный практикум / Р. Н. Файзутдинов : Казанский гос. ун-т. – Казань: Изд-во Казанского гос. ун-та, 2013г. – 69 стр.
2. Интеллектуальная модель системы массового обслуживания с очередью типа D/D/1 в среде SimEvents (MATLab/SimuLink) [Электронный ресурс]: статья. – Режим доступа: http://model.exponenta.ru/cl_gva_02.html
3. Сирота, А. А. Компьютерное моделирование сложных систем: учеб. пособие / А. А. Сирота: Воронеж, гос. ун-т. – Воронеж: Изд-во Воронеж. гос. ун-та, 2006. – 248 с.

ИСПОЛЬЗОВАНИЕ КЛАСТЕРНОГО АНАЛИЗА В РАСПОЗНАВАНИИ НОТНЫХ ЛИСТОВ

А.Б. Горовой
Студент
В.В. Фертиков
Доцент

Введение

Тема оптического распознавания символов сейчас достаточно популярна, но тема распознавания нотных листов практически не исследована. Среди систем, которые позволяют решать данную задачу, можно выделить следующие:

– PhotoScore [1] – программное обеспечение от компании Neuratron. Обладает достаточно большим функционалом, который основан на использовании собственной системы распознавания OmniScore.

– SharpEye [2] – программное обеспечение от компании visiv. Обладает функционалом, схожим с PhotoScore. Для распознавания музыкальных листов использует библиотеку Liszt.

– Audioversis [3] – открытое программное обеспечение. Его работа основана на выделении элементов нот при помощи различных морфологических преобразований.

Однако, согласно отзывам пользователей [4,5] данные системы не всегда позволяют получить качественный результат.

Исходя из этого, в рамках данной работы были поставлены и выполнены следующие задачи:

1. Определить применимость кластерного анализа к данной задаче.
2. Разработать алгоритм для распознавания элементов на нотных листах.

1. Разработка алгоритма.

Перед непосредственной разработкой алгоритма необходимо выявить основные проблемы данной предметной области, установить границы работы алгоритма и выделить основные этапы работы алгоритма.

Основные проблемы, связанные с распознаванием нотных листов в сравнении с традиционным распознаванием текста:

– На разных нотных листах могут использоваться различные шрифты, что усложняет распознавание элементов.

– На нотных листах присутствует множество различных символов и их комбинаций, тогда как количество символов в обычном тексте заранее известно и строго ограничено.

– В случае распознавания нотных листов необходимо не только распознать сами символы, но и проанализировать их расположение на нотном стане, что добавляет сложности алгоритмам распознавания.

Из-за описанных выше проблем существует достаточно мало систем распознавания нотных листов, а исследовательские работы на эту тему достаточно сложно найти.

При разработке алгоритма были сделаны следующие предположения:

– На вход алгоритму подается отсканированное изображение, на котором присутствует только нотный лист, без лишнего фона.

– Изображение не должно быть искривлено или деформировано, иначе алгоритм может выдавать некорректные результаты.

– Предполагается, что изображение представляет из себя черные символы на белом фоне.

На основе данной информации и условий разработан алгоритм распознавания элементов на основе кластерного анализа. В нем можно выделить следующие основные шаги: выделение элементов изображения и кластеризация выделенных элементов.

2. Выделение элементов листа.

На первом шаге необходимо выделить препарат для кластеризации с исходного изображения. Для иллюстрации приведем фрагмент изображения нотного листа (рис. 1).

14.

Adagio sostenuto.
Si deve suonare tutto questo pezzo delicatissimamente e senza sordini.
sempre pp e senza sordini simile

Рис. 1. Исходное изображение

Для изображения выполняются следующие преобразования:

1. Преобразование изображения в негатив. Операции могут выполняться и с позитивным изображением, но для простоты рассуждений используется негативное представление, так как в этом случае значащим точкам соответствует большее значение, чем фону.

2. Разделение листа на отдельные нотные станы. Для разделения листа используется разделение по порогу яркости для каждой линии. Порог яркости вычисляется на основе средней яркости изображения.

3. Выделение на каждом нотном стане линий и вычисление расстояния между ними. Для этого используется преобразование Хафа и поиск достаточно удаленных друг от друга пиков.

В результате получается изображение нотного стана, как на рис. 2.



Рис. 2. Изображение нотного стана

С использованием определенного значения расстояния между линиями выполняются следующие операции с изображением нотного стана:

- Выделение нотного стана.
- Выделение длинных вертикальных линий на изображении. Такие линии обычно являются соединяющими аккорды или разделяющими такты.
- Выделение элементов, объединяющих ноты одной длины.
- Выделение коротких линий, которые соединяют ноты с линиями, в свою очередь их объединяющими их в единую смысловую конструкцию.

Для всех этих операций используются морфологические преобразования закрытия и открытия со структурным элементом похожим на объект, который нужно выделить. Так как преобразования проводятся над негативным изображением, то для удаления лишних элементов следует производить операцию открытия. Она заключается в сужении изображения структурным элементом и последующим расширении им же. При сужении удаляются элементы изображения, которые меньше структурного элемента. Если элемент не был удален, то его исходная форма (или близко к ней приближенная) восстанавливается при операции расширения. Поэтому с помощью такого преобразования можно выделить нужные элементы на изображении.

После нахождения этих элементов они убираются с изображения, чтобы остались только значащие элементы – ноты, диезы, бемоли, ключи и прочее.

Также с изображения убираются линии нотного стана. Однако, при этом связность компонентов может быть нарушена, поэтому после удаления применяется операция закрытия. В результате получается

очищенное изображение, на котором есть только значащие элементы, как на рис. 3.

На очищенном изображении происходит поиск связанных компонент размером больше, чем половина расстояния между линиями на нотном стане и не превосходящем четыре расстояния между линиями на нотном стане. Исходя из границ связанных компонент с исходного изображения набирается препарат для кластерного анализа.



Рис. 3. Изображение, подготовленное к выделению связанных компонент

3. Кластерный анализ.

Как было сказано ранее, для распознавания элементов на нотном листе используется кластерный анализ. Это связано с необходимостью распознавать символы, количество и шрифт которых заранее неизвестны. Кластерный анализ позволяет собрать похожие элементы в кластеры и сравнить их с эталонными. Это можно делать как на основании принадлежности к кластеру, так и для каждого элемента в отдельности. Эталоны также можно получить на основе кластерного анализа. Для этого необходимо выделить элементы на нотном листе, провести их кластеризацию и разметить, а при необходимости и отредактировать, полученные кластеры.

Для проведения кластеризации необходимо определить алгоритм, с помощью которого планируется осуществлять разбиение на кластеры. Наиболее популярными алгоритмами кластеризации являются метод k-means [6] и иерархический метод [7]. Алгоритм k-means требует заранее известного количества кластеров, поэтому данный метод плохо применим к этой задаче, исходя из ее условия. Иерархический метод подходит к данной задаче гораздо лучше, он позволяет проводить кластеризацию для заранее неизвестного количества кластеров. Это связано с тем, что алгоритм составляет дендрограмму вплоть до полного объединения кластеров и пользователь (либо программа в автоматическом режиме) сам может выбрать, на каком уровне производить разделение кластеров. Помимо количества кластеров, существует два варианта, позволяющих производить разделение кластеров:

- Уровень дендрограммы.
- Коэффициент несогласованности (inconsistency coefficient).

Уровень дендрограммы близок по смыслу к количеству кластеров и не может быть точно определен. В отличие от него, коэффициент

несогласованности является параметром, обратным прочности связи между кластерами. Соответственно, если он достаточно велик, то кластеры можно разделить. Поэтому он удобен для формирования кластеров. На рис. 4 представлен пример кластера, сформированного таким образом. Черные квадраты в верхнем левом и нижнем правом углу не являются значащими элементами, а просто дополняют отображение кластера до квадрата.

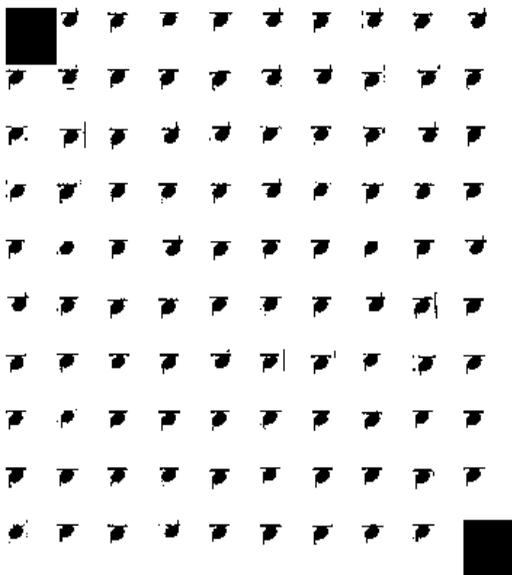


Рис. 4. Пример кластера

Таким образом, иерархическая кластеризация с разделением по порогу, основанном на коэффициенте несогласованности, подходит для создания кластеров и базы эталонов на их основе

Заключение

В данной статье описан алгоритм выделения препарата кластеризации, кластеризации и формирования базы эталонов. В результате было показано, что кластерный анализ подходит для распознавания нотных листов. Данный алгоритм был реализован в среде MATLAB и в дальнейшем может быть использован для распознавания конкретных нот и формирования результата в виде, который может использоваться другими средствами для редактирования и воспроизведения музыкальных произведений.

Список литературы.

1. Сайт PhotoScore [Электронный ресурс] : сайт системы OMR. – Режим доступа : <http://www.neuratron.com/photoscore.htm>
2. Сайт SharpEye [Электронный ресурс] : сайт системы OMR. – Режим доступа : <http://visiv.co.uk/>
3. Audioversis [Электронный ресурс] : репозиторий. – Режим доступа : <https://github.com/Audiveris/audiveris>
4. Обзор SharpEye [Электронный ресурс] : сайт посвященный ПО для музыкантов. – Режим доступа : <http://www.notovodstvo.ru/library/omr/#fn1>
5. Отзыв о Sibelius [Электронный ресурс] : форум. – Режим доступа : <http://www.sibelius.com/cgi-bin/helpcenter/chat/chat.pl?com=thread&start=570559&groupid=3&&guest=1>
6. Обзор алгоритма k-means [Электронный ресурс] : форум. – Режим доступа : <https://habrahabr.ru/post/67078/>
7. Описание иерархической кластеризации и ее применение в среде MATLAB [Электронный ресурс] : документация к ПО компании MathWorks. – Режим доступа : <https://www.mathworks.com/help/stats/hierarchical-clustering.html>

АНАЛИЗ ВЛИЯНИЯ В СОЦИАЛЬНОЙ СЕТИ НА ОСНОВЕ ВЫДЕЛЕНИЯ СЕТЕВЫХ СООБЩЕСТВ

М.В. Данилова

Студент

А.В. Сычев

Доцент

Введение

Многие люди при выборе товара полагают, что они совершают собственный выбор путем анализа советов пользователей из социальных сетей. Однако подобное “сарафанное радио” может быть результатом мощной пиар кампании и активной деятельности маркетологов. Подобная скрытая реклама представляет собой целое направление в маркетинге, его называют также сарафанным маркетингом. Сарафанный маркетинг – это маркетинговая технология, позволяющая спровоцировать позитивные разговоры о бренде в среде потенциальных потребителей [1].

С развитием социальных сетей деятельность маркетологов данного направления значительно упростилась, т. к. именно социальные сети представляют собой мощную платформу для распространения информации. Было разработано специальное направление, занимающееся

развитием различных кампаний через социальные сети – Social Media Marketing. Оно отвечает за различную деятельность в области маркетинга, начиная от продажи товаров, заканчивая брендингом, т. е. кампанией по увеличению узнаваемости бренда. Вот несколько постулатов, которые позволят провести успешную кампанию: запустить информацию о товаре через социальные сети; привлечь на свою сторону экспертов или лидеров мнений; заставить всех говорить о товаре. Т.е. ключевым этапом является привлечение к кампании лидеров мнений, т.е. пользователей сети, способных повлиять на аудиторию. Однако обрабатывать данные вручную – довольно сложный процесс.

1. Задачи и методы исследования

В связи с этим, актуальность данной статьи обусловлена необходимостью автоматизации поиска лидеров мнений в социальной сети для проведения различных маркетинговых кампаний.

Соответственно, целью является поиск всех ключевых пользователей, которые обладают авторитетом среди своего окружения, что проявляется на основе количества связей между ними. Исходя из цели исследования, задачами являются:

1. Извлечь информацию из социальной сети: названия аккаунтов, которые взаимодействовали в рамках заявленной тематики.
2. Построить социальный граф на основе связей пользователей социальной сети.
3. Разбить социальный граф на сообщества пользователей.
4. Найти лидеров мнений.
5. Определить, что работа только с лидерами мнений более выгодна.

Исследование проведено на основе данных социальной сети Twitter. За основу исследования взята тема о недавно прошедшем конкурсе Евровидение, поэтому информация была извлечена из Twitter с помощью ключевого слова «Евровидение». Извлечение и анализ данных проведены при помощи надстройки Excel – приложения NodeXL.

Для решения поставленных задач был построен социальный граф, в котором узлы являются пользователями, а ребра – связями между ними. Социальный граф является одним из основных способов решения задач Social Mining. Разбиение графа на сообщества проведено при помощи алгоритм Клозета-Ньюмана-Мура, в результате которого произошло разбиение графа на части на основе плотности связей. При помощи данного алгоритма вычисляются связи между узлами, принадлежащими разным кластерам, и узлами одного кластера. Сообщество можно охарактеризовать, как часть графа, в которой средняя плотность между узлами превышает плотность связей между остальными подграфами. Алгоритм Клозета-Ньюмана-Мура вычисляется по формуле [2]:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j), \quad (1)$$

где A — матрица смежности графа, A_{ij} — (i, j) элемент матрицы, d_i — степень i вершины графа, C_i — метка вершины (номер сообщества, к которому относится вершина), m — общее количество ребер в графе. $\delta(C_i, C_j)$ — дельта-функция: равна единице, если $C_i = C_j$, иначе нулю.

В NodeXL разбиение социального графа на сообщества произведено через функцию «Group by cluster», которая предполагает применение алгоритма Клозета-Ньюмана-Мура. В результате все узлы графа скучковались по сообществам на основе плотности связей. Также для более удобного визуального анализа, на графе, напротив всех узлов, подписано количество входящих связей соответствующего узла, т.е. это означает, количество пользователей, которые каким либо образом ссылались на соответствующего пользователя в рамках определенной тематики. Это можно задать через дополнительные настройки NodeXL «Autofilling Columns». Таким образом, узлы, которые имеют больше всего связей, разместились в центре сообществ и они отличаются по размеру от других узлов. После были проанализированы сформированные графы.

2. Результаты эксперимента

В некоторых сообществах вершины были выстроены вокруг одного центрального узла (рис. 1). Это может говорить о слабой связанности пользователей сообщества между собой. На рисунке видно, что сообщество состоит из 32 пользователей, основная связь между пользователя осуществляется через центральный узел.

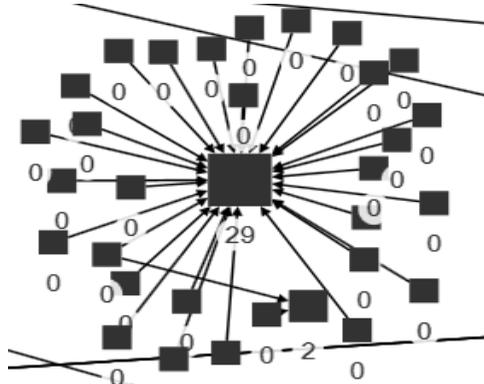


Рис. 1. Сетевое сообщество пользователей социальной сети Twitter, в котором участники сообщества связаны только через центральный узел

Найдены сообщества, в которых находятся несколько лидеров мнений, которые также соединены между собой (рис. 2). По изображению видно, что данный подграф содержит три узла, обладающие большим количеством входящих связей, и так как они соединены между собой, информация по сети может распространяться более консолидировано между ее участниками.

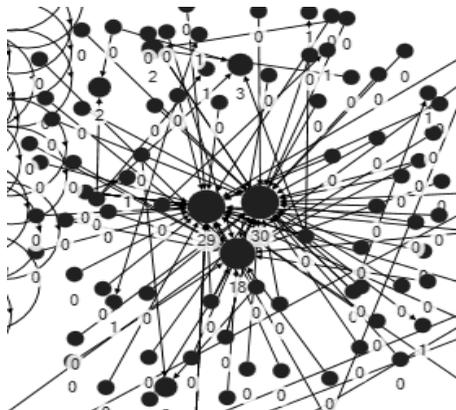


Рис. 2. Консолидация сообщества через лидеров мнения через NodeXL

Также выявлены сообщества, представители которых связаны с представителями других сообществ (рис. 3). Однако подобная связь пользователей не гарантирует распространение информации между представителями нескольких сообществ консолидировано, что видно по рисунку, поскольку ключевой пользователь из сообщества, которое отображено синим цветом связан с представителем оранжевого сообщества, но так как последний пользователь не обладает входящими связями, данную связь можно считать не перспективной.

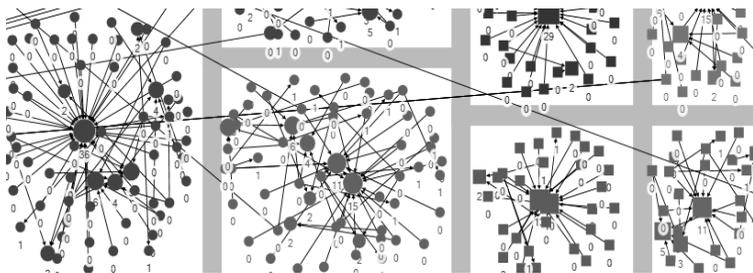


Рис. 3. Распространение информации между пользователями, объединенными в различные сообщества

В результате, выявлено, что среди 3000 извлеченных аккаунтов социальной сети Twitter, которые обсуждали тему Евровидение, только от 426 пользователей распространяется информация как минимум на два других аккаунта, что говорит о слабой связанности представителей, составивших социальный граф. Информация по сети, в основном, распространяется благодаря деятельности лидеров мнений. Всего в данной сети содержится 7 пользователей, на которые ссылаются больше 20 человек. Окружение влиятельного пользователя совершает такие же действия как и он сам, например, если один из лидеров мнений опубликует твит на странице своего аккаунта, есть высокая вероятность, что его окружение сделает ре-твит данного сообщения. Исходя из этого, показано, что работать только с лидерами мнений более выгодно, чем работать с целевой группой. Также выявлено, что вероятность консолидированного распространения информации по сети выше, если лидеры мнения связаны между собой. Это может гарантировать, что информация не будет противоречить друг другу, а также обеспечит более высокую скорость распространения.

Выводы

Итак, при помощи метода выделения сетевых сообществ найдены лидеры мнений сети, которые могут быть привлечены к проведению маркетинговых кампаний. Данный метод позволяет визуально проанализировать данные, извлеченные из социальной сети, найти фигуры, играющие ключевую роль в распространении информации, т.к. лидеры мнений в группах чаще всего являются центральными узлами в сообществе и обладают множеством связей

Также исследование показало, что не все лидеры мнений играют одинаковую роль в процессе распространения информации, и уже здесь большую роль будет играть способ взаимодействия маркетологов с ними. В любом случае, данный способ позволяет снизить затраты на проведение маркетинговой кампании.

Список литературы

1. Что такое сарафанный маркетинг [Электронный ресурс]. – Режим доступа : <https://adindex.ru/publication/tools/2012/09/5/92949.phtml>
2. Гусарова, Н. Ф. Анализ социальных сетей. Основные понятия и метрики : учебное пособие / Н. Ф. Гусарова. – СПб. : Университет ИТМО, 2016. – 67 с.

ОПРЕДЕЛЕНИЕ РЕГЛАМЕНТНЫХ ПАРАМЕТРОВ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ НА ОСНОВЕ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ

Д.С. Дедов
Студент
М.Г. Матвеев
Профессор

Введение

В хозяйственно-финансовой деятельности организации постоянно присутствует необходимость выбора одного из нескольких возможных вариантов решения. Каждое управленческое решение имеет свой конкретный результат, поэтому целью управленческой деятельности является нахождение таких методов, средств и инструментов, которые могли бы способствовать достижению оптимального результата и в конкретных условиях, и в конкретных обстоятельствах.

Одним из методов принятия управленческих решений является алгоритмический. Данный метод основан на исследовании операций, под которым понимается комплекс научно-обоснованных методов, применяемых для поиска и выбора решений, наиболее эффективных и отвечающих заданным критериям. С помощью исследования операций находятся решения, в частности, и в сфере управления

Научная новизна данной работы заключается в том, что в ходе исследования были рассмотрены особенности процесса поиска и принятия управленческих решений, применены на практике основные методы алгоритмического процесса принятия решения и в результате исследования создана конкретная математическая модель обеспечения сервиса в ИТ-области.

В ходе работы рассмотрен типовой непрерывный процесс поступления заявок от клиентов на решение разнообразных проблем, устранения ошибок или на изменение настроек в пользовательской системе. На каждую заявку необходимо выделить ресурс для ее решения. Время решения заявки строго регламентировано. Процесс поступления и закрытия заявок можно считать непрерывным, в то же время поступление заявок носит случайный характер и не зависит от владельца процесса. Таким образом, потребность на заявках одного вида возрастает, а на

другого не изменяется или падает. Таким образом владелец процесса сталкивается с проблемой оптимального размещения персонала в зависимости от выполняемых работ и их количества. Необходимо выбрать число каналов обслуживания (консультантов), определить их занятость, обеспечить минимальную очередь. Противоречие заключается в том, что процент занятости сотрудников должен быть максимальным, а вероятность отказа в сервисе минимальной.

1. Цель работы

Целью данной работы является рассмотрение методологии принятия управленческих решений, оценка параметров договора на техническую поддержку, математическое моделирование системы технической поддержки в рамках систем массового обслуживания (СМО) и расчет характеристик эффективности функционирования СМО.

Предметом исследования является анализ теоретических и практических вопросов, связанных с оценкой, выбором, подготовкой и организацией процесса сопровождения системы клиента.

Практическая значимость исследования заключается в том, что результаты исследования могут служить пособием для принятия управленческих решений при заключении компанией нового договора с клиентом. Проведенные работы по определению целесообразного отношения между количеством заявок, поступающих в систему, и количеством консультантов, обслуживающих их, помогут набрать оптимальную команду на проект, а рассчитанные вспомогательные критерии вести правильную кадровую политику, а именно эффективность вовлеченности консультанта в проект.

2. Система обслуживания предприятия

Любая деятельность предприятия представляет собой сложную систему. Для успешного ведения деятельности необходимо внедрение системы управления предприятия. После внедрения следует этап ее эксплуатации – поддержка. Данный этап необходим для того, чтобы верно выстроить взаимоотношения между службой ИТ и конечными пользователями.

Выделяют 3 уровня технической поддержки.

Первый уровень состоит из одного или нескольких сотрудников службы ИТ, выполняющих роль консультантов. Консультант опрашивает пользователя и пытается решить его проблему, объяснив, например, что необходимо заполнить обязательные поля перед сохранением информации. Если пользователю необходимо добавить какие-то дополнительные возможности к существующему функционалу, то он тоже звонит в службу поддержки и консультант определяет, есть ли возможность настроить

систему (именно настроить, не внося изменений в программный код). Если возможности нет, то вопрос передается на **второй уровень поддержки** [1].

Второй уровень состоит из консультанта высокой квалификации и разработчиков, которые в свою очередь определяют возможность настройки и доработки системы под запрос пользователя. Когда вопрос не может быть решен силами второго уровня, либо требует существенных изменений системы вплоть до внедрения нового модуля или разработки новой функциональности системы, запрос передается на третий уровень.

Третий уровень включает высококвалифицированных специалистов, выполняющих роли консультантов, архитекторов системы и разработчиков.

3. Понятие Service Level Agreement

Менеджеры крупных компаний стремятся подобрать наиболее эффективное решение для управления масштабной и постоянно развивающейся информационной системой. Одним из самых эффективных способов для решения этой задачи считается создание системы эксплуатации, интегрирующей программные продукты и формализованное описание процессов.

Вводится персональная ответственность за эффективность и результативность процессов и отдельных действий, что позволяет не только отслеживать состояние, но и управлять им; Реализация данного подхода к созданию системы управления возможна при создании системы управления эксплуатацией, построенной на принципах соглашения об уровне услуг (SLA) [2].

SLA – это полноценный контракт, заключаемый между провайдером услуг и потребителем, в котором находят отражение и финансовые вопросы.

В таком соглашении может содержаться детальное описание предоставляемого сервиса, в том числе перечень параметров качества, методов и средств их контроля, времени отклика поставщика на запрос от потребителя, а также штрафные санкции за нарушение этого соглашения. Для того, чтобы соблюсти принципы SLA, поставщик услуг в свою очередь заключает операционное соглашение об уровне услуг (OLA) с другими внутренними подразделениями от которых зависит качество предоставления услуг.

В идеале, SLA определяется как особый сервис. Это позволяет сконфигурировать аппаратное и программное обеспечение для максимизации способности удовлетворять SLA. Параметры качества услуги, указанные в SLA, должны быть измеримыми, то есть представимыми в виде числовых метрик. Различные примеры соглашений SLA приведены в описаниях стандартов ITIL и COBIT, где также даны

развернутые рекомендации по оценке ключевых показателей эффективности (KPI) при анализе работы со SLA.

4. Системы массового обслуживания

Системы, в которых, с одной стороны, возникают массовые запросы (требования) на выполнение каких-либо видов услуг, а с другой стороны, происходит удовлетворение этих запросов, называются системами массового обслуживания (СМО).

Система массового обслуживания включает следующие элементы: источник требований, входящий поток требований, очередь, обслуживающее устройство (обслуживающий аппарат, канал обслуживания), выходящий поток требований [3].

Системы массового обслуживания классифицируют, в частности, по условиям ожидания требования начала обслуживания. В соответствии с этим признаком системы подразделяются на следующие виды:

- системы массового обслуживания с потерями (отказами);
- системы массового обслуживания с ожиданием;
- системы массового обслуживания с ограниченной длиной очереди.

Для данной задачи использовалась СМО без отказа и для вычисления КПИ1 и КПИ2 использовалась формулы СМО с ограниченным временем ожидания.

Рассмотрим СМО с ограниченным временем ожидания. Заявка, ставшая в очередь, уже не покидает ее и дожидается обслуживания.

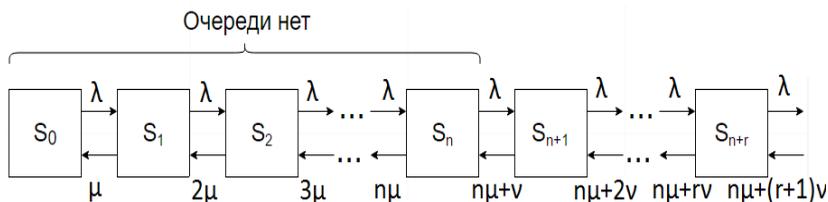


Рис. 1. СМО с ограниченным временем ожидания

Предполагается, что имеется n -канальная СМО с ожиданием, в которой число мест в очереди не ограничено, но время пребывания заявки в очереди ограничено некоторым случайным сроком $T_{оч}$ и средним значением $\bar{t}_{оч}$ таким образом, на каждую заявку, стоящую в очереди, действует как бы «поток уходов» с интенсивностью, рассчитываемой по следующей формуле [4]:

$$v = \frac{1}{t_{oc}}$$

Процесс протекающий в СМО марковский, так как поток пуассоновский. Найдем для него вероятности состояний. Нумеруются состояния системы по числу заявок, связанных с системой — как обслуживаемых, так и стоящих в очереди:

- S_0 – очереди нет, все каналы свободны;
- S_1 – очереди нет, занят один канал;
- S_2 – очереди нет, заняты два канала;
- S_n – очереди нет, заняты все n каналов;
- S_{n+1} – все n каналов заняты, одна заявка в очереди;
- S_{n+r} – все n каналов заняты, r заявок в очереди.

Граф состояний системы показан на рис.1.

Разметим этот граф, т. е. проставим у стрелок соответствующие интенсивности. Снова, как и раньше, у всех стрелок, ведущих слева направо, будет стоять интенсивность потока заявок λ . Для состояний без очереди у стрелок, ведущих из них справа налево, будет, как и раньше, стоять суммарная интенсивность потока обслуживаний всех занятых каналов. Что касается состояний с очередью, то у стрелок, ведущих из них справа налево будет стоять суммарная интенсивность потока обслуживаний всех n каналов $n\mu$ плюс соответствующая интенсивность потока уходов из очереди. Если в очереди стоят r заявок, то суммарная интенсивность потока уходов будет равна $r\nu$.

Как видно из графа, перед нами опять схема гибели и размножения; применяя общие выражения для предельных вероятностей состояний в этой схеме, напишем или, вводя обозначения [4]:

$$\rho = \frac{\lambda}{\mu}, \beta = \frac{\nu}{\mu},$$

$$p_0 = \left\{ 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^n}{n!} \left[\frac{\rho}{n+\beta} + \frac{\rho^2}{(n+\beta)(n+2\beta)} + \dots \right. \right. \\ \left. \left. \dots + \frac{\rho^r}{(n+\beta)(n+2\beta)\dots(n+r\beta)} \right] \right\}^{-1}$$

$$p_1 = \frac{\rho}{1!} p_0,$$

$$p_2 = \frac{\rho^2}{2!} p_0,$$

.....

$$p_n = \frac{\rho^n}{n!} p_0,$$

$$p_{n+1} = \frac{\rho^n}{n!} \frac{\rho}{n+\beta} p_0,$$

$$p_{n+2} = \frac{\rho^n}{n!} \frac{\rho^2}{(n+\beta)(n+2\beta)} p_0,$$

.....

$$p_{n+r} = \frac{\rho^n}{n!} \frac{\rho^r}{(n+\beta)(n+2\beta)\dots(n+r\beta)} p_0.$$

Отметим некоторые особенности рассмотренной СМО с «нетерпеливыми» заявками по сравнению с ранее рассмотренной СМО с «терпеливыми» заявками. Если длина очереди не ограничена заранее никаким числом и заявки «терпеливы» (не уходят из очереди), то стационарный предельный режим существует только в случае $\rho < n$ (при $\rho \geq n$ соответствующая бесконечная геометрическая прогрессия расходится, что физически соответствует неограниченному росту очереди при $t \rightarrow \infty$). Напротив, в СМО с «нетерпеливыми» заявками, уходящими рано или поздно из очереди, установившийся режим обслуживания при $t \rightarrow \infty$ достигается всегда, независимо от приведенной интенсивности потока заявок ρ [4].

Относительную пропускную способность q такой СМО можно подсчитать следующим образом. Обслужены будут все заявки, кроме тех, которые уйдут из очереди досрочно. Подсчитаем, какое в среднем число

заявок покидает очередь досрочно. Для этого вычислим среднее число заявок в очереди [4]:

$$\bar{r} = 1 \cdot p_{n+1} + 2 \cdot p_{n+2} + \dots + r \cdot p_{n+r} + \dots$$

На каждую из этих заявок действует «поток уходов» с интенсивностью ν . Значит, из среднего числа заявок в очереди в среднем будет уходить, не дождавшись обслуживания, \bar{r} заявок в единицу времени; всего в единицу времени в среднем будет обслужено заявок [4]

$$A = \lambda - \nu \bar{r}.$$

Относительная пропускная способность СМО рассчитывается как [4]:

$$q = \frac{A}{\lambda} = 1 - \frac{\nu}{\lambda} \bar{r}.$$

Среднее число занятых каналов \bar{z} по-прежнему получим, деля абсолютную пропускную способность на μ

$$\bar{z} = \frac{A}{\mu} = \rho - \beta \bar{r}.$$

5. Практическая реализация

Первоначальный экран программы имеет 5 окон для заполнения, как показано на рисунке 2:

- Интенсивность заявок (количество заявок, приходящих в интервал времени);
- Интенсивность обслуживания (количество заявок, обрабатываемых одним консультантом в интервал времени);
- Стоимость консультанта (сумма, которую необходимо платить консультанту для работы в проекте);
- Время ожидания в очереди (время, в течение которого заявка должна быть взята в обработку после создания пользователем);
- Требуемое время обслуживания (необходимое время для обслуживания заявки).

После заполнения данных полей, программа выводит таблицу выходящих параметров для каждого консультанта. В таблице видно среднее число заявок в очереди, число просроченных заявок, КПИ1, КПИ2, средняя загрузка и суммарная стоимость под каждого консультанта.

Интенсивность заявок, час	<input type="text" value="10"/>	Время ожидания в очереди, час	<input type="text" value="0.166666666"/>
Интенсивность обслуживания, час	<input type="text" value="3"/>	Требуемое время обслуживания, час	<input type="text" value="0.5"/>
Стоимость консультанта, руб	<input type="text" value="20 000"/>		

	Кол-во конс-ов	Средн. число заявок в очереди	Число просроченных заявок	КПИ1, %	КПИ2, %	Средняя загрузка, %	Сумм. стоимость, руб
	2	,883	5,301	46,992	27,703	166,667	40000
	3	,502	3,01	69,898	73,035	111,111	60000
	4	,273	1,641	83,594	88,069	83,333	80000
	5	,135	,807	91,927	94,887	66,667	100000
	6	,059	,355	96,454	97,988	55,556	120000
	7	,023	,139	98,607	99,277	47,619	140000
	8	,008	,049	99,507	99,761	41,667	160000

Рис. 2. Интерфейс программы

Заключение

В ходе данной работы рассмотрена методология принятия управленческих решений, произведена оценка параметров договора на техническую поддержку, выполнено математическое моделирование системы технической поддержки в рамках СМО и, как результат исследования, был создан программный продукт для оценки параметров договора на техническую поддержку с помощью системы массового обслуживания, а также проанализировано изменение критериев эффективности от количества консультантов, приведенной интенсивности потока заявок обслуживания и поступления в систему и времени обслуживания.

Список литературы

1. Смирнов Э.А. Разработка управленческих решений: Учебник для вузов / Э.А. Смирнов. – М.: ЮНИТИ – ДАНА, 2000. – 271 с.
2. Галушко В.П. Управленческие решения и их формализация / В.П. Галушко. – Киев : ВШЭ, 1983. – 416 с.
3. Вентцель Е.С. Исследование операций / Е.С. Вентцель. – М. : «Советское радио», 1972. – 552 с.
4. Овчаров Л.А. Прикладные задачи теории массового обслуживания / Л.А. Овчаров. – М. : Машиностроение, 1969. – 245 с.

ПЕРЕДАЧА ДАННЫХ МЕЖДУ АЛГОРИТМАМИ В ЭКСПЕРИМЕНТАХ С ВЛОЖЕННЫМИ ЦИКЛАМИ

А.И. Денисенко

Студент

П.С. Лысачев

Старший преподаватель

Введение

При исследовании влияния хозяйственной деятельности человека на характеристики речного стока, параметры водохранилищ и режим водохозяйственных систем Красовым В.Д. был разработан ряд алгоритмов [1, 2], в которых требуется обрабатывать различные массивы данных с целью нахождения различных статистических характеристик и формирования на их основе новых массивов данных.

С целью унификации всех алгоритмов была разработана единая система для проведения подобных статистических экспериментов, позволяющая единообразно работать с анализируемыми данными, применяемыми алгоритмами и получаемыми отчетами. Данная система позволяет с использованием визуального конструирования создавать модели систем, состоящих из узлов различных типов (узлов инициализации, узлов алгоритмов, узлов циклов). Входные и выходные данные представлялись в виде множеств таблиц, которые передавались между всеми узлами системы. Подобный подход имел ряд проблем, основными из которых являются следующие:

– Большое использование оперативной памяти. Данная проблема связана с тем, что каждый узел, завершив свою работу, передавал всем последующим узлам массив таблиц, содержащий копии всех таблиц, пришедших ему на вход, а также все таблицы, созданные в процессе выполнения узла. При больших объемах таблиц это приводило к резкому возрастанию количества используемой системой оперативной памяти.

– Сложность работы с параметрами для алгоритмов. В данном подходе все данные – таблицы, а, следовательно, для того, чтобы на вход алгоритма подать значения отдельных параметров, нужно использовать дополнительную таблицу типа «ключ-значение» для хранения в ней параметров.

– Жесткая привязка к названиям таблиц. Проблема состоит в том, что каждый алгоритм должен знать, как называется таблица, содержащая необходимые входные данные. При этом она создается в процессе

выполнения другого алгоритма, где ей и дается определённое имя. То есть для работы одного алгоритма необходимо знать детали реализации другого алгоритма.

– Отсутствие областей видимости. Каждый алгоритм имеет доступ ко всем данным, которые поступали на вход всех предыдущих узлов алгоритмов. Помимо описанной выше проблемы с использованием памяти, это могло привести к коллизии данных (при использовании одинакового названия таблиц), из-за чего работа алгоритма могла быть некорректной.

Решением данных проблем является разработка нового механизма передачи данных между алгоритмами, в рамках которого каждый алгоритм будет иметь множество входов и множество выходов, соединённых связями, показывающими потоки данных между ними. Данные могут быть представлены в виде различных структур и иметь различные типы элементов. Для понимания подобного принципа передачи данных между узлами удобно рассмотреть математическое описание взаимодействия элементов системы.

1. Математическое описание взаимодействия элементов системы

Рассмотрим систему S , состоящую из элементов F_1, \dots, F_N [3].

Обозначим входные данные элемента $F_i, i = \overline{1, N}$ вектором

$$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_{n_i}^{(i)}). \quad (1)$$

Выходные данные представим вектором

$$y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \dots, y_{m_i}^{(i)}). \quad (2)$$

Вход элемента F_i представляет собой совокупность $X^{(i)}$ из n_i входных портов $X_j^{(i)}, j = \overline{1, n_i}$. Аналогично выход элемента F_i – совокупность $Y^{(i)}$ из m_i выходных портов $Y_j^{(i)}, j = \overline{1, m_i}$. Входной порт $X_j^{(i)}$ выделен только для элемента $x_j^{(i)}$ вектора входных данных, а выходной порт $Y_j^{(i)}$ – только для элемента $y_j^{(i)}$ вектора выходных данных.

Каждый элемент системы является функциональным преобразователем информации, реализующим отображение вида

$$\Phi_i : X_1^{(i)} \times \dots \times X_{n_i}^{(i)} \rightarrow Y_1^{(i)} \times \dots \times Y_{m_i}^{(i)}, i = \overline{1, N}, \quad (3)$$

то есть выполняющим преобразование

$$y^{(i)} = \Phi_i(x^{(i)}). \quad (4)$$

Следовательно, каждый элемент описывается кортежем

$$F_i = \langle X^{(i)}, Y^{(i)}, \Phi_i \rangle. \quad (5)$$

Выходные порты элемента F_i соединяются с входными портами элемента $F_j, j \neq i$ с помощью связей, представляющих собой направление передачи данных.

Для исключения дублирования выходной информации и неоднозначности входной информации вводится следующее ограничение: входной порт может быть связан только с одним выходным портом, а выходной порт может быть связан с любым количеством входных портов при условии, что входной порт не связан с каким-либо другим выходным портом.

Рассмотрим теперь случай, когда система S декомпозируется на подсистемы $S_k, k = \overline{1, N}$. Так как подсистема является элементом системы, то она должна иметь совокупность входных $X^{(k)}$ и выходных $Y^{(k)}$ портов. С другой стороны она является системой и содержит элементы $F_i^{(k)}, i = \overline{1, N_k}$, каждый из которых в соответствии с (5) описывается тройкой $\langle X^{(k,i)}, Y^{(k,i)}, \Phi_{k,i} \rangle$. Таким образом, система должна учитывать и связи на уровне подсистемы, и связи на уровне системы в целом. Для этого на уровне подсистем вводятся дополнительные входные $X^{(k,0)}$ и выходные $Y^{(k,0)}$ порты и устанавливается взаимно однозначное соответствие между ними и входными и выходными портами подсистемы S_k . В результате получаем пары входных портов

$$(X_j^{(k)}, X_j^{(k,0)}), j = \overline{1, n_k} \quad (6)$$

и выходных портов

$$(Y_j^{(k)}, Y_j^{(k,0)}), j = \overline{1, m_k}. \quad (7)$$

Рассмотрим пример подобного представления системы (рис. 1). На нем представлена математическая схема системы, состоящей из трех подсистем S_1, S_2, S_3 . Входы и выходы подсистем связаны между собой следующим образом: $Y_1^{(1)}$ с $X_1^{(2)}$ и $X_1^{(3)}$, $Y_2^{(1)}$ с $X_2^{(2)}$ и $Y_1^{(2)}$ с $X_2^{(3)}$. Порт $X_1^{(1)}$ получает информацию из внешней среды от порта $X_1^{(0)}$, а порт $Y_1^{(3)}$ отдает информацию во внешнюю среду через порт $Y_1^{(0)}$.

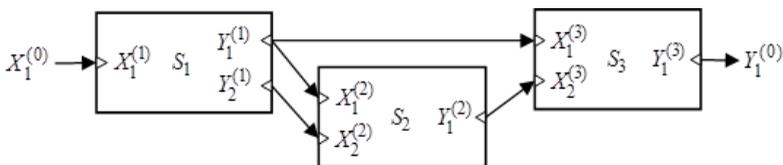


Рис. 1. Пример системы S , имеющей три подсистемы

Рассмотрим теперь, например, декомпозицию подсистемы S_2 (рис. 2), состоящей из трёх элементов $F_1^{(2)}, F_2^{(2)}, F_3^{(2)}$. Порты элементов связаны между собой следующим образом: $Y_1^{(2,1)}$ с $X_1^{(2,3)}$ и $Y_1^{(2,2)}$ с $X_2^{(2,3)}$. Порты $X_1^{(2,1)}$ и $X_1^{(2,2)}$ получают данные от порта $X_1^{(2,0)}$, $X_2^{(2,2)}$ – от $X_2^{(2,0)}$, порт $Y_1^{(2,3)}$ передает данные порту $Y_1^{(2,0)}$. Порты $X_1^{(2,0)}$, $X_2^{(2,0)}$, $Y_1^{(2,0)}$ по отношению к подсистеме S_2 являются внешними и соответствуют портам $X_1^{(2)}$, $X_2^{(2)}$, $Y_1^{(2)}$ подсистемы при рассмотрении на уровне системы. То есть пары портов $(X_1^{(2)}, X_1^{(2,0)})$, $(X_2^{(2)}, X_2^{(2,0)})$, $(Y_1^{(2)}, Y_1^{(2,0)})$ образуют двойные порты для обмена данными между различными уровнями представления. Так, например, выходные данные подсистемы S_1 из порта $Y_1^{(1)}$ поступают на входной порт $X_1^{(2)}$ подсистемы S_2 . Порт $X_1^{(2)}$ дублируется портом $X_1^{(2,0)}$, из которого данные попадают на входные порты $X_1^{(2,1)}$ и $X_1^{(2,2)}$ элементов $F_1^{(2)}$ и $F_2^{(2)}$ соответственно.

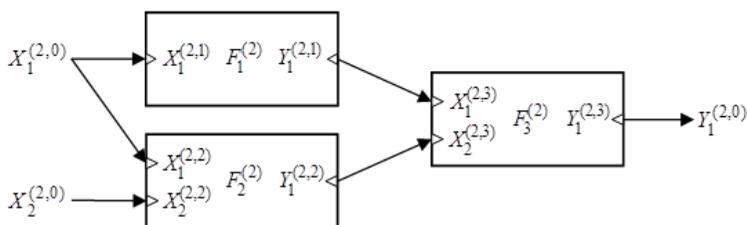


Рис. 2. Декомпозиция подсистемы S_2

Подобное математическое описание системы позволяет сформулировать принцип работы разрабатываемого программного средства для визуального моделирования. Такое программное средство должно иметь возможность визуального программирования моделей с использованием стандартных графических компонентов – узлов, представляющих элементы системы, на основе технологии drag and drop.

В соответствии с (5) каждый элемент системы определяется тройкой $\langle X^{(i)}, Y^{(i)}, \Phi_i \rangle$, то соответствующий узел должен содержать настройки для входов $X^{(i)}$ и выходов $Y^{(i)}$ и реализовывать некий алгоритм Φ_i , который преобразует входные значения $x^{(i)}$ в выходные $y^{(i)}$. Кроме того, в программном средстве должна быть возможность указывать связи между

выходами и выходами различных элементов, а также задавать значения входов в случае, если они не зависят от значений выходов других элементов.

Рассмотрим подробнее основные узлы системы – узлы алгоритмов, блоков и циклов.

2. Узлы алгоритмов

Узлы алгоритмов служат для преобразования одного набора данных в другой, соответствующий желаемому результату. Такие узлы являются расширяемой частью приложения, так как их набор реализуется в виде плагинов – динамически подключаемых библиотек классов (DLL), реализующих специализированный интерфейс `IAlgorithm`. Каждый класс алгоритма должен содержать свойства, соответствующие входным и выходным данным. Такие свойства должны быть помечены атрибутом `[Input]`, если оно отражает входной параметр, и `[Output]`, если выходной. Кроме того, класс должен реализовывать метод `Run()`, определенный в интерфейсе `IAlgorithm`. Этот метод содержит набор операций, которые вычисляют значения выходных свойств с использованием значений входных свойств. При визуальном редактировании узлов возможно задать значение входных параметров, а также указать, сохранять ли значения выходных значений в файлы.

3. Узлы блоков и циклов

Узлы блоков используются для выделения в системе подсистем. При настройке такого узла определяется структура подсистемы с использованием узлов алгоритмов, вложенных блоков и циклов, а также специальные узлы для входных и выходных параметров блока.

В соответствии с (6) и (7) узлы для входных и выходных параметров блока определяют пару портов, один из которых используется на уровне подсистемы, а второй на уровне родительской системы.

Узлы цикла являются расширением узлов блоков. Они позволяют осуществлять многократное выполнение блока с различным значением параметра – переменной цикла. На уровне родительской системы узел цикла получает на вход массив значений переменной цикла, в то время как на уровне подсистемы используется специальный узел переменной цикла, который содержит текущее значение переменной цикла.

4. Программная реализация узлов и связей между ними

Итоговая диаграмма классов разработанной системы для проведения эксперимента представлена на рис. 3.

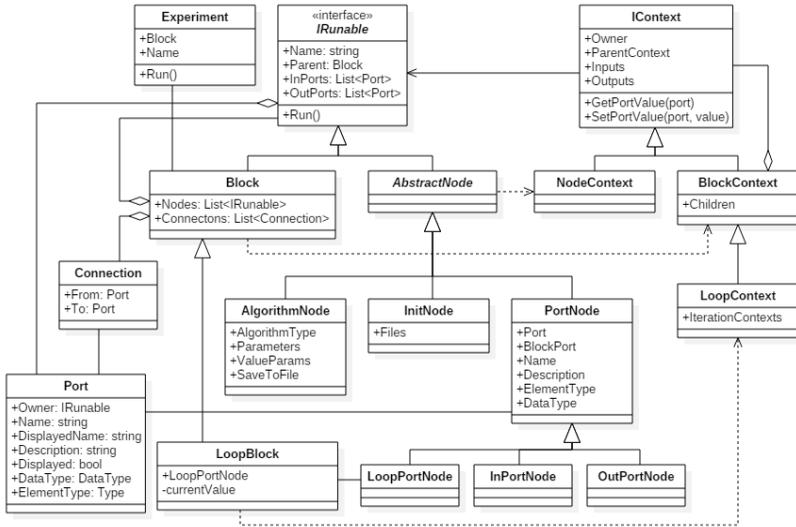


Рис. 3. Диаграмма классов системы

Узлы эксперимента образуют иерархическую структуру, элементы которой реализуют интерфейс `IRunnable`. Он имеет имя, родительский блок, а также набор входных и выходных портов. Каждый порт описывается классом `Port`, содержащем информацию о родительском узле, имени, отображаемом имени, описании порта, а также об используемой структуре данных и типе элементов. Интерфейс `IRunnable` реализуют классы `Block` и `AbstractNode`. `Block` – класс для узла блока, содержащего дочерние узлы `Nodes`, а также связи между портами узлов `Connections`. Каждая такая связь описывается классом `Connection`, который содержит информацию о портах `From` и `To`. От класса `Block` наследуется `LoopBlock`, реализующий узел цикла. `AbstractNode` – абстрактный класс для листовых видов узлов. Его наследниками являются классы для описания узла алгоритма `AlgorithmNode`, узла инициализации `InitNode`, а также узла порта `PortNode`. `PortNode` содержит информацию о двух портах: используемом на уровне блока – `Port` и используемом на уровне родительского блока – `BlockPort`. Его наследниками являются классы `InPortNode`, `OutPortNode` и `LoopPortNode`, описывающие узел входного параметра, узел выходного параметра и узел переменной цикла соответственно. `Experiment` – класс для эксперимента, содержащий корневой блок `Block` и имеющий метод `Run()` для выполнения эксперимента. В процессе выполнения эксперимента строится и заполняется контекст выполнения, который также

представляет собой иерархическую структуру с элементами, реализующими интерфейс IContext. Каждому типу узлов соответствует свой тип контекста: NodeContext – для наследников AbstractNode, BlockContext – для Block и LoopContext – для Loop. При вызове метода Run() любого узла создается соответствующий контекст, в который записываются значения входных и выходных портов узла. Таким образом, в конце выполнения эксперимента получаем древовидную структуру, содержащую результаты выполнения всех узлов.

Заключение

В данной статье была рассмотрена реализация механизма передачи данных между алгоритмами в экспериментах со вложенными циклами. Подобная реализация позволяет проводить эксперименты, в рамках которых могут моделироваться системы самой различной природы, так как набор алгоритмов является расширяемым, а рассмотренный подход к передаче данных позволяет создавать системы любого уровня сложности.

Список литературы

1. Красов В.Д. Применение моделированных последовательностей при оценке антропогенного воздействия на сток рек / В.Д. Красов, П.С. Лысачев // Матер. IX межд. научн.-метод. конф. «Информатика». – Воронеж, 2009. – Т. 1. – С. 375-378.
2. Красов В.Д. Управление режимом первоначального наполнения водохранилищ многолетнего регулирования стока / В.Д. Красов, П.С. Лысачев // Вестн. Воронеж. гос. ун-та. Сер. География. Геоэкология, 2009. – № 2. – С. 72-78.
3. Бусленко, Н. П. Моделирование сложных систем / Н. П. Бусленко. – М. : Наука, 1978. – 400 с.

РАЗРАБОТКА ВЕБ ПРИЛОЖЕНИЯ ВИДЕОЧАТ

А.А. Денищенко

Студент

А.Р. Нехаев

Ассистент

Введение

Видеочат относится к средствам общения с помощью электронных устройств, поэтому предметной областью данной статьи является

компьютерно-опосредованная коммуникация. Стоит отметить следующие особенности компьютерно-опосредованной коммуникации [1]:

- участниками коммуникации обязательно являются люди;
- характер средства связи – электронный;
- передаваемые сообщения чаще всего мультимедийны.

Компьютерно-опосредованная коммуникация является важной частью повседневной жизни многих людей. Благодаря компьютерно-опосредованной коммуникации, любой человек может общаться с другими людьми, не зависимо от времени и своего местоположения. Важным является и то, что этот вид коммуникации могут использовать люди с ограниченными возможностями.

В данной работе рассматривается задача разработки веб приложения с поддержкой видеочата. Особенностью проекта является соединение пользователей по принципу P2P. Задача включает в себя реализацию системы звонков, аутентификации пользователей, разделения пользователей на авторизованных и неавторизованных.

1. Постановка задачи

Цель работы: разработка веб-приложения, обеспечивающего потоковый обмен текстовыми сообщениями, аудио и видео данными.

К разрабатываемому приложению выдвинуты следующие функциональные и технические требования:

- возможность обмена данными между пользователями (текст, аудио, видео, файлы);
- система регистрации пользователей;
- возможность использования приложения для неавторизованных пользователей.

Главной задачей проекта является разработка веб-приложения, позволяющего пользователям этого приложения обмениваться между собой данными, как в текстовом, так и в медиа формате. Стоит отметить ещё один аспект проекта – возможность неавторизованным пользователям так же использовать приложение для осуществления коммуникации. Для наглядности ниже, на рис. 1 и рис. 2 представлены use case диаграммы, демонстрирующие функционал зарегистрированных и незарегистрированных пользователей.

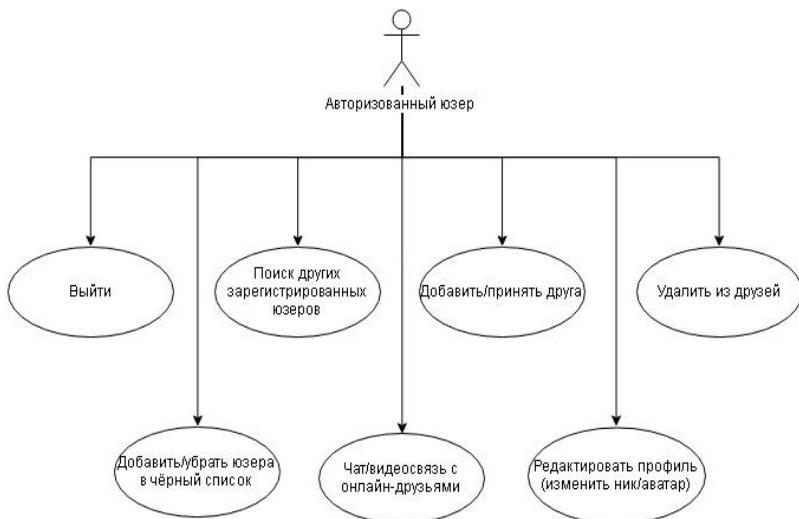


Рис. 1. Use case диаграмма для авторизованного пользователя

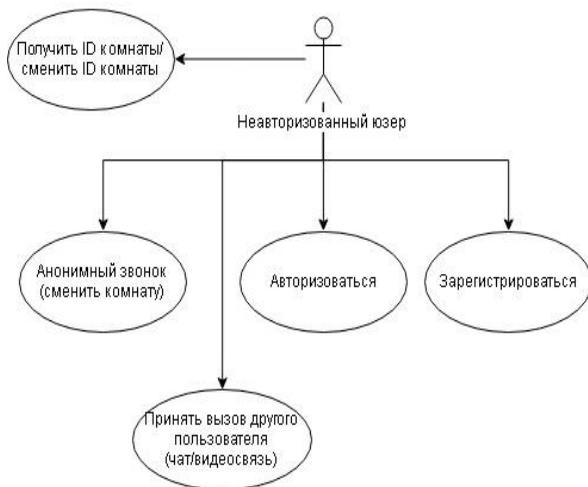


Рис. 2. Use case диаграмма для неавторизованного пользователя

2. Потокочная передача данных

В основе потоковой передачи данных между браузерами для проекта лежат технологии WebRTC. WebRTC – это крупный проект, который

содержит в себе множество технологий, объединённых в одно целое ради единственной цели – предоставление инструмента для разработки проектов по передаче данных между браузерами пользователей напрямую (P2P) [2]. Принцип соединения пиров приведён на рис. 3.

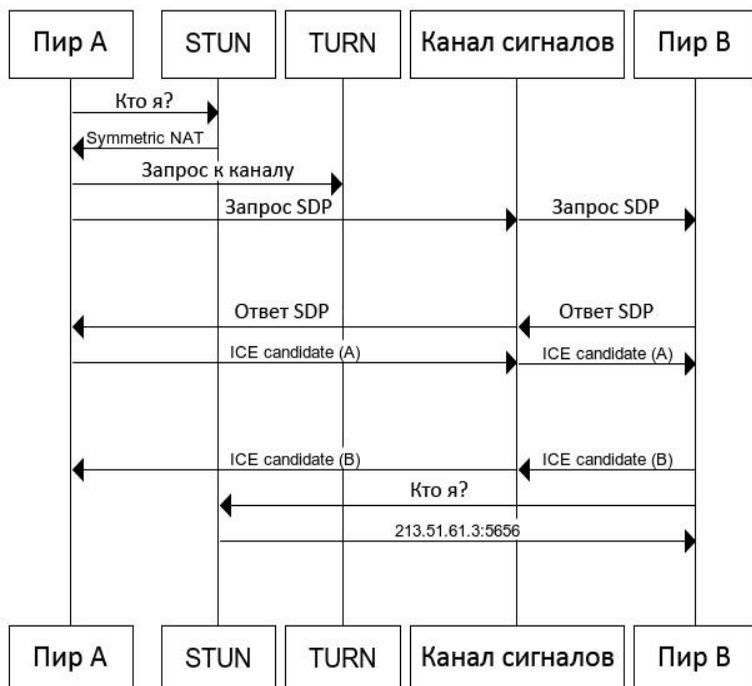


Рис. 3. Схема соединения пиров WebRTC

Как видно из данного рис. 3, WebRTC содержит в себе множество технологий – для установки соединения пользователей друг с другом используется сервер сигнализации, который по протоколу вебсокетов работает с пирами в режиме реального времени. Для построения оптимального маршрута между пирами используется метод ICE (Interactive Connectivity Establishment) [3]. А для работы с клиентами за NAT осуществлена поддержка серверов STUN и TURN. Как говорилось выше, тип соединения между пирами – P2P, и для его реализации используется API – PeerConnection, а в качестве канала связи выступает DataChannel.

3. Реализация

В качестве языка программирования был выбран JavaScript, так как платформа WebRTC реализована на этом языке. Для хранения

регистрационных данных используется СУБД MySQL. Для упрощения разработки приложения используется фреймворк easyRTC, так как настройка всех компонентов системы в WebRTC занимает достаточно большой промежуток времени.

Ниже приводятся примеры функций по подключению пира (листинг 1), передачи локального потока в DOM-элемент (листинг 2) [4].

Листинг 1

Вызов функции подключения пира

```
easyrtc.connect(  
    number_room,  
    callback_connect,  
    callback_connect_fail  
);
```

Эта функция реализует подключение клиента к комнате. В данном примере number_room – номер комнаты, callback_connect – функция обратного вызова при удачном подключении, callback_connect_fail – функция обратного вызова при неудачном подключении.

Листинг 2

Вызов функции передачи локального потока в DOM-элемент

```
easyrtc.setVideoObjectSrc(  
    document.getElementById("self"),  
    easyrtc.getLocalStream()  
);
```

Функция выше, при условии, что локальные устройства (камера/микрофон) захвачены, передаёт из устройств поток (easyrtc.getLocalStream()) в указанный DOM – элемент (document.getElementById("self")).

Разрабатываемое приложение имеет уклон в приватность разговоров, так например, содержание переписки пользователей нигде не хранится, а сама передача данных осуществляется напрямую из браузера в браузер по принципу P2P.

Стоит отметить, что в проекте есть чёткое разделение пользователей на авторизованных и неавторизованных. Это позволяет использовать приложение в качестве видеочата без регистрации. Сохраняя функциональность видеочата, система реализации звонков немного различается для авторизованных и неавторизованных пользователей.

Так, например, для авторизованных пользователей есть ограничение на выбор собеседника – любой авторизованный клиент может вызвать только тех участников, которые находятся у него в списке контактов. Помимо этого, система позволяет устраивать конференции с большим количеством участников (более тысячи). Фактическим ограничением на

количество участников выступает нагрузка на центральный процессор, вызываемая WebRTC, при взаимодействии пиров.

На рис. 4 приведена диаграмма последовательности, показывающая принцип осуществления звонков между авторизованными пользователями.

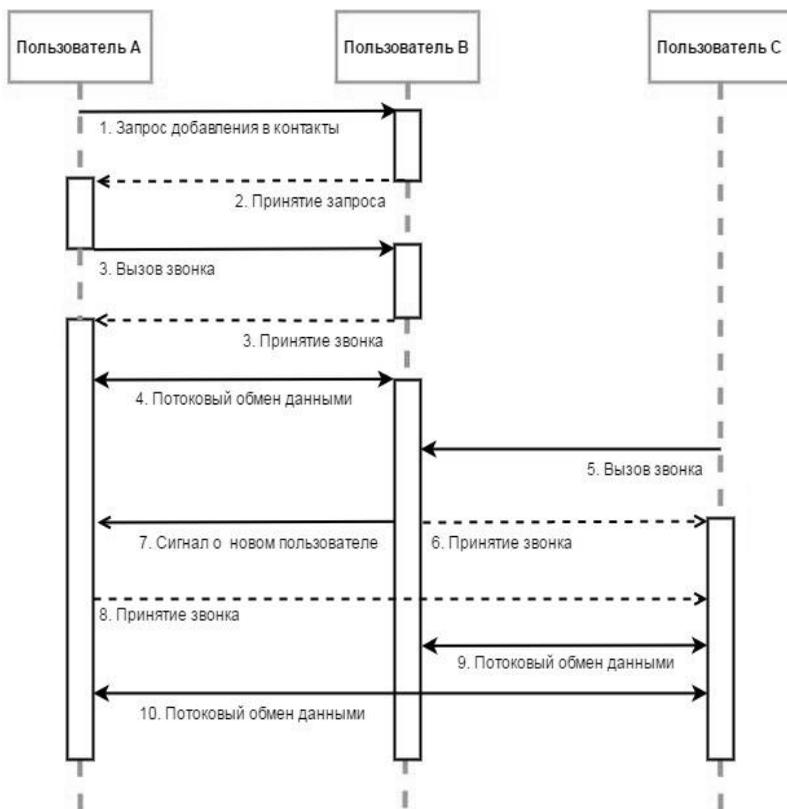


Рис. 4. Диаграмма последовательности, показывающая реализацию звонка

На данной диаграмме приведён пример случая, когда пользователи А и В не добавлены друг к другу в список контактов. При этом пользователи В и С уже знакомы, поэтому в диаграмме отсутствует между ними процесс добавления друг друга в контакты. Также особенностью конференций в системе является корневой (root) пользователь. По умолчанию корневым пользователем является принимающая вызов сторона. Корневой пользователь отвечает за добавление новых участников в конференцию. Таким образом, root-пользователь выступает единой точкой входа для

новых пиров. При этом, если пользователь покидает конференцию, либо связь с ним нарушается, то goot-пользователем становится участник, который находится в конференции раньше остальных.

Что касается неавторизованных пользователей, то их взаимодействие осуществляется через так называемые «комнаты». Каждый неавторизованный пользователь, не находящийся в общении с другими пирами, имеет свою комнату с уникальным идентификатором. Для того, чтобы связаться с другим клиентом, необходимо узнать его номер комнаты. Недостатком такой реализации является необходимость других способов коммуникации для обмена номерами комнат. После того, как пользователь узнал номер комнаты другого пользователя, он делает запрос на вход в эту комнату (аналог «звонка» для авторизованных пользователей). Как и в случае с зарегистрированными пользователями, вызываемая сторона решает – принять вызов или отклонить (будет получено уведомление с номером вызывающей стороны). Для создания конференций среди незарегистрированных пользователей, достаточно сообщить участникам номер одной и той же комнаты. Это и есть главное отличие авторизованных и неавторизованных пользователей.

Заключение

В рамках данной работы было реализовано веб приложение видеочат, позволяющее пользователям, как с регистрацией, так и без, полноценно использовать это приложение для общения друг с другом. Была реализована система звонков с корневым пиром, через которого осуществляется подключение новых пользователей. А отсутствие хранения переписки пользователей и передача данных по принципу P2P делает это приложение с уклоном на приватность, и отличает от существующих решений коммуникации.

Список использованной литературы

1. Баркович, А. А. Интернет-дискурс: компьютерно-опосредованная коммуникация : учебное пособие по интернет-лингвистике / А. А. Баркович. – М. : ФЛИНТА, 2015. – 285 с.
2. Sergiienko A. WebRTC Cookbook / A. Sergiienko. – Birmingham : Packt Publishing Ltd, 2015. – 230 с.
3. HTML5Rocks [Электронный ресурс] : Getting Started with WebRTC. – Режим доступа : <https://www.html5rocks.com/en/tutorials/webrtc/basics/>
4. EasyRTC [Электронный ресурс] : Documentation class Easyrtc. – Режим доступа : <https://easyrtc.com/docs/client-api/Easyrtc.php>

УНИВЕРСАЛЬНЫЙ АВТОМАТИЧЕСКИЙ ТЕСТЕР РАДИОДЕТАЛЕЙ

А.С. Железной

Студент

А.С. Зуев

Ассистент

Введение

Каждый радиолюбитель хоть раз встречался со следующей проблемой: вы выпаяли транзистор из печатной платы или достали один из коробки. Если на нем есть маркировка, у Вас уже есть паспорт или Вы можете получить документацию об этом элементе, то все в порядке. Но если таковая отсутствует, то Вы скорее всего, понятия не имеете, что это за элемент. Традиционный подход измерения всех параметров сложный и трудоемкий. Элемент может быть N-P-N, P-N-P, N или P-канальным полевым транзистором или MOSFET транзистором и т.д. Идея данного прибора заключается в том, чтобы переложить ручную работу на AVR микроконтроллер.

1. Цель работы

Целью данной работы являлась разработка прецизионного прибора для определения доколевки и рабочих параметров радиодеталей. Идея является актуальной в сфере электроники и образования. Прибор можно использовать как на уроках физики для обучения детей принципам работы различных электронных компонентов, так и в мастерских радиоэлектроники или в домашних условиях для починки некоторых приборов [1, 2].

2. Принцип работы

Принцип работы устройства (рис. 1) может быть рассмотрен на примере резисторов и конденсаторов.

Резисторы измеряются дважды (в обоих направлениях) и значения сравниваются [1]. Если значения отличаются намного, тестер предполагает, что есть два резистора вместо одного. В этом случае тестер выводит на экран результат как два резистора с номерами контактов «1 – 2 – 1», и отличающимися двумя значениями сопротивления. Для резисторов с сопротивлением ниже, чем 10 Ом выполняется дополнительное измерение с более высоким разрешением. В некоторых, редких случаях, тестер не может обнаружить очень малое сопротивление.

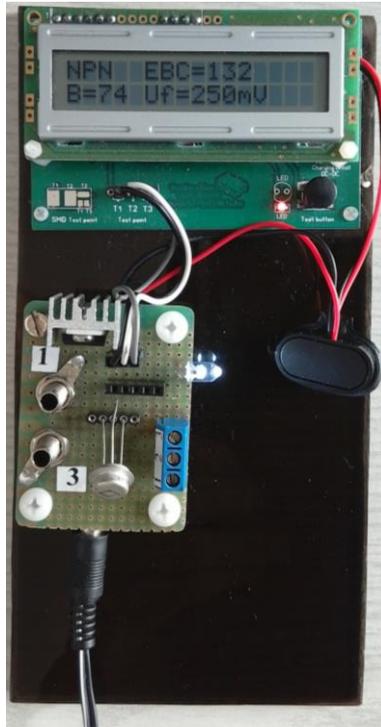


Рис. 1. Наглядное представление работы прибора

Для измерения емкости конденсаторов используется три метода [2]. Конденсаторы емкостью больше $47\mu\text{F}$ измеряются методом зарядки импульсами длительностью 10 мс. Конденсаторы в диапазоне от $4.7\mu\text{F}$ и $47\mu\text{F}$ тестируются таким же образом, но с зарядным импульсом длительностью 1 мс. Конденсаторы с малой емкостью тестируются методом аналогового компаратора.

3. Проблемы и решения

На сегодняшний день, на рынке существуют электронные приборы со схожим функционалом, например, LCD Multimeter LCR-T4 ESR Tester (рис. 2).



Рис. 2. LCD Multimeter LCR-T4 ESR Tester (with battery) Capacitance Inductance Meter Transistor Tester for DIY Electronic Kit

Однако, в их работе замечены некоторые проблемы, а именно:

- Неправильное определение типов полупроводниковых приборов, которое связано с тем, что некоторые параметры у биполярных транзисторов (при замене коллектора на эмитер) и у полевых транзисторов (при переключении истока и стока) мало отличаются;
- Неправильное определение параметров полупроводниковых приборов, которое связано с тем, что в данном приборе применяется режим микротоков и низкого рабочего напряжения. При этом, рабочие токи соизмеримы или даже меньше обратных токов и сквозных токов транзисторов;

В нашем приборе реализованы следующие методы решения:

- Для преодоления первого недостатка применен подход, основанный на измерении температуры испытуемой детали. При неправильном подключении транзисторов температура увеличивается более резко, чем при правильном.
- Для исправления второго недостатка используются режимы рабочих токов и напряжений и усовершенствованы схмотехнические решения.

4. Типы тестируемых радиодеталей

Далее представлены типы электронных компонентов, пригодных для тестирования [1]:

- NPN транзисторы – на дисплее «NPN»;
- PNP транзисторы – на дисплее «PNP»;
- N-канальные-обогащенные MOSFET – на дисплее «N-E-MOS»;

- P-канальные-обогащенные MOSFET – на дисплее «P-E-MOS»;
- N-канальные JFET – на дисплее «N-JFET»;
- P-канальные JFET – на дисплее «P-JFET»;
- Тиристоры – на дисплее «Tyrystor»;
- Симисторы – на дисплее «Triak»;
- Диоды – на дисплее «Diode»;
- Двухкатодные сборки диодов – на дисплее «Double diode CK»;
- Двуханодные сборки диодов – на дисплее «Double diode CA»;
- Два последовательно соединенных диода – на дисплее «2 diode series»;
- Диоды симметричные – на дисплее «Diode symmetric»;
- Резисторы – диапазон от 0,5 К до 500К [К];
- Конденсаторы – диапазон от 0,2нF до 1000уF [нF, уF].

Заключение

Данная статья посвящена созданию нового подхода к проектированию автоматизированных точных приборов для тестирования радиодеталей. Описаны проблемы в определении реальных характеристик электронных компонентов, а также способы их решения. В итоге данного эксперимента получился прибор с реализованными усовершенствованными методами определения типов полупроводниковых приборов и их параметров. Его применение возможно, как в сфере образования, так и радиолюбительской практики и автоматизации измерений при входном контроле на производстве.

Список литературы

1. Тестер ЭРЭ с AVR микроконтроллером и минимумом дополнительных элементов для Atmega 328/324/644/1284 с символьным или графическим дисплеем [Электронный ресурс] : http://vrtp.ru/screenshots/3588_ТТ_Markus_Reschke_Ru_10.04.17.pdf
2. Определитель выводов и основных параметров транзисторов и диодов / А. В. Гольшко [и др.] // Радио. – 2013. – № 12. – С. 43-61.

ФРЕЙМВОРК ДЛЯ РАЗРАБОТКИ ЧАТБОТОВ НА ЯЗЫКЕ JAVA

М.О. Желтоухов

Студент

А.И. Другалев

Ассистент

Введение

Создавая клиент-серверное приложение, разработчик сталкивается с проблемой выбора способа реализации клиентской части. Это действительно непростая задача, так как клиентская часть является посредником между пользователем и сервером, в связи с чем, необходимо учитывать и удобство со стороны пользователя, и практичность использования при установке взаимодействия с серверной частью. На сегодняшний день наиболее распространены три варианта реализации клиентской части: десктопное приложение, веб-приложение для браузера и мобильное приложение. Однако в 2016 году появился еще один способ – чатботы.

Чатбот – это результат взаимодействия человека с искусственным интеллектом: робот, который автоматически отвечает на сообщения, введенные пользователем в чате [1]. Иначе говоря, это программа, которая использует мессенджер (Telegram, Facebook, VK, Skype) как интерфейс для обмена данными с пользователями. Принцип работы следующий: пользователь отправляет команду в виде текстового сообщения чатботу, сообщение поступает на сервер мессенджера, затем, серверная часть чатбота с помощью Telegram Bot API (Application Programming Interface) мессенджера получает переданное сообщение-команду, обрабатывает её и отправляет пользователю ответ [2]. Таким образом, пользователь в диалоге с ботом взаимодействует с различными сервисами.

Фреймворк, описанный в данной статье, предназначен для упрощения и ускорения создания чатботов. А именно, позволяет распознавать команды в текстовом сообщении, выполнять соответствующую им бизнес логику, структурировать команды. Фреймворк предназначен для работы с чатботами мессенджера Telegram и написан на языке Java.

1. Классификация чатботов

Прежде чем разрабатывать фреймворк, необходимо определить, какие варианты чатботов могут быть. Это необходимо для того, чтобы

фреймворк был универсальным и подходил для реализации различных целей.

С точки зрения поведения ботов их можно разделить на две категории: виртуальный собеседник и исполнители команд (рис. 1). К первой относятся боты, имитирующие поведение человека, способные держать диалог с собеседником. Данный фреймворк посвящен второй категории, которая предназначена для предоставления услуг путем выполнения конкретных, заранее установленных текстовых команд. В свою очередь, таких ботов можно разделить на stateless (не зависящие от состояния пользователя) и stateful (поддерживающих хранение состояния).

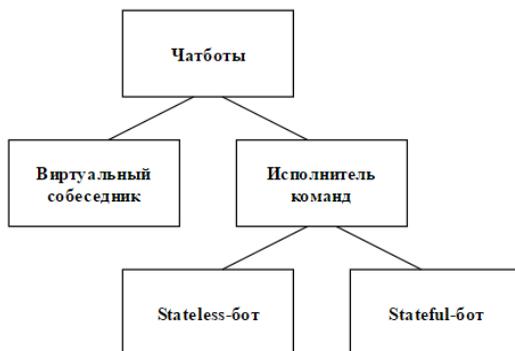


Рис. 1. Классификация чатботов

Stateless-чатбот выполняет команды и ему при этом не надо знать, от какого пользователя она последовала. Например, бот, предоставляющий прогноз погоды, где любой пользователь может запросить прогноз в конкретном городе, а бот вернет ему необходимую информацию. Очевидно, в таком простом варианте работы чатбота, возвращаемый результат не зависит от пользователя, а зависит только от команды, которую он пришлет.

Stateful-чатбот – более сложный вариант ботов. Он учитывает то, от какого пользователя пришел запрос, его состояние и, соответственно, для разных пользователей и при одинаковых командах, возвращает разный результат. Сложность же в создании таких ботов в том, что необходимо хранить информацию состоянии бота для пользователя, взаимодействующих с ботом. То есть, реализовывать хранилище информации (БД, файлы или, в худшем случае, списки объектов). Под состоянием пользователя подразумевается то, на каком этапе он находится в ходе какого-либо процесса. В качестве примера можно рассмотреть прохождение теста с помощью бота. Когда пользователь выбирает, какой

тест пройти – это одно состояние, бот ожидает получить идентификатор или название теста. Когда пользователь проходит тест – это уже другое состояние, и бот ожидает получить ответ на текущий вопрос теста.

Определение вида чатбота и возможных состояний во время выполнения различных процессов, поддерживаемых ботом, лежит на разработчике. Для этого необходимо детально спроектировать бота, прежде чем приступать к разработке с использованием данного фреймворка.

2. Архитектура и принцип работы фреймворка

Как уже упоминалось ранее, фреймворк упрощает и ускоряет создание Telegram чатботов. Для этого в нем разработаны следующие механизмы:

- Механизм создания бота на основе порождающего паттерна проектирования «Фабричный метод»;
- Механизм обработки команд на основе поведенческого паттерна проектирования «Команда» [3].

Прежде чем создавать чатбота, необходимо определиться, какому из перечисленных ранее видов он относится, а именно `stateful` или `stateless`. Для создания `stateful`-бота используется класс `StatefulBotCreator` (рис. 2). Он наследуется от абстрактного класса `BotCreator` и переопределяет метод `create`, который служит для инициализации бота. В качестве входных параметров метод принимает `Bot Token` (текстовый идентификатор бота), имя бота и объект типа `stateProvider`.

`stateProvider` необходим для предоставления информации, в каком состоянии находится бот по отношению к конкретному пользователю. Это абстрактный класс, поэтому разработчик должен создать своего наследника `stateProvider` и реализовать метод `getState`. Данный метод принимает на вход идентификатор чата в Telegram, из которого пришло сообщение, то есть, идентифицирует пользователя. В теле метода необходимо описать механизм определения состояния данного пользователя, например, путем обращения к БД, куда сохраняется данная информация. В итоге, метод возвращает объект перечислимого типа `BotState`, который в дальнейшем приходит в `BotMessageHandler` для определения соответствующей команды.

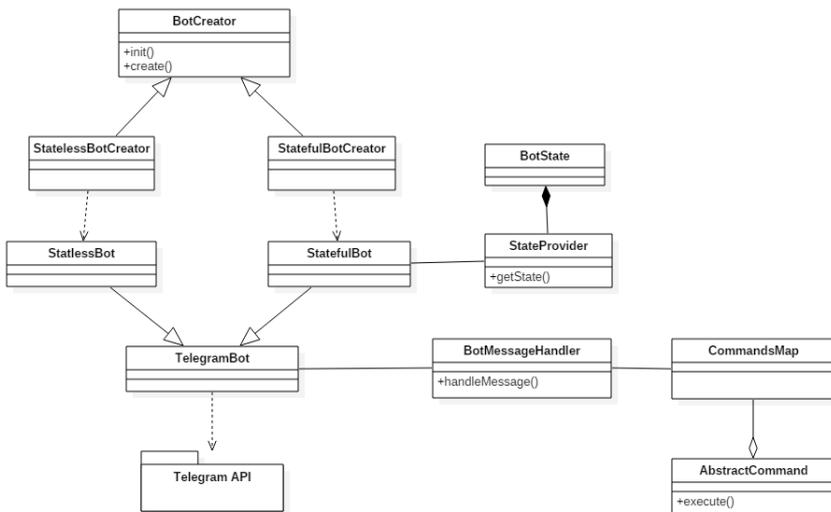


Рис. 2. Диаграмма классов фреймворка

В свою очередь, метод `create` класса `StatefulBotCreator` создает экземпляр класса `StatefulBot`, который унаследован от класса `TelegramBot`. Данный класс является основным в работе бота. Он взаимодействует с `Telegram API`. В нем переопределен метод `onUpdateChanges`, который вызывается, когда бот получает какое-либо сообщение. При получении сообщения, `TelegramBot` обращается к экземпляру класса `BotMessageHandler`, который отвечает за обработку сообщений и вызов соответствующих команд.

Основной особенностью фреймворка является механизм создания команд. Команда представляет собой класс, унаследованный от абстрактного класса `AbstractCommand`, в котором реализуется метод `execute`. Чтобы фреймворк определил, какой конкретно командой является класс, его необходимо пометить соответствующей Java-аннотацией [4]. Всего существует 4 аннотации:

1. `@TextBotCommand` – аннотация, для команд, которые определяются по конкретному ключевому слову.
2. `@RegexBotCommand` – аннотация для команд, которые определяются в соответствии заданному шаблону в виде регулярного выражения.
3. `@StateBotCommand` – аннотация для команд, которые вызываются при определенном состоянии пользователя бота. Поддерживаются только `stateful`-чатботами.

4. `@DefaultBotCommand` – аннотация для команды по умолчанию. Если такая задана, то она вызовется, если ни одна из описанных команд не подошла.

Приоритет выбора команд такой: текстовые команды, команды по шаблону, команды по состоянию, команда по умолчанию.

Команды инициализируются и помещаются в соответствующие ассоциативные массивы во время создания экземпляра класса `CommandsMap`. После обработки сообщения с помощью `BotMessageHandler`, у экземпляра класса соответствующей команды вызывается метод `execute`, который возвращает необходимый результат. В методе `execute` реализуется бизнес-логика команды и взаимодействие со сторонними сервисами.

3. Пример использования фреймворка

Для того, чтобы создать чатбота с помощью данного фреймворка, необходимо создать всего два класса, подключив при этом библиотеку фреймворка. В качестве примера рассмотрим создание простого stateless чатбота, который переводит слова с русского языка на английский.

В первую очередь, необходимо создать класс `Main` – исходную точку программы. Внутри этого класса создается метод `main`, который будет запускаться при старте приложения. В нем нужно вызвать статический метод `create` класса `StatelessBotCreator`. В качестве параметров передаются имя бота и `access token`. `Access token` разработчик получает при регистрации чатбота в Telegram.

Второй класс, который необходимо создать, это команда, отвечающая за перевод слова. Его необходимо унаследовать от класса `AbstractCommand` и реализовать в нем метод `execute`. Самое главное при создании команды – пометить ее соответствующей аннотацией. В данном случае это аннотация `@TextBotCommand`. В качестве параметра `value` для неё необходимо указать ключевое слово, например, «перевести». Это значит, что, когда пользователь отправит боту команду «перевести привет», вызовется метод `execute` описанного выше класса, в котором должна быть реализована логика для перевода текста, находящегося после ключевого слова. Например, в этом методе может происходить обращение к API сервиса «Яндекс Переводчик», который вернет перевод слова. Таким образом, в ответ на команду «перевести привет» бот отправит слово «hello».

Чтобы чатбот начал работу, необходимо запустить программу. Это можно сделать прямо в середине разработки на домашнем компьютере. Пока программа запущена, чатбот будет работать.

Заключение

Данная статья посвящена разработке фреймворка для создания чатботов в мессенджере Telegram. Он позволяет создавать ботов путем написания всего нескольких строчек кода и без установки программы на сервер приложений. Подход, когда команды реализуются в виде классов, достаточно удобен. Помимо простоты обозначения команды с помощью аннотации, разработчик получает структурированный набор команд в виде отдельных файлов (java-классов). При этом, учитывается тот факт, что команда может быть обозначена разными способами: ключевое слово, шаблон, состояние. Более того, несмотря на свою простоту, фреймворк предоставляет достаточно гибкие возможности по разработке как stateless, так и stateful чатботов. Таким образом, фреймворк предоставляет удобный механизм для разработки чатботов в качестве клиентской части своего приложения.

Список литературы

1. Чатботы захватывают мир [Электронный ресурс] : статья. – Режим доступа : <https://texterra.ru/blog/>
2. Telegram Bot API [Электронный ресурс] : документация. – Режим доступа: <https://core.telegram.org/bots/api>
3. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма [и др.]. – 4-е изд. – М. : Питер, 2016. – 366 с.
4. Java Platform, Standard Edition 8, API Specification [Электронный ресурс] : документация. – Режим доступа : <https://docs.oracle.com/javase/8/docs/api/>

ОПТИМИЗАЦИЯ АЛГОРИТМА ПОИСКА БЛИЖАЙШИХ ТОЧЕК ДЛЯ ПРИМЕНЕНИЯ В ИТЕРАТИВНОМ АЛГОРИТМЕ БЛИЖАЙШИХ ТОЧЕК (ICP)

М.В. Землянухин

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Итеративный алгоритм ближайших точек (англ. Iterative Closest Point – ICP) – алгоритм, использующийся для сведения к минимуму разницы между двумя облаками точек. ICP часто используется для восстановления

двухмерных (2D) или трёхмерных (3D) поверхностей из разных сканов (другими словами, для совмещения данных трехмерного сканирования – отдельных частей 3D модели), для определения местоположения роботов и планирования оптимального их пути, регистрации модели кости, сопоставление отдельных частей 3D моделей и т.д.

В итеративном алгоритме ближайших точек основные затраты производительности приходятся на многократное нахождение ближайших точек. При этом, в случае, если изначально облака точек находятся на большом расстоянии друг от друга, стандартно используемый в таких случаях алгоритм нахождения ближайших точек обрабатывает очень медленно, показывая неприемлемую производительность.

В работе представлено улучшение алгоритма поиска ближайших точек, которое решает данную проблему.

1. Стандартные структуры данных, применяемые в итеративном алгоритме ближайших точек

Итеративный алгоритм поиска ближайших точек концептуально прост и состоит из следующих шагов:

- Для каждой точки первого (перемещаемого) облака точек найти ближайшую точку во втором (неподвижном) облаке точек. Для определения ближайшей точки метрикой является обычное евклидово расстояние.

- Определить комбинацию поворота и перемещения второго облака точек, используя технику минимизации среднеквадратичного отклонения облаков точек, которая наилучшим образом выровняет каждую точку перемещаемого облака с соответствующей ей точкой неподвижного облака, найденной на предыдущем шаге.

- Преобразовать первое облако точек в соответствии с посчитанной на предыдущем шаге комбинацией поворота и перемещения.

- Если не достигнута требуемая точность приближения точек или количество итераций не достигло разрешенного максимума, алгоритм продолжается с первого шага, иначе – завершается.

В итеративном алгоритме ближайших точек основная часть затрат процессорного времени приходится на многократное нахождение ближайших точек (на нахождение ближайших точек приходится более 99% процессорного времени). Среднее количество итераций, необходимых для сходимости алгоритма, обычно равняется 20-40, однако в некоторых случаях вследствие изначально неудачного расположения облаков точек относительно друг друга может потребоваться более 100 итераций. 3D-модели, к которым, как правило, применяется алгоритм, могут содержать 100 тыс. – 1 млн. точек (возможно и больше), что в среднем случае приводит к 10-20 млн. операций нахождения ближайшей точки до

завершения алгоритма (и гораздо большему числу таких операций для худших случаев).

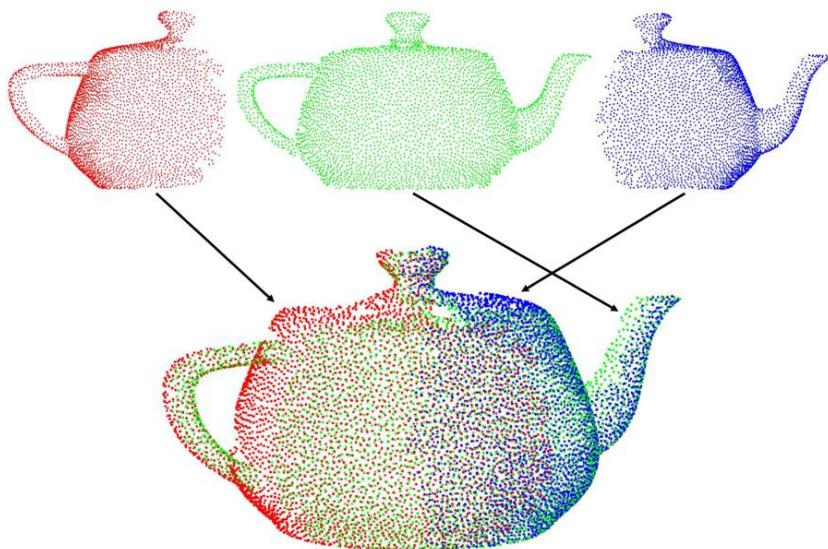
Для быстрого нахождения ближайших точек существуют различные техники и структуры данных. Примерами таких структур данных являются:

- мульти-размерные деревья (Kd-tree) [1];
- метрические деревья (R-tree);
- другие пространственные структуры [2].

В большинстве случаев, возникающих при решении различных задач, такие структуры данных дают требуемый прирост производительности, осуществляя нахождение ближайшей точки за асимптотику $O(\log(N))$, где N – количество точек в неподвижном облаке точек. Проблема возникает, когда требуется найти ближайшую точку для точки, находящейся на большом расстоянии от неподвижного облака точек. Такой случай для алгоритмов нахождения ближайших точек является вырожденным и для нахождения ближайшей точки требуется совершить гораздо больше операций. Например, для расстояния до ближайшей точки, превышающего максимальный размер всего облака, потребуется проверить большую часть точек, что ведет к асимптотике $O(N)$. В результате первые 1-3 шага итеративного алгоритма ближайших точек (см. рис. 1) рассчитываются очень долгое время даже в распараллеленном варианте алгоритмов (несколько минут).

Помимо структур данных для точного нахождения ближайшей точки, существуют их различные варианты для приближенного нахождения ближайших точек [3]. Такие структуры данных имеют другую проблему – «слишком» приближенное нахождение искомой ближайшей точки, настолько неточное, что расстояние до найденной точки и настоящей ближайшей точки может иметь порядок размера всего облака точек при достаточно большом расстоянии между двумя облаками точек. В случае с итеративным алгоритмом ближайших точек это может привести к неточной трансформации облака точек на первом шаге, которая потенциально может привести алгоритм в состояние несходимости.

В данной работе предложена структура данных, которая ускоряет нахождение ближайших точек для облаков точек, находящихся на большом расстоянии друг от друга. При этом сохраняется приемлемая для итеративного алгоритма ближайших точек точность нахождения точек. Расстояние до найденной точки отличается не более чем на 1% от расстояния до реальной ближайшей точке на первых итерациях алгоритма. На последующих итерациях точность увеличивается, пока не происходит перехода к абсолютно точному поиску.



Совмещение различных частей 3D объекта

Рис. 1. Одно из применений итеративного алгоритма ближайших точек

2. Оптимизация итеративного алгоритма ближайших точек

Предлагаемый метод использует комбинацию структур данных, которые позволяют точно определить ближайшую точку и применяются в случаях, когда точка запроса находится в облаке неподвижных точек или рядом с ним. Эта комбинация состоит из двух (в простейшей реализации) или нескольких мульти-размерных деревьев (Kd-tree). Алгоритм может использоваться и вне итеративного алгоритма ближайших точек, но был разработан и дает максимальных выигрыш именно как часть данного алгоритма.

Идея алгоритма заключается в том, чтобы использовать для начальных шагов итеративного алгоритма ближайших точек мульти-размерное дерево, построенное не на полном наборе неподвижных точек, а на частичном наборе. Чтобы получить частичный набор, происходит замена нескольких близко расположенных точек исходного облака единственной точкой, которая представляет это облако. Другими словами, происходит децимация исходного облака точек. Децимация задается радиусом децимации – максимальным расстоянием между паров точек в исходном облаке, при котором эти точки будут объединены в одну.

Усеченное мульти-размерное дерево содержит в 100-1000 раз меньше точек, позволяя получить соответствующий прирост производительности

для случая поиска ближайшей точки от отдаленной точки. В то же время, величина погрешности, с которой осуществляется поиск ближайшей точки, зависит только от радиуса децимации и не зависит от расстояния до точки запроса. В большинстве случаев, в которых в реальности применяется итеративный алгоритм ближайших точек, для которого было сделано данное улучшение, работает с облаками точек большой плотности. В таких облаках радиус децимации равный 0.1-1% размера области точек на несколько порядков уменьшает количество точек в мульти-размерном дереве.

Построение усеченного мульти-размерного дерева осуществляется путем повторного использования полного мульти-размерного дерева, содержащего все точки неподвижного облака. Структура мульти-размерного дерева позволяет за асимптотику $O(\log(N))$ находить все точки в заданном радиусе от данной, за счет чего и происходит замена облаков из точек одной точкой. Учитывая, что количество таких поисков сравнительно мало (т. к. децимированное мульти-размерное дерево содержит намного меньше точек), шаг, требующий построения дополнительного усеченного дерева, не несет заметной нагрузки с точки зрения производительности.

Как уже упоминалось выше, усеченное мульти-размерное дерево используется только для первых шагов итеративного алгоритма ближайших точек. После того, как наступают поздние шаги итеративного алгоритма ближайших точек, происходит плавный переход на поиск в полном мульти-размерном дереве для простого случая. В более сложных случаях может использоваться несколько усеченных мульти-размерных деревьев с различным радиусом децимации, тогда переход происходит к последующим усеченным мульти-размерным деревьям в направлении от большего радиуса децимации к меньшему. Решение о том, с какой итерации осуществлять переход на поиск в очередном мульти-размерном дереве, принимается для каждой точки подвижного облака отдельно. Таким образом, для каждой точки подвижного облака поддерживается индекс мульти-размерного дерева, в котором происходит поиск ближайшей к ней точки. Переключение на новое дерево происходит, если расстояние до ближайшей точки после очередного поиска оказывается маленьким относительно радиуса децимации мульти-размерного дерева, в котором происходил. Если расстояние мало по сравнению с радиусом децимации, увеличивается относительная погрешность нахождения точки, что критично для итеративного алгоритма ближайших точек. После перехода к очередному мульти-размерному дереву немедленно осуществляется повторный поиск в нем с целью увеличения точности нахождения ближайшей точки. Таким образом, в одной итерации

итеративного алгоритма ближайших точек для точки из подвижного облака может произойти несколько переходов к поиску в новых усеченных мульти-размерных деревьях.

На практике лучшие результаты были получены с применением одного-двух дополнительных усеченных мульти-размерных деревьев, по сравнению с применением большего их количества. Для простоты понимания ниже приводится пошаговый вариант алгоритма с одним дополнительным деревом, который легко можно распространить на вариант с несколькими дополнительными деревьями.

- Составить полное мульти-размерное дерево из всех точек неподвижного облака.

- Используя полное дерево, составить усеченное мульти-размерное дерево.

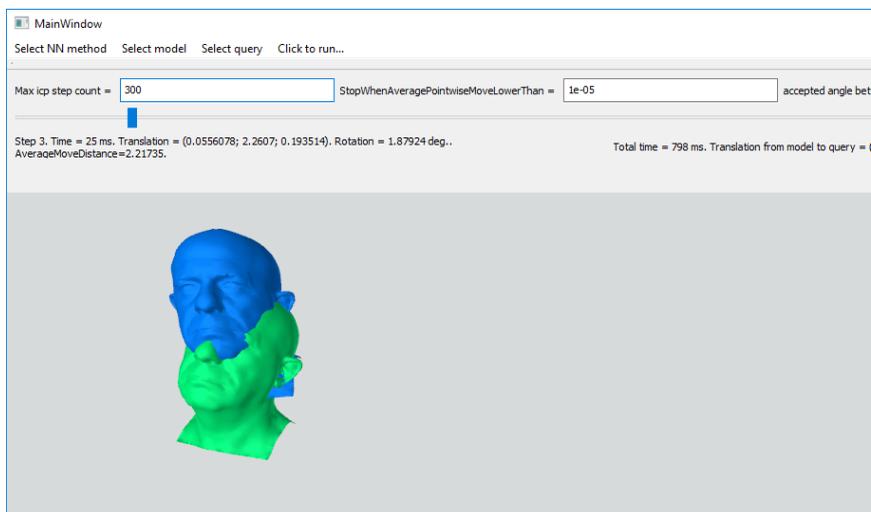
- Пометить все точки подвижного облака так, чтобы поиск ближайших к ним точек осуществлялся с использованием усеченного дерева.

- Выполнить стандартный итеративный алгоритм ближайших точек. После каждого поиска ближайшей к точке из подвижного облака точки из неподвижного облака проверять, насколько расстояние до найденной точки мало по сравнению с радиусом. Если расстояние мало (например, превышает радиус в 5 раз и меньше), пометка с данной точки снимается, и дальнейший поиск ближайших точек для нее осуществляется в полном дереве.

Заслуживает упоминания тот факт, что полученный алгоритм, так же, как и стандартный алгоритм поиска ближайших точек с использованием мульти-размерного дерева, можно естественным образом распараллелить на несколько ядер процессора.

Заключение

Тестирование алгоритма проводилось на процессоре Intel Core i7 3770. В ходе тестирования проводились многократные запуски полного итеративного алгоритма ближайших точек на различных 3D моделях с различными начальными положениями моделей относительно друг друга (см. пример на рис. 2) до его полного схождения. В качестве алгоритма поиска ближайших точек использовался вышеописанный алгоритм и стандартный алгоритм поиска ближайших точек с использованием одного мульти-размерного дерева, после чего проводилось их сравнение.



Третья итерация итеративного алгоритма ближайших точек

Рис. 2. Пользовательский интерфейс тестирующей программы

Представленный алгоритм обеспечил прирост производительности в 3-5 раз по сравнению со стандартной версией. Использование памяти увеличивалось менее, чем на 1-2%. При этом не было зафиксировано случаев, когда стандартный алгоритм успешно сходился (разница между облаками точек сводилась к минимуму), а представленный алгоритм нет.

Список литературы

1. Greenspan M. Approximate K-D Tree Search for Efficient ICP / M. Greenspan, M. Yurick // Proceedings of the 4th IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling (Banff, Canada, October 2003). – 2003. – P. 442-448.
2. Nuchter A. Cached k-d tree search for ICP algorithms / A. Nuchter, K. Lingemann, J. Hertzberg // Proceedings of the 6th IEEE International Conference in 3D Digital Imaging and Modeling (Montreal, Canada, August 2007). – 2007. – P. 419-426.
3. Guttman A. R-trees: A Dynamic Index Structure for Spatial Searching / A. Guttman // Proceedings of the 1984 ACM SIGMOD international conference on Management of data (Boston, USA, June 1984). – 1984. – V. 14. – №. 2. – P. 47-57.

ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ БОТ-ТРАФИКА СРЕДИ ЗАПРОСОВ К ВЕБ- ПРИЛОЖЕНИЮ

Ю.Ю. Иванников

Студент

Е.Ю. Митрофанова

Доцент

Введение

С развитием современных веб-технологий появляется все больше методов нелегитимного использования веб-ресурсов и контента, который они предоставляют пользователям. Для подобных целей в подавляющем количестве случаев злоумышленники используют ботов – программы, замаскированные под рядовых пользователей веб-ресурса, выполняющие определенную последовательность действий. Целью использования ботов может являться:

- парсинг веб-страниц с целью копирования контента;
- проведение DDoS-атак, направленных на деградацию сервиса;
- сканирование уязвимостей для проведения хакерских атак и так далее.

Существуют и «полезные» боты, например: боты поисковых машин или сервисов мониторинга доступности ресурса, однако владельцы этих ботов обычно не маскируют их и передают информацию о принадлежности бота в поле User-agent заголовка запроса к веб-ресурсу.

В данных условиях актуальной является проблема выявления ботов среди всех пользователей. Данная работа посвящена исследованию возможности применения методов машинного обучения и интеллектуального анализа данных для выявления ботов в общем трафике веб-ресурса.

1. Выявление бот-трафика

Подходы к выявлению запросов ботов в общем трафике приложения можно разделить на группы по следующим критериям:

- по используемым методам классификации (статистические, поведенческие, гибридные);
- по объектам анализа (единичные запросы, сессии, группы сессий);
- по используемым показателям запроса.

Основная сложность выявления ботов заключается в преднамеренной маскировке трафика атакующего ботнета под трафик обычных

пользователей, причем методы маскировки зачастую являются достаточно изоощренными.

Алгоритмы генерации запросов с устройств, состоящих в атакующем ботнете, можно классифицировать по следующим показателям:

- корректность запрашиваемого пути: только существующие пути, случайные пути, смешанные пути;
- содержание запроса: одинаковое или различное;
- интервал между запросами: одинаковый или разный;

Кроме того, следует упомянуть, что шаблоны поведения разных ботов могут очень сильно различаться, и чем ближе будет модель поведения бота к модели поведения легитимного пользователя, тем сложнее будет отличить «полезный» трафик от «паразитного».

Классическим подходом к ограничению доступа ботов к веб-ресурсу является принуждение всех пользователей проходить тест «на робота» – вводить текст с картинки, распознавать искаженные изображения и т.д. Этот подход является действенным, однако имеет побочный эффект в виде задержки использования сервиса пользователями, и, как следствие, потерь для бизнеса в случае, если действительный пользователь отказывается проходить тест.

2. Цель работы

Цель работы – исследование возможности использования методов машинного обучения для решения задачи распознавания бот-трафика среди всех запросов к веб-ресурсу.

Задачами являются:

- поиск и подготовка данных для анализа;
- определение метрики для оценивания качества распознавания ботов;
- оценка качества алгоритмов машинного обучения или их ансамблей с использованием выбранной метрики.

3. Подготовка данных

Для достижения необходимого объема обучающей выборки в данной работе предлагается использовать логи веб- или прокси-сервера за некоторый период времени, с последующей их обработкой. В качестве источника данных для обучения в данной работе выступает веб-сервер Nginx 1.10.2 со стандартным форматом ведения лога, используемый в качестве балансировщика нагрузки в реальном проекте.

Данные из логов перед обучением необходимо предварительно обработать. В данной работе единицей наблюдения является сессия – последовательность запросов с одного IP-адреса, интервал между которыми не превышает 30 минут.

После распределения всех запросов по сессиям из выборки удаляются сессии, в которых менее 2 запросов (как неинформативные).

Далее, для каждой сессии выявляются следующие численные характеристики (которые будут выступать в качестве признаков при обучении):

- количество запросов / продолжительность сессии в секундах;
- количество запросов к страницам / продолжительность сессии в секундах;
- размер ответа (максимальный, минимальный и средний);
- время между запросами (максимальное, минимальное и среднее);
- время между запросами к страницам (максимальное, минимальное и среднее).

Общее количество используемых на данном этапе признаков: 11.

Далее данные масштабируются на отрезок [0..1] во избежание введения в заблуждение алгоритмов машинного обучения, использующих меры дальности между объектами как критерий для классификации.

Общее количество объектов в выборке: 11435, из них помечены как боты: 299.

4. Выбор метрики качества

Метрикой качества в данной работе выбрана F-score [1].

Для расчета F-score вводятся понятия “точность” (precision) и «полнота» (recall).

$$\text{precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (2)$$

$$F = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \quad (3)$$

где TP – истинно-положительное решение; TN – истинно-отрицательное решение; FP – ложно-положительное решение; FN – ложно-отрицательное решение.

F-score является средним гармоническим между точностью и полнотой, что, в целом, подходит для применения в данной работе в качестве метрики качества классификации.

5. Сравнение методов машинного обучения

Исходя из анализа публикаций по теме, были выделены несколько перспективных методов машинного обучения:

- логистическая регрессия;
- SVM;

- нейронные сети;
- «случайный лес» (random forest);
- градиентный бустинг над решающими деревьями (AdaBoost, xgboost и т.д.).

Все вышеперечисленные алгоритмы реализованы в библиотеке sklearn [2] для языка программирования python.

Для одновременного обучения и оценивания работы классификаторов использовалась техника кросс-валидации [3], позволяющая с большой точностью оценить качество классификатора без необходимости подготовки отдельных обучающей и тестовой выборок.

Далее для каждого классификатора был запущен процесс поиска оптимальной комбинации параметров настройки (так называемый «поиск по сетке»).

Сравнение качества работы классификатором представлены в следующей таблице:

Таблица

Сравнение качества алгоритмов классификации

Алгоритм	Значение F-score
Логистическая регрессия	0.90
SVM	0.89
XGBoost	0.95
Многослойный персептрон (2 слоя)	0.84
Случайный лес	0.93

Наибольшую точность показал ансамблевый алгоритм градиентного бустинга над деревьями решений (в реализации xgboost), дальнейшее исследование будет основано именно на нем.

6. Повышение точности классификации

Для повышения точности классификации ботов были использованы следующие подходы:

- сэмплинг (sampling) [4] выборки данных для обучения. Данный метод использовался для уменьшения дисбаланса классов (т.к. количество объектов, помеченных как «бот», в выборке данных значительно меньше, чем объектов класса «человек»);
- очистка данных от выбросов и экстремальных значений;
- применение алгоритма RFE (Recursive Feature Elimination) для отбора наиболее значимых для классификации признаков.

Достигнутая в результате вышеописанных действий значение F-score: 0.981.

Заключение

Целью данной статьи было исследование возможности применения методов машинного обучения для решения задачи выявления ботов из общего трафика веб-ресурса. В ходе работы была подготовлена выборка данных для обучения, выбрана метрика качества классификации, произведен сравнительный анализ методов машинного обучения на подготовленных данных с выбранной метрикой. По результатам сравнения классификаторов был выбран наилучший (по критерию выбранной метрики), после чего был произведен ряд действий, направленных на повышение качества работы классификатора.

В ходе работы была достигнута F-score = 0.981 на кросс-валидации, что является положительным результатом и позволяет считать цель исследования достигнутой.

Список литературы

1. Большев, А.К. Алгоритмы преобразования и классификации трафика для обнаружения вторжений в компьютерные сети: дис. канд. техн. наук: 05.13.11. – СПб., 2011. – 142 с.
2. Scikit-learn: Machine Learning in Python / Fabian Pedregosa [и др.] // Journal of Machine Learning Research. – 2011. – №12. – С. 2825-2830.
3. A survey of cross-validation procedures for model selection / Sylvain Arlot [и др.] // Statistics Surveys. – 2010. – №4. – С. 40-79.
4. SMOTE : Synthetic Minority Over-sampling Technique / Nitesh V. Chawla [и др.] // Journal of Artificial Intelligence Research. – 2002. – №16. – С. 321-357.

АНАЛИЗ СРЕДСТВ ПО УСКОРЕНИЮ ВЫЧИСЛЕНИЙ ДЛЯ УСТРОЙСТВ НА ПЛАТФОРМЕ ANDROID

М.С. Ильин

Студент

В.С. Тарасов

Ассистент

Введение

В мире современных информационных технологий все больший оборот набирают задачи, связанные с искусственным интеллектом и машинным обучением. Самыми яркими примерами являются такие задачи, как: распознавание объектов на изображениях, анализ и синтез речи, задачи анализа данных.

Одним из основных и наиболее эффективных подходов к решению задач машинного обучения является использование искусственных нейронных сетей. В задачах обработки изображений одним из самых эффективных методов являются сверточные нейронные сети (СНС) [1]. СНС используют многоуровневые модели, в которых последовательно чередуются сверточные и субдискретизирующие слои. При подобной архитектуре производится большое количество математических операций.

Несомненно, одним из ключевых моментов является производительность вычислений. На настольных и серверных платформах для оптимизации вычислений применяются такие технологии, как: OpenMP, Nvidia CUDA, ATI Stream Computing. Иначе дело обстоит с мобильными платформами: мобильные графические чипы не совместимы с CUDA или Stream Computing. В данной статье представлен краткий обзор и анализ производительности некоторых технологий в применении к устройствам Android.

1. Цель работы

Целью данного исследования заключается в получении информации о доступных средствах оптимизации вычислений для устройств Android, а также получение информации о производительности тех или иных решений.

2. Native Development Kit

При анализе средств по оптимизации вычислений для Android в первую очередь следует обратить внимание на Native Development Kit. Это набор утилит для разработки, расширяющий стандартный Android SDK, который позволяет осуществлять сборку библиотек, написанных на языках C/C++ под архитектуры мобильных процессоров [2]. Работа с такими библиотеками может осуществляться через стандартный для языка Java интерфейс JNI. Использование кода, скомпилированного непосредственно в машинные инструкции, дает значительный прирост в производительности вычислений.

3. General-Purpose computing for Graphics Processing Units

GPGPU – это подход к разработке программного обеспечения, в котором часть прикладных вычислительных операций выполняется на графическом процессоре [3]. GPU проектируются для выполнения множества параллельных потоков и содержат в себе сотни процессорных ядер, способных исполнять команды параллельно. В GPU не возникает накладных расходов на синхронизацию потоков, поскольку, данные изначально предназначены для независимой обработки. Существует несколько технологий, которые позволяют применить данный подход при написании алгоритмов для устройств на ОС Android.

4. RenderScript и FilterScript

RenderScript – это компонент операционной системы Android, разработанный специально для обеспечения высокой производительности в 3D рендеринге и сложных вычислительных операциях [4]. RS предоставляет язык, являющийся подмножеством языка C, код на котором обрабатывается исполнительным модулем, встроенным в ОС и может выполняться как на CPU, так и на GPU. Данная технология является кроссплатформенной для Android устройств и не требует сборки под каждый CPU и GPU. FilterScript является более строгим подмножеством RenderScript (содержит ряд ограничений), однако, это обеспечивает более высокую совместимость с различными типами устройств.

RS и FS предоставляют API схожий с технологией CUDA, что позволяет использовать их для различных вычислительных операций, не предназначенных для обработки компьютерной графики.

Аппаратное устройство, на котором будут производиться вычисления неизвестно непосредственно до момента начала вычислений и определяется операционной системой. Это значит, что RS не дает гарантий того, что код будет выполняться на GPU. Ограничения FilterScript позволяют сделать его более строгим, что дает гарантию запуска скриптов на графических устройствах при их наличии.

5. Open Graphics Library for Embedded Systems

OpenGL ES является подмножеством графического интерфейса OpenGL адаптированным для работы на встраиваемых устройствах, в том числе и мобильных платформах [5]. OGL является инструментом для работы с двухмерной и трехмерной компьютерной графикой не зависящим от конкретной платформы. Данный программный интерфейс предоставляет широкий набор возможностей для проведения вычислений предназначенных для отрисовки изображений.

Основная спецификация OpenGL предоставляет набор функций предназначенных для отображения визуальных элементов. Несмотря на это, в OpenGL имеется возможность для выполнения отдельно взятых подпрограмм непосредственно на видеопроцессоре. Данные подпрограммы называются шейдеры и пишутся на языке GLSL (Graphics Library Shading Language), предназначенном для работы с графическими вычислениями. Данную возможность OpenGL можно использовать при написании GPGPU программных решений.

Поскольку OGL предназначен исключительно для работы с задачами компьютерной графики, использование его для GPGPU требует дополнительных знаний в области компьютерной графики и рендера изображений.

6. Open Computing Language

OpenCL это фреймворк разработанный, непосредственно, для написания высокоэффективных алгоритмов и осуществления параллельных вычислений как на CPU, так и на GPU [6]. Данное решение является открытым и кроссплатформенным. OpenCL позволяет явно воздействовать на выбор устройства для вычислений в момент работы приложения.

OpenCL не является частью операционной системы Android, и реализован на более низком уровне и для запуска программ, написанных на OpenCL необходима поддержка непосредственно со стороны аппаратного обеспечения. В настоящий момент данная технология поддерживается большинством современных мобильных видеочипов.

Для написания подпрограмм, предназначенных для исполнения на GPU, данная технология предоставляет язык, являющийся подмножеством языка C и удобный API, изначально спроектированный для GPGPU. Данный факт значительно упрощает разработку по сравнению с использованием OpenGL ES.

7. Производительность

В 2013-2014 годах в работе [7] проводились исследования производительности рассматриваемых решений непосредственно для мобильной архитектуры. В качестве тестового устройства использовался Google Nexus 10 со следующими характеристиками:

- CPU: 1.7 GHz, dual-core Cortex-A15
- GPU: Mali T604
- RAM: 2 GB LPDDR3

В рамках данного исследования производилось измерение времени выполнения фильтра Гаусса размерностью 3x3 для фильтрации изображений различных разрешений при помощи C++, RenderScript, OpenGL ES и OpenCL. На диаграммах (рис. 1 и рис. 2) представлена информация времени выполнения фильтра, а также информация о времени копирования данных между основной и видео памятью.

В связи со значительным разбросом временных интервалов, для удобства отображения, на всех графиках в данном разделе используется логарифмическая шкала.

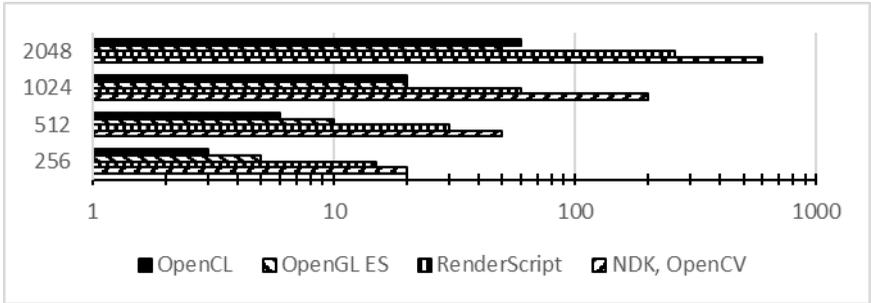


Рис. 1. Время работы фильтра Гаусса 3x3

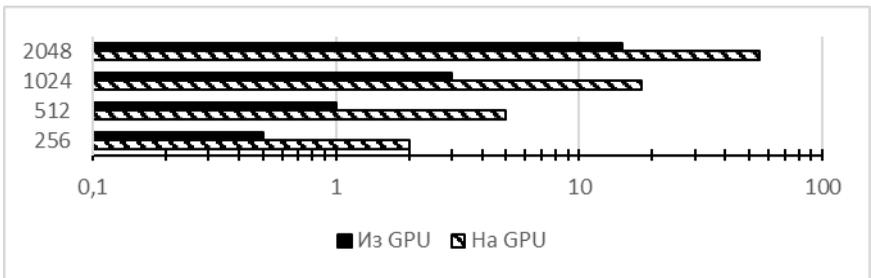


Рис. 2. Время копирования между основной и видеопамятью для OpenCL

Как можно видеть наибольшую производительность дает использование OpenCL или OpenGL ES, даже при учете затрат на копирование данных.

С 2014 года существенно выросла как мощность аппаратного обеспечения, так и производительность программной поддержки со стороны ОС Android или производителей графических чипов. В рамках текущей работы было проведено исследование подобное представленному выше. В рамках исследования использовался фильтр Гаусса размером 5x5 для фильтрации изображений различных разрешений. Были произведены замеры времени для RenderScript, FilterScript и OpenCL. В качестве тестового устройства использовался смартфон Xiaomi Redmi Note 2 со следующими характеристиками:

- CPU: 2.0 GHz, octa-core Cortex-A53
- GPU: PowerVR G6200
- RAM: 2 GB LPDDR3

Было произведено 10 измерений для каждой размерности. На рисунках представлены усредненные результаты измерений.

Время непосредственного выполнения алгоритма с использованием OpenCL в большинстве случаев значительно меньше времени работы RS или FS (рис. 3). Кроме того, видна нестабильность в работе RS и FS: на протяжении всех 10 тестов возникали случаи заметного отклонения результатов от среднего значения.

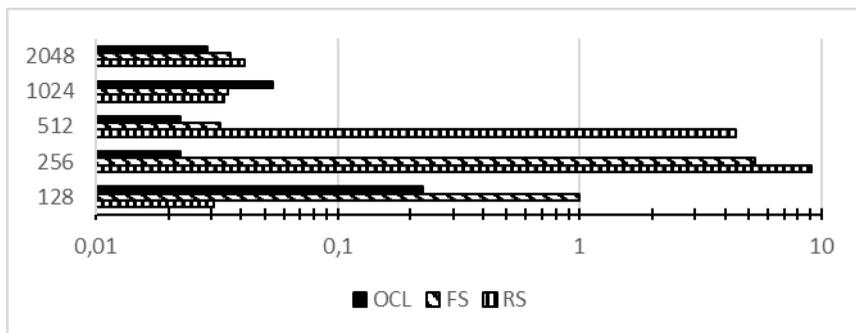


Рис. 3. Время работы фильтра Гаусса 5x5

Время выполнения алгоритма с учетом затрат на копирование данных также меньше для OpenCL (рис. 4).

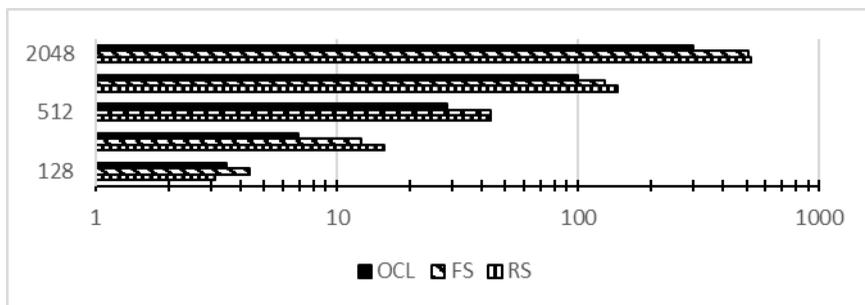


Рис. 4. Суммарное время работы алгоритма с учетом копирования данных

На диаграмме, представленной на рис. 5 можно видеть суммарное время для обработки изображений всех разрешений. OpenCL быстрее всего выполняет работу алгоритма. Временные издержки на копирование данных велики для всех алгоритмов, однако для OCL суммарные задержки на чтение/запись несколько ниже.

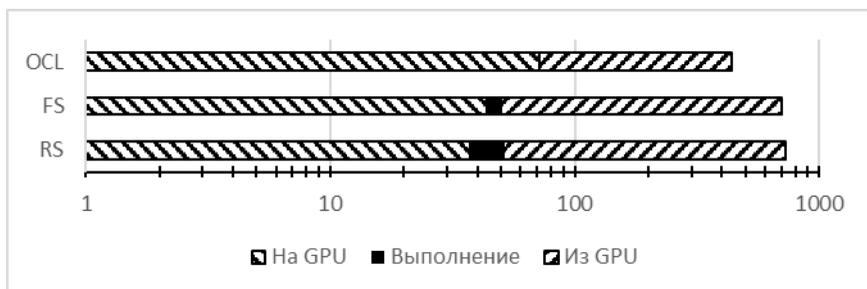


Рис. 5. Суммарное время для всех размерностей

Проведем сравнение полученных результатов времени работы алгоритма (рис. 6).

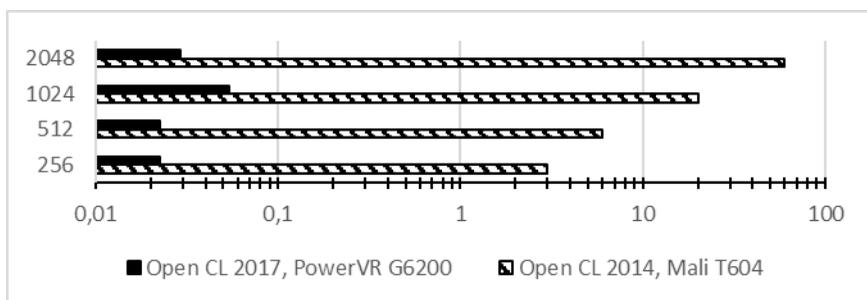


Рис. 6. Сравнение времени работы алгоритма фильтрации

Как можно видеть, временные затраты на обработку изображения значительно сократились. Более того, временные затраты на обработку изображения размерностью 2048 пикселей незначительно выше затрат на обработку изображения в 256 пикселей.

Заключение

В рамках данной работы были изучены актуальные средства для осуществления высокоэффективных вычислений на смартфонах Android и проведен анализ их производительности. Технология OpenCL является мощным кроссплатформенным средством для применения GPGPU подхода при разработке алгоритмов и показывает значительный прирост производительности за последние несколько лет.

Список литературы

1. Goodfellow, I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – MIT Press, 2016. – 785 p.

2. Getting Started with the Native Development Kit [Электронный ресурс] : Android Developers. – Режим доступа : <https://developer.android.com/ndk/guides/index.html>
3. Kim, H. Performance Analysis and Tuning for General Purpose Graphics Processing Units (GPGPU) / H. Kim, R. Vuduc, S. Bagsorkhi. – Morgan & Claypool Publishers, 2012. – 96 p.
4. Computation with the RenderScript [Электронный ресурс] : Android Developers. – Режим доступа : <https://developer.android.com/guide/topics/renderscript/compute.html>
5. The Standard for Embedded Accelerated 3D Graphics [Электронный ресурс] : Khronos Group. – Режим доступа : <https://www.khronos.org/opengles/>
6. Munshi A. OpenCL Programming Guide / A. Munshi [и др.]. – Addison-Wesley Professional, 2011. – 648 p.
7. Konrad M. Parallel Computing for Digital Signal Processing on Mobile Device GPUs / M. Konrad. – University of Applied Sciences, 2014. – 108 p.

ТОНКАЯ НАСТРОЙКА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

А.А. Ильина

Студент

В.С. Тарасов

Ассистент

Введение

Технологии в современном мире развиваются быстрыми темпами. Все большую актуальность приобретает возможность создания искусственного интеллекта, одним из направлений развития которого являются искусственные нейронные сети (ИНС). Они используются в медицине, экономике, интернете, робототехнике, охранных системах и многих других областях.

Искусственные нейронные сети являются математической моделью, вобравшей в себя принципы организации нейронной сети живого организма. Они состоят из искусственных нейронов, которые обмениваются сигналами и способны решать сложные задачи за счет взаимодействия.

Существенным отличием ИНС от алгоритмов является возможность их обучения. Это позволяет им обобщать входные данные, анализировать новые и принимать решения, основываясь на предыдущем опыте. Они обладают и другими преимуществами, такими как: адаптация к изменениям окружающей среды, устойчивость к шумам. Но их использование связано и с решением многих вопросов, например, выбор топологии сети, подбор характеристик сети, проверка результатов обучения, а также формирование набора данных для обучения нейронной сети.

1. Цель работы

Цель работы состоит в исследовании зависимости точности тонкой настройки нейронной сети от размера обучающей выборки на основе весов предобученной модели.

Результаты работы нейронных сетей в большинстве своем зависят от качества формирования обучающей выборки. Существуют такие наборы данных, как CIFAR-10, CIFAR-100, ImageNet, содержащие в себе тысячи изображений, но такого количества данных может быть недостаточно, чтобы обучить сеть распознавать объекты в частных случаях, что приводит к задаче исследования тонкой настройки нейронной сети.

2. Тонкая настройка глубокой сверточной нейронной сети

Ключевой особенностью глубоких сверточных нейронных сетей (ГСНС) является многослойность. Их преимущество состоит в способности самостоятельно находить признаки, по которым формируются выходные данные [1]. Обобщение признаков с увеличением уровня позволяет переходить от выделения таких признаков как контрастность, границы, яркость, углы и т.д. к более общим – частям лица, таким как: нос, глаза, рот, и далее объединять и их. Пример представлен на рис. 1.



Рис. 1. Обобщение признаков нейронной сетью

Качество работы нейросети зависит от обучающей выборки на которой ее тренировали. Существующие наборы данных CIFAR-10 и CIFAR-100 имеют 10 классов по 6000 изображений и 100 классов по 600

изображений соответственно, что не охватывает частные случаи. Большим объемом данных располагает ImageNet, включающий в себя больше 14 миллионов изображений для более чем 21 тысячи классов, но и этот объем данных не полон.

Тонкая настройка позволяет обучить нейронную сеть распознавать специфические признаки определенных объектов. Так, например, могут не распознаваться люди с татуировками, шрамами, в специфичных условиях окружающей среды.

При инициализации ГСНС изначальные веса берутся малыми случайными величинами [2]. На обучение сети уходит большое количество времени, которое растет с увеличением объема данных и количества слоев.

Тонкая настройка состоит в использовании весов заранее обученной ГСНС для доучения на ее основе распознаванию частных случаев, что существенно экономит время и позволяет проделать эту работу на меньших вычислительных мощностях [3]. При переиспользовании весов можно настроить, веса каких слоев мы продолжаем настраивать, а каких оставляем без изменений.

Примером некорректной работы нейронной сети может служить изображение, приведенное на рис. 2, которое нейронная сеть распознала как компьютерную мышь с вероятностью 0.3167. Далее с вероятностью 0.1238 она распознала породу собак Шелти, а после, с вероятностью 0.0915 – клавиатуру компьютера и т.д. Класс кошка оказался только на 7 месте с вероятностью 0.0410, что говорит о необходимости тонкой настройки сети.

Класс: мышь, компьютерная мышь, вероятность: 0,3167

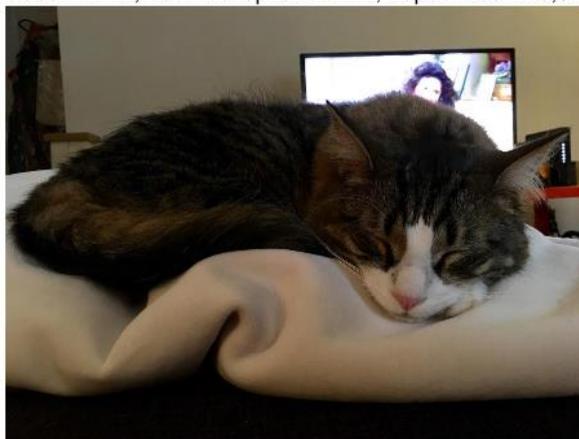


Рис. 2. Некорректно распознаваемое изображение кошки

3. Формирование выборки данных

Тонкая настройка сети требует наличия набора данных для обучения. Формирование выборки данных было решено провести, используя API социальной сети Instagram, в которую пользователи выкладывают специфические фотографии с наложенными эффектами, искажениями, засветами и т.д.

Тонкая настройка глубоких сверточных нейронных сетей может происходить по четырем сценариям:

- Новый набор данных меньше по размеру и аналогичен по содержанию исходному набору данных.
- Новый набор данных относительно крупный и аналогичен по содержанию исходному набору данных.
- Новый набор данных меньше по размеру и существенно отличается по содержанию от исходного набора данных.
- Новый набор данных относительно крупный и существенно отличается по содержанию от исходного набора данных.

В случае формирования массива данных для классов кошки и собаки из Instagram ситуация подходит под последний случай. Это значит, что выборка достаточно велика, чтобы обучить сеть с нуля, но это справедливо не для всех классов и на практике выгоднее использовать веса предобученной нейронной сети для инициализации модели (рис. 3).

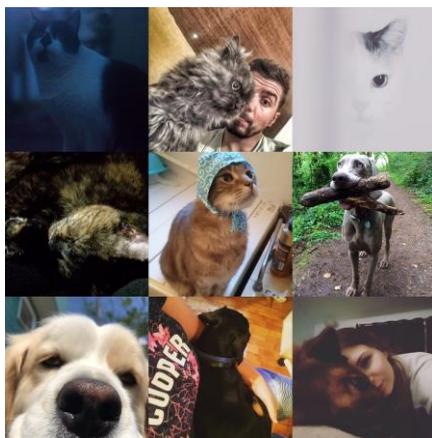


Рис. 3. Часть выборки данных

Данные для увеличения качества обучающей выборки необходимо нормализовать и обработать. В качестве обработки фото проводилось выравнивание гистограммы и изменение размера фото (рис. 4).



Рис. 4. Обработка изображений

Выравнивание гистограммы – один из способов регулировки контрастности изображения.

4. Результаты

В качестве нейронной сети была взята AlexNet, имеющая пять сверточных слоев и три полносвязных, реализованная на платформе Tensorflow [4]. Модель доучалась на выборках в 500, 1000, 2000 и 4000 изображений двух классов, кошки и собаки. Последние два слоя были взяты в качестве доучающихся. Веса остальных были оставлены без изменений (рис. 5).

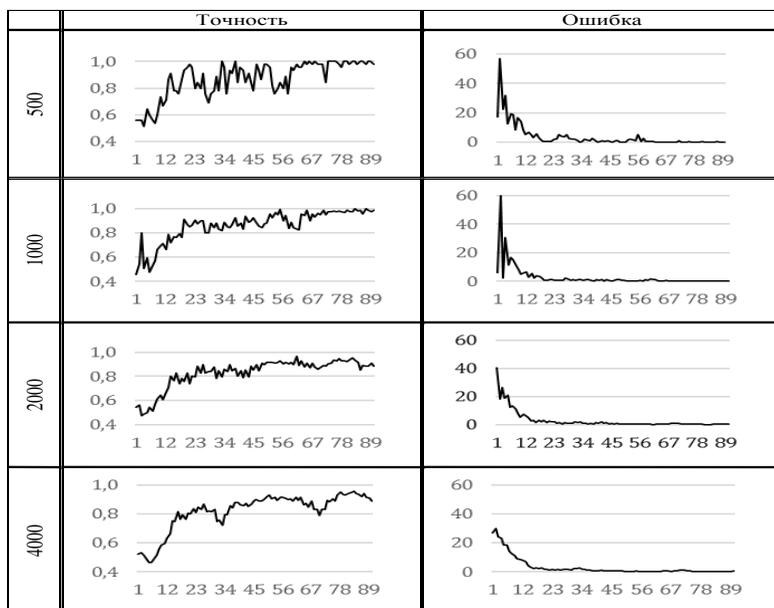


Рис. 5. Обработка изображений

Так можно видеть, что результаты тонкой настройки с увеличением количества изображений дают более плавное изменение результатов, что говорит о большей стабильности этих данных. Но, анализируя результаты, полученные при использовании выборок в 2000 и в 4000 изображений, обучение на которых привело к точности в 88.6 и 88.9 соответственно, можно говорить о том, что с определенного момента увеличение объема выборки перестает давать значительное улучшение результата в сравнении со временем на обучение, которое растет экспоненциально и составило 32.34 и 81.54 минуты соответственно.

Заключение

Из полученных результатов можно говорить о том, что тонкая настройка применима для обучения нейронных сетей частным случаям. Увеличение размера обучающей выборки до определенного момента положительно влияет на качество тонкой настройки сверточной нейронной сети, а дальнейшее увеличение не дает значительного прироста в точности в сравнении с ростом времени.

В дальнейшем планируется провести исследование зависимости качества тонкой настройки ГСНС от размеров изображения, накладываемых фильтров.

Список литературы

1. Goodfellow I. Deep Learning (Adaptive Computation and Machine Learning series) / I. Goodfellow, Y. Bengio , A. Courville. – Cambridge, MA : MIT Press, 2016. – 800 p.
2. Neural Networks and Deep Learning [Электронный ресурс] : книга. / М. А. Nielsen. – Режим доступа : <http://neuralnetworksanddeeplearning.com>
3. Camposa V. From Pixels to Sentiment: Fine-tuning CNNs for Visual Sentiment Prediction / V. Camposa, B. Joub, X. Giro-i-Nieto // The Cornell University Library. – 2017. – 10 p.
4. Géron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems / A. Géron. – Sebastopol : O'Reilly Media, 2017. – 566 p.

РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ ДЛЯ МОНИТОРИНГА ОКРУЖАЮЩЕЙ СРЕДЫ ВНУТРИ ПОМЕЩЕНИЯ

Д.В. Исакова

Студент

П.С. Лысачев

Старший преподаватель

Введение

В последние годы наблюдается тенденция к миниатюризации устройств. Компьютеры становятся всё меньше, телефоны – всё тоньше, появляются достаточно мощные микропроцессоры, превосходящие по мощности ЭВМ прошлого века, которые занимали целые этажи помещений.

Безусловно, это способствует проникновению «умных устройств» в нашу жизнь. Под «умными устройствами» подразумеваются предметы обихода, которые помимо своих стандартных функций обладают способностью анализировать данные, поступающие на вход, и отображать их в комфортном для человека виде и/или по результатам анализа предпринимать какие-либо действия.

Разработка сервера для обработки данных, отправляемых с устройства посредством Wi-Fi соединения, значительно увеличивает отказоустойчивость системы, а также облегчает контроль ее работы.

1. Принцип работы системы

Система состоит из двух взаимодействующих модулей: устройства и сервера (рис. 1). В каждом цикле своей работы устройство считывает и обрабатывает показания подключенных к нему датчиков, а также анализирует так называемые GSM-команды пользователя, формат которых заранее определен. Далее устройство формирует и отправляет пользователю ответ посредством sms-сообщения, если в этом есть необходимость, после чего показания датчиков, дата измерения и действия устройства отправляются на сервер в формате, который так же предопределен.

Сервер получает указанные выше данные и при необходимости информирует пользователя о нештатной ситуации.

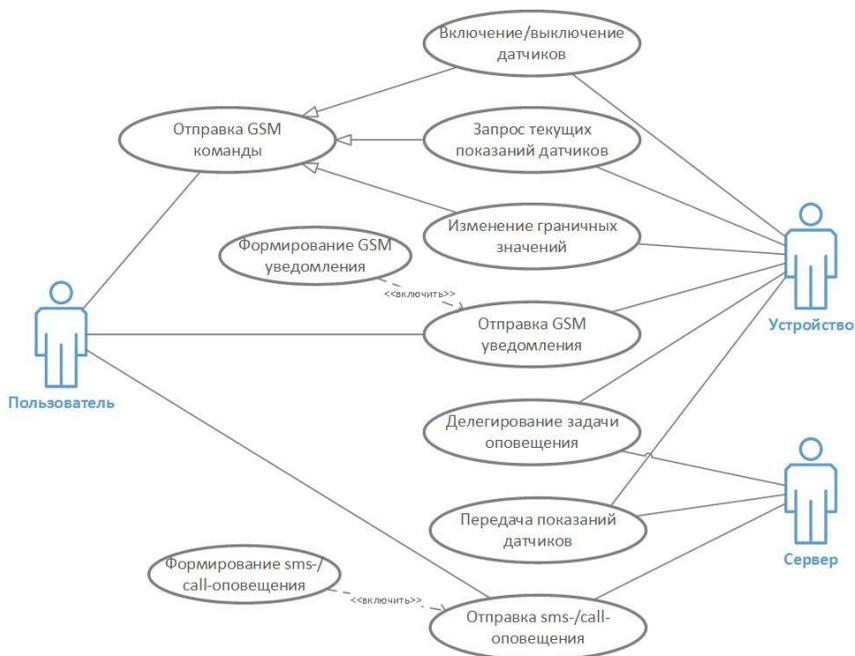


Рис. 1. Диаграмма вариантов использования системы для мониторинга окружающей среды внутри помещения

В данной статье наиболее подробно рассматривается серверная часть системы, поэтому далее будет рассмотрен состав и специфика работы исключительно этой части системы.

2. Входные и выходные данные сервера

Входные данные сервера – показания датчиков и служебная информация, поступающие с устройства. Служебная информация – это действия, предпринятые устройством после анализа показаний датчиков и GSM-команд.

На сервере они обрабатываются, дополняются новой служебной информацией и добавляются в файл резервной копии данных. Сервер способен генерировать и отправлять sms- и call-уведомления с целью оповещения пользователя (рис. 2).



Рис. 2. Входные и выходные данные сервера

Таким образом, на выходе сервера формируется файл резервной копии данных, а также sms- и call-оповещения.

3. Логическая архитектура сервера

Серверная часть реализована на языке С# [1] и расположена в облаке Azure. Эта часть системы занимается хранением полученной информации в виде так называемого log-файла. Этот файл содержит полученные от устройства данные, включающие показания датчиков и действия, которые были осуществлены после анализа текущих параметров среды. В ходе обработки сообщений, полученных от устройства, сервер способен информировать пользователя с помощью альтернативных средств: Nexmo, Twitter.

С помощью сервиса Nexmo [2] он может отправлять пользователю sms- и call-уведомления. При необходимости создания уникального call-уведомления сервер генерирует его, используя сервис Ivona [3]. Сообщения могут быть сгенерированы на английском, французском и немецком языках. К сожалению, на момент написания программы не было библиотеки, позволяющей работать с Nexmo Voice на языке С#, поэтому связь с Nexmo API прописывалась самостоятельно.

По желанию пользователя сервер способен публиковать твиты, содержащие показания датчиков и предпринятые действия (рис. 3).

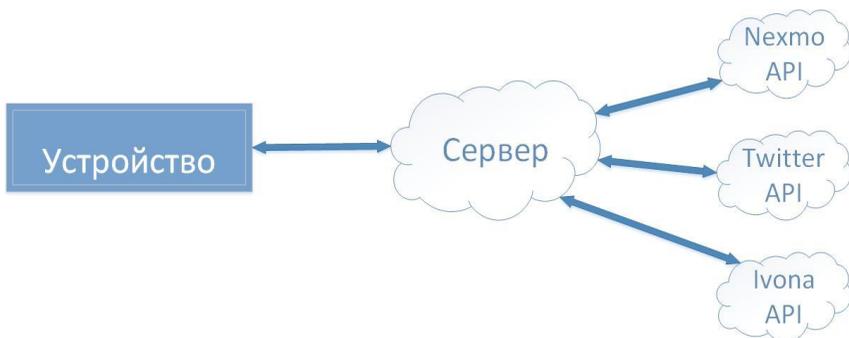


Рис. 3. Архитектура системы

Сервер – это программная часть системы, способная к взаимодействию с интернет-сервисами. Аналогично устройству, он может анализировать поступающие данные и предпринимать решения в зависимости от текущей ситуации. Именно на сервере хранятся все данные, сгенерированные системой: дата и время измерения показаний датчиков, сами показания и предпринятые действия (рис. 4).

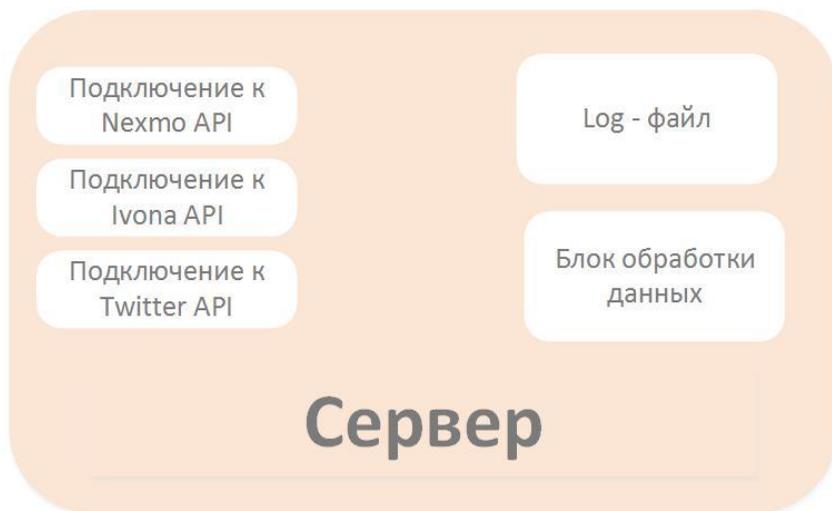


Рис. 4. Логическая схема сервера

4. Программная архитектура сервера

Диаграмма классов представлена на рис. 5:

1. Program – основной класс программы, в котором осуществляется запуск сервера, обращение к классам NexmoAPI и IvonaAPI, а также подключение к Twitter API. [4]
2. NexmoAPI – класс, который подключается и работает с Nexmo API.
3. NexmoResponse – класс, содержащий структуру формируемого запроса к сервису Nexmo.
4. SmsParametres – структура, содержащая все необходимые параметры для отправки sms-сообщения с помощью сервиса Nexmo.
5. IvonaAPI – класс, который подключается и работает с Ivona API.

Перечисления:

1. GenderOfVoice – мужской/женский голос.
2. Language – перечисление поддерживаемых языков.

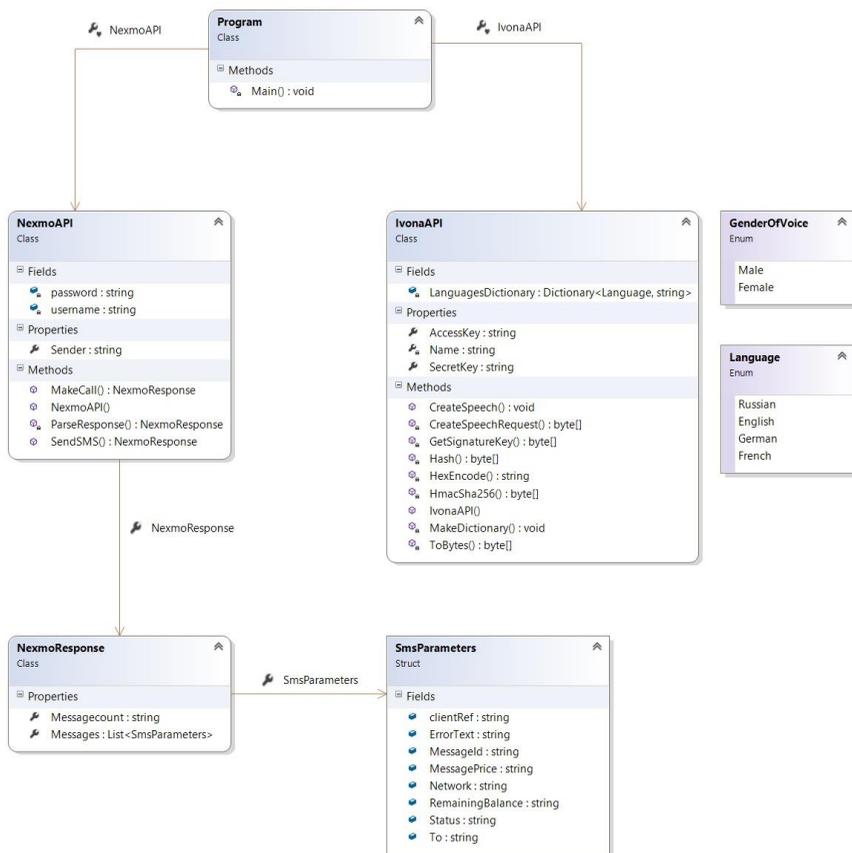


Рис. 5. Диаграмма классов сервера

Заключение

Данная статья посвящена разработке серверной части системы мониторинга окружающей среды внутри помещения. Описано взаимодействие подобного рода систем с интернет-сервисами, помогающими значительно увеличивать отказоустойчивость системы и информировать пользователя о внештатных ситуациях не одним, а сразу несколькими способами.

По завершении работ был разработан сервер, который выполняет следующие функции:

- принимает показания устройства
- поддерживает отправку sms-сообщений и голосовые вызовы (call-оповещения) через интернет сервис;
- подключается к Twitter API и публикует сообщения с показаниями датчиков.

Список литературы

1. Schildt Н. С# 4.0: The Complete Reference / Н. Schildt 2010. – The McGraw-Hill, 2010. – 895 с.
2. Nexmo [электронный ресурс]: nexmo docs. – Режим доступа : <https://docs.nexmo.com/voice>
3. Ivona, An Amazon Company [электронный ресурс]: Text to Speech Ivona. – Режим доступа: <https://www.ivona.com>
4. Twitter [электронный ресурс]: Twitter Developer Documentation. – Режим доступа: <https://dev.twitter.com/rest/public>

МОДИФИКАЦИЯ МЕТОДА ВИОЛЫ-ДЖОНСА НА ОСНОВЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ЦВЕТА КОЖИ

Г.И. Казун
Студент
А.А. Сирота
Профессор

Введение

Современные технологии распознавания образов решают сложные задачи, предоставляя удобный инструмент для обнаружения необходимых объектов. При обработке изображения достаточно большого размера его анализ может занять значительное время, а также может привести к ложным срабатываниям детектора.

Многие задачи, выполняемые на персональных компьютерах, связаны с быстрой обработкой информации. Существует технологии, позволяющие с помощью GPU ускорить процесс вычислений.

Для увеличения эффективности детектора нужно отсеять области изображения, на которых не изображена кожа.

В данной статье рассматривается задача разработки существующих алгоритмов обнаружения цвета кожи, переопределение фильтров для данных алгоритмов, модификация разработанных методов Виолы-Джонса с помощью данных алгоритмов и их сравнительный анализ с использованием различных входных данных.

Для решения поставленных задач использовалась машина со следующими техническими характеристиками:

- Материнская плата: ASUS Z87M-PLUS
- Процессор: Intel Core i7 4770K 3.50GHz
- Видеокарта: NVIDIA GeForce GTX 970 4Gb
- ОЗУ: DDR3 16Gb

1. Метод Виолы-Джонса

Метод был разработан и представлен в 2001 году Полом Виолой и Майклом Джонсом.

Основные принципы, на которых основан метод:

- Изображение в интегральном представлении
- Признаки Хаара
- Каскады признаков
- Сканирующее окно

В методе Виолы-Джонса сканирующее окно перемещается следующим образом (рис. 1). Окно принимает минимальный размер, далее оно с определённым шагом передвигается по горизонтальной линии изображения. Достигнув границы изображения, окно передвигается на шаг по вертикали и проходит следующую горизонтальную линию. Этот цикл будет повторяться пока не будет достигнута граница изображения по вертикали, после чего сканирующее окно увеличится с помощью заданного коэффициента и будет заново проходить по изображению. Его увеличение будет происходить до тех пор, пока размер не станет равен размеру изображения.

Данный метод можно модифицировать с помощью параллельных вычислений на основе технологии CUDA. CUDA — архитектура параллельных вычислений, которая позволяет значительно увеличить вычислительную производительность систем благодаря использованию графических процессоров NVidia. GPU — отдельное устройство в персональном компьютере, ноутбуке, мобильном устройстве или игровой приставки, выполняющее вычисления, предназначенные для обработки и вывода графической информации. С помощью специально разработанной параллельной архитектуры они в разы превосходят стандартные центральные процессоры в обработке графической информации.

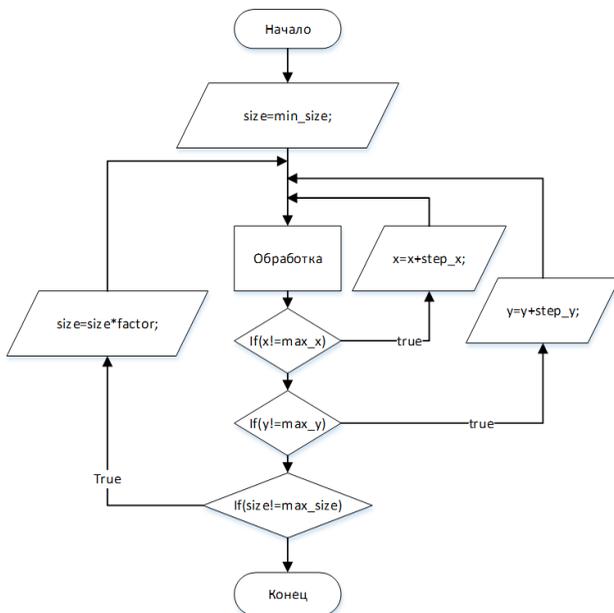


Рис. 1. Структурная схема алгоритма сканирующего окна стандартного метода

2. Методы обнаружения цвета кожи

Первым методом обнаружения цвета кожи [1] является фильтрация пикселей изображения по критериям для цветового пространства RGB: $R > 90$, $G > 40$, $B > 20$, $R - G > 15$, $R > B$. Данные условия были получены в ходе анализа заранее подготовленной выборки.

Второй метод основывается на цветовых пространствах Log opponent и YIQ [2]. Цветовое пространство log opponent использует противоположные оттенки: желто-синий и зелено-красный, YIQ применяется в телевидении. В первом используются составляющие I, Rg, Bu, из которых получаем H, во втором YIQ. Критерии: $100 < H < 150$, $20 < I < 90$.

Третий метод основывается на цветовых пространствах HSV и YIQ [2]. HSV было создано Элви Реем Смитом, одним из основателей Pixar, в 1978 году. Оно является нелинейным преобразованием модели RGB. Критерии: $0.20 < S < 0.75$, $V > 0.35$, $0 < H < 25$, $20 < I < 90$. Данный метод модифицированный, изначально в нем не использовалось YIQ.

3. Анализ, реализация и модификация известных результатов в области обнаружения объектов методом Виолы-Джонса

Был проанализирован и реализован алгоритм, основанный на версии метода Виолы-Джонса библиотеки OpenCV.

Далее, используя технологию CUDA [3, 4], можно привести получившийся алгоритм к следующему виду (рис. 2). Алгоритм распознавания разбивается на блоки, каждому из которых отводится уникальный размер распознающего окна. Каждый блок итеративно проходит изображение по вертикали и с помощью параллельных нитей анализирует каждое окно, расположенное на горизонтальной линии.

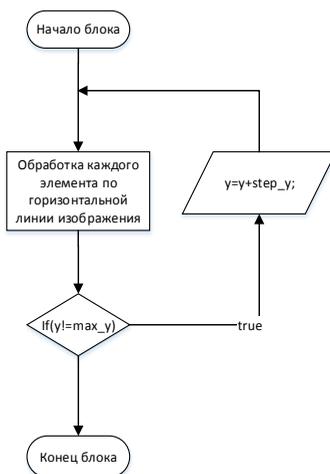


Рис. 2. Структурная схема алгоритма сканирующего окна, основанном на параллельной обработке размеров окна и строк

Так же на основе статьи [5] можно реализовать алгоритм (рис. 3), в котором размер сканирующего окна увеличивается итеративно, а на обработку каждой линии по вертикали отведён параллельный блок, в котором нити одновременно по горизонтали сканируют элементы.

Перед работой данных алгоритмов выполняется обработка изображения с помощью методов обнаружения кожи. Все пиксели, не прошедшие фильтрацию, закрашиваются черным. После чего, изображение сканируется окном 20 x 20 пикселей (минимальный размер сканирующего окна метода Виолы-Джонса). Если количество пикселей с цветом кожи в данном окне больше заданного порога, то оно полностью восстанавливается из исходного изображения, так как в данной области возможно расположение искомого объекта. Далее, перед выполнением

распознавания происходит схожая процедура для принятия решения сканирования этого участка или его игнорирования.

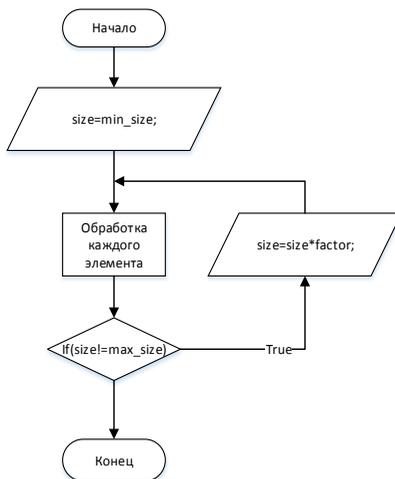


Рис. 3. Структурная схема алгоритма сканирующего окна, основанном на параллельной обработке каждого элемента

4. Анализ полученных методов с различными входными данными

Для наглядного получения результатов обработки изображения методами фильтрации по цвету кожи используется цветовой спектр (рис. 4).

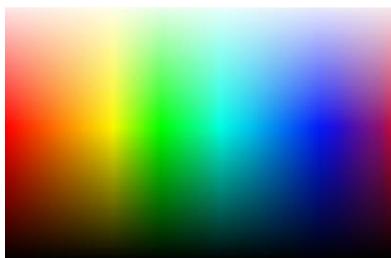


Рис. 4. Цветовой спектр

Применение методов к изображению с цветовым спектром (рис. 5).



Рис. 5. RGB, Log opponent и YIQ, HSV, HSV и YIQ

Модификация алгоритма распознавания с помощью методов обнаружения кожи ускоряет процедуру и уменьшает количество ложных обнаружений. Немаловажным фактором для этого является восстановление обрезанных областей, так как после поиска областей с цветом кожи часто появляется сетка, которая воспринимается алгоритмом как искомый объект.

В данной работе для детального изучения ложных срабатываний рассматривается метод Виолы-Джонса, в котором используется минимальный шаг между сканирующими окнами, минимальный порог для обработки области, а также не уменьшается исходное изображение.

GPU без использования параллельных вычислений не является эффективным средством для ускорения алгоритма, поскольку результаты однопоточного алгоритма распознавания методом Виолы-Джонса в разы хуже, чем тот же алгоритм, выполняемый на CPU, например, для анализа изображения размером 1024 x 1024 графическому процессору потребовалось 57 секунд.

При обработке изображения размером 1024 x 1024 сканирование на CPU Intel заняло 2.8 секунды, в то время как методу №1, основанному на параллельной обработке размеров окна и строк, потребовалось 2.9 секунды, а методу №2, основанному на параллельной обработке каждого элемента, 1.5 секунды. Более детальная информация расположена в табл. 1. Преимущество метода №2 в данном тесте заключается в меньшем количестве итеративных циклов прохождения изображения сверху вниз, чем в методе №1.

Таблица 1

Показатели затраченного времени

Тип процессора	Среднее время, с (с поиском кожи в среднем)	Среднее время, с (без поиска кожи)
I7 4770	3.1	2,8
GTX 970 №1	2,7	2,9
GTX 970 №2	1,4	1,5

При значительном увеличении изображения до 4096 x 4096 пикселей метод №2, выполняемый 21.5 секунды уступает методу №1 в

быстродействию, который выполнен почти в 2 раза быстрее по сравнению с ним и в 4 раза быстрее по сравнению со средним значением затраченного времени на CPU. Более детальная информация расположена в табл. 2.

Таблица 2

Показатели затраченного времени

Тип процессора	Среднее время, с (с поиском кожи в среднем)	Среднее время, с (без поиска кожи)
I7 4770	56.7	43.4
GTX 970 №1	8.6	10.3
GTX 970 №2	17.9	21.5

При использовании методов поиска цвета кожи время работы практически не отличается между ними, за исключением фильтрации по RGB. Данный метод работает немного быстрее на фоне остальных, но он пригоден только для европеоидной расы. В отличие от быстродействия, заметно возрастает эффективность детектора, результаты анализа которого расположены в табл. 3.

Таблица 3

Сравнение эффективности

Размер изображения	Найдено объектов в среднем	Ложные в среднем (без поиска кожи)	Ложные в среднем (с поиском кожи)
1024 x 1024	82	8	3
4096 x 4096	1240	132	29

Заключение

В рамках данной работы были реализованы алгоритмы распознавания образов Виолы-Джонса с помощью параллельных вычислений и модифицированы с помощью методов поиска цвета кожи, проанализированы результаты с использованием различных входных данных.

С помощью параллельных вычислений, основанных на технологии CUDA, можно добиться значительного прироста в быстродействии задач. Так, например, скорость распознавания образов методом Виолы-Джонса можно сократить в 3-5 раз в зависимости от входных данных. Что касается метода, основанном на параллельной обработке сканирующих окон и элементов по горизонтали, по сравнению с методом, обрабатывающим каждый элемент параллельно, можно подчеркнуть, что эффективность

первого появляется только при обработке крупных изображений, так как он не обрабатывает каждое окно независимо, а сконцентрирован на парализации увеличения сканирующего окна, которое зависит от размеров входного изображения.

Модификация с помощью методов поиска цвета кожи увеличивает качество работы алгоритма за счет значительного снижения ложных срабатываний детектора, так как отбрасываются участки изображения с оттенками, не проходящих фильтрацию по цвету.

Список литературы

1. Обнаружение лиц на цветном растровом изображении с применением модифицированного метода Виолы-Джонса [Электронный ресурс] : Молодежный научно-технический вестник / Моск. гос. тех. унив. им. Н.Э. Баумана. – Электрон. журн. – Режим доступа: <http://www.technomag.bmstu.ru/authors/230274.html>

2. Tarek Abd El Hafeez. A New System for Extracting and Detecting Skin Color Regions from PDF Documents / Tarek Abd El Hafeez // (IJCSE) International Journal on Computer Science and Engineering. – 2010. – №9. – С. 2838-2846.

3. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учебное пособие / А. В. Боресков [и др.]. – М. : Издательство Московского университета, 2012. – 336 с.

4. Боресков А. В. Основы работы с технологией CUDA / А.В. Боресков, А. А. Харламов. – М. : Издательство ДМК-Пресс, 2010. – 232 с.

5. Акимов А. В. Разработка и исследование алгоритмов распознавания изображений на основе метода Виолы-Джонса с использованием технологии вычислений на графических процессорах CUDA / А.В. Акимов, А.А. Сирота // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2014. – № 3. – С. 100-109.

АНАЛИЗ ТОНАЛЬНОСТИ ТЕКСТА

К.С. Калабухов

Студент

В.В. Гаршина

Доцент

Введение

К наиболее актуальным задачам, из числа находящихся в области исследования и анализа текстовых данных, относится определение эмоциональной окрашенности текста, позволяющая извлекать из него мнение человека определённой полярности, направленное к объекту высказывания.

Применение методов, направленных для анализа тональности текстовой информации реализуется не только в качестве научно-исследовательских инструментов, но и в коммерческой деятельности.

Активное использование подходов и решений, применяемых при анализе тональности текстов, отмечается в следующих областях и сферах:

- Создание систем рекомендаций
- Аналитика общественного мнения
- Извлечение отношения пользователя к продукту

На сегодняшний день, проведение исследований в данной области является актуальным направлением, в связи с тем, что какое-то либо единое решение данной проблемы отсутствует.

Проведение многочисленных конференций международного уровня, в частности: «TACL», «ICAICT», «Диалог», «АИСТ», относительно проблем компьютерной лингвистики и анализа тональности текста, подтверждают данный факт.

1. Подходы для выделения эмоциональной окрашенности текста

Для выделения бинарной эмоциональной окрашенности текста в автоматическом режиме используется несколько следующих подходов:

- Основанные на лингвистических правилах;
- Основанные на методах машинного обучения.

Подход, основанный на правилах, для большого количества исследователей различного уровня зачастую является основоположным.

Алгоритмы, создаваемые на основе данных правил, позволяют учитывать семантические, структурные особенности различных слов и самого языка.

Применение подобного подхода с учётом данных особенностей, как правило, предоставляет достаточно высокие результаты в показателях точности и полноты, в сравнении с методами машинного обучения.

Однако, следует учитывать то, что данный подход также несёт за собой ряд различных существенных недостатков:

- необходимость в формировании корпуса лингвистических правил;
- незначительная область охвата различных текстовых конструкций;
- привязка к языку анализируемого текста.

Использование подхода, основанного на лингвистических правилах, может обеспечить высокие показатели результативности лишь в тех случаях, когда анализируемые тексты будут грамматически верны, а так же если различные конструкции анализируемого языка будут покрыты корпусом правил.

Подход, который основывается на методах машинного обучения [1] подразумевает наличие некоторого набора входных данных, применяемых для обучения классификатора, в свою очередь делится на:

- обучение без учителя;
- обучение с учителем.

Использование метода обучения без учителя подразумевает распределение текстов на определенное число кластеров, которое задаётся заранее. Впоследствии производится присвоение соответствующей метки, характеризующей полярность в режиме автоматического или ручного формата для каждого из кластеров.

При применении метода обучения с учителем, требуется наличие текстового корпуса, который заранее размечается метками полярности, в свою очередь идентифицирующими полярность для каждого текста из корпуса, а определение тональности производится непосредственно автором текста, либо экспертом или их группой.

Применение подходов, основываемых на методах машинного обучения, позволяет подстраиваться под языковые особенности, включать в учёт дополнительные признаки, производить обработку текстов, которые, с точки зрения принятых языковых правил, являются грамматически неверными.

Именно поэтому подход, основанный на методах машинного обучения с учителем, используется в данной работе в качестве одного из основных инструментов для решения задачи выделения эмоциональной окрашенности текстов.

2. Классификационные алгоритмы

Для обеспечения работы классификатора необходимо, чтобы текст представлялся в форме вектора определённой длины N , где N -

является длинной словаря терминов [2], причем значение элемента i в векторном пространстве будет определяться, как вхождение термина i из словаря текста.

Для проведения эксперимента были выбраны следующие алгоритмы:

– Алгоритм наивного Байесовского классификатора с мультиномиальной моделью распределения, при применении которого рассчитывается вероятность появления термина в каком-либо из классов для каждого текста, впоследствии чего будет рассчитываться общая вероятность принадлежности текста к определённому классу. Предсказание принадлежности для новых текстов к какому-либо из классов будет осуществляться на основе перечисленных вероятностей.

– Алгоритм на основе метода k ближайших соседей, применение которого подразумевает расчёт косинусовой близости нового объекта между каждым из элементов из обучающей выборки. Происходит выбор k ближайших соседей, для присвоения метки для нового элемента, как наиболее встречаемого, после чего объект классифицируется.

3. Нормализация текстовых данных

Для повышения качества работы классификационного алгоритма и уменьшения размерности словарей терминов, реализовываются следующие этапы предварительной обработки:

- Текст приводится к нижнему регистру;
- Удаляются стоп-слова;
- Удаляются все символы, которые не являются буквенными;
- Удаляются повторяющиеся символы;
- Удаляются слова, которые встречаются реже всего;
- Производится процедура стемминга.

4. Программная реализация

В программном модуле `SentimentTextClassifier` производится реализация подхода, основываемого на методе машинного обучения с учителем, включающий в себя следующие этапы:

1. Процесс предобработки входных текстовых данных
2. Процесс извлечения из входных данных признаков, необходимых для представления информации в векторной форме.
3. Процесс обучения алгоритма классификации на обучающей выборке
4. Процесс классификации объектов, исходя из решающего правила алгоритма
5. Процесс тестирования качества классификации алгоритма

На рисунке представлена схема работы программного модуля анализа тональности текста, представляющая собой определённую последовательность действий при обучении модели, а также при непосредственном использовании вновь обученной модели.

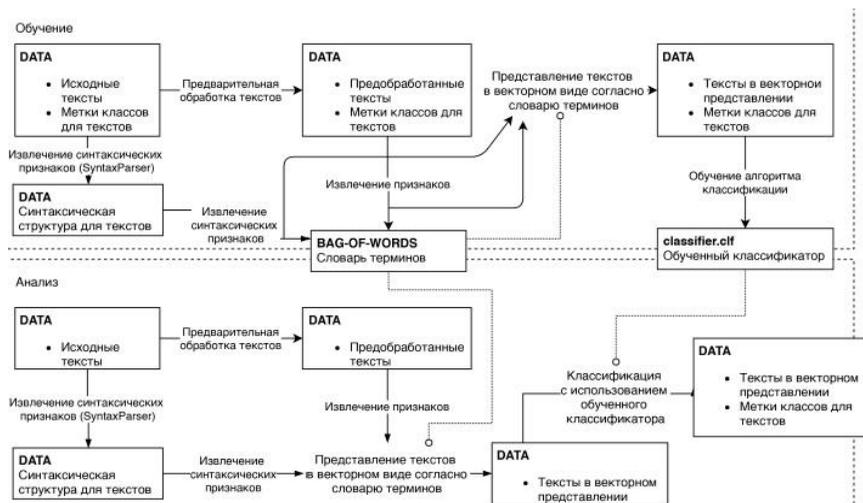


Рисунок. Схема взаимодействия частей программного модуля анализа тональности

5. Результаты эксперимента

Тестирование качества классификации эмоциональной окрашенности текстов производилось с использованием оценок метрик эффективности [3], выражаемых в критериях:

- точность;
- полнота;
- F-мера.

Для получения данных оценок использовался метод 10-fold cross validation, включающий в себя следующие этапы:

1. Разбиение обучающей выборки на k – количество частей
2. Обучение алгоритма на тестовом подмножестве
3. Тестирование качества на контрольном подмножестве
4. Получение усредненных результатов показателей качества классификации

В таблице приводятся наилучшие результаты экспериментов по оценке качества работы классификатора при использовании различных процессов для предварительной текстовой обработки.

Результаты экспериментов

Алгоритм	Параметры		10-fold cross validation		
	Обработка	Стемминг	Точность %	Полнота %	F-мера %
MNB	+	+	95,18	91,52	93,43
SVM	+	+	86,37	89,54	87,31
kNN	+	+	69,14	77,36	73,72

Исходя из полученных результатов, наиболее высокие показатели по критериям точности, полноты и F-меры демонстрирует наивный Байесовский классификатор с мультиномиальной моделью распределения.

Заключение

По результатам выполненной работы можно сделать следующие выводы:

- Формирование обучающей выборки должно производиться из текстов той же предметной области, для которой будет использоваться классификатор;
- Качество результатов, полученных при классификации, напрямую зависит от подобранной обучающей выборки;
- Применение процедур по предварительной текстовой обработке в значительной мере улучшают результат классификации.

Список литературы

1. Котельников, Е. В. Автоматический анализ тональности текстов на основе методов машинного обучения / Е.В. Котельников, М.В. Клековкина // Компьютерная лингвистика и интеллектуальные технологии: по материалам ежегодной Международной конференции «Диалог». – М. : РГГУ, 2012. – С. 15-21.
2. Поляков, П. Ю. Автоматическое определение тональности объектов с использованием семантических шаблонов и словарей тональной лексики / П.Ю. Поляков, М.В. Калинина, В.В. Плешко // Компьютерная лингвистика и интеллектуальные технологии: по материалам ежегодной Международной конференции «Диалог». – М. : РГГУ, 2015. – С. 68-76.
3. Оценка классификатора (точность, полнота, F-мера) [Электронный ресурс] / Сайт Дениса Баженова. – Режим доступа: <http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html>

КЛАССИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ ПО ТИПУ ПОЛУЧЕННОГО ВЫСШЕГО ОБРАЗОВАНИЯ НА ОСНОВЕ СООБЩЕНИЙ В СОЦИАЛЬНОЙ СЕТИ

Е.Д. Кашин

Студент

Д.И. Соломатин

Старший преподаватель

Введение

В настоящее время всё чаще применяется классификация пользователей по разным признакам. Например – по покупательской способности, по вероятности подключения нового тарифа, по демографической категории. Это производится с помощью анализа пользовательского поведения в интернете – сообщений, комментариев, списка покупок и даже личных фотографий.

С каждым годом пользователи Интернета, общаясь, просматривая новости, ставя лайки на фотографии, генерируют все больше информации, поэтому, для ее анализа применяются алгоритмы машинного обучения. Машинное обучение в задаче классификации подразумевает, что имеется множество объектов, разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающим. Классовая принадлежность остальных примеров не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

В данной работе рассматривается классификация пользователей на два типа по принципу полученного высшего образования:

- «гуманитарное»;
- «техническое».

Такая классификация пользователей может использоваться для целевой рекламы в социальной сети. Определяется категория пользователей, для которых будет интересно конкретное рекламное предложение. Например – более вероятно, что для пользователя из «технического» класса реклама курсов по программированию будет более релевантная, а для пользователя из «гуманитарного» с объявление о предстоящем литературном вечере.

Также, такая модель может применяться для проведения профориентации. Тестируемый пользователь отмечает группы, в которых он состоит, а модель, на основе его поведения, определяет, к какому классу

он вероятнее относится. В таком подходе, испытуемому не придется проходить нудные тесты и тратить время.

1. Поиск и получение необходимых данных

Для того, чтобы модели машинного обучения хорошо работали, необходимы большие объемы данных. В социальных сетях в свободном доступе находятся миллионы страниц, с которых можно собирать информацию. В данной работе использовались данные крупнейшей социальной сети в СНГ – «ВКонтакте».

Публичные страницы бывают разных категорий – спорт, природа, кино, знаменитости, автомобили, техника, кулинария, новости, дизайн и т.д. Страница должна быть нейтральной тематики для того, чтобы количество человек в двух классах, на которые будет происходить классификация, примерно совпадало. Если взять страницы про технику, то количество человек в классе «техническое» будет намного больше, чем в классе «гуманитарное», а если взять тематику про дизайн, то наоборот. Также количество комментариев под каждой записью должно быть как можно больше.

В качестве источника комментариев была выбрана публичная страница – «Типичный Воронеж» [1]. Её направленность – обсуждение последних новостей в городе. Аудитория – от школьников до пенсионеров. Из-за этого нет смещения в сторону одного из классов. Количество человек – около 300 тыс., количество публикаций – около 50 тыс., среднее количество комментариев к публикации – 20.

2. Получение данных

Существуют инструменты для автоматической сборки данных с социальных сетей. В основном они агрегируют информацию о странице (количество просмотров, количество лайков) и не позволяют скачать все данные. Другой вариант – воспользоваться Open API от «ВКонтакте» [2]. Это система для разработчиков, позволяющая получать доступ практически ко всем данным социальной сети.

Было загружено 2.6 млн. комментариев. Этого количества с запасом хватит для анализа. Но возникает вопрос о том, как же разделить все эти данные на два класса, чтобы потом обучить модель. Делается допущение, что классификацию по типу полученного образования можно проводить по указанной в профиле пользователя информации о факультете. Но в выборку также попадут люди, которые еще не закончили учиться.

Из 130 тыс. в выборке оказалось 11 тыс. человек, у которых указан факультет. К ним относится 100 тыс. комментариев. Определение класса по факультету было осуществлено вручную. Всего в выборке было около 150 различных факультетов. Каждому факультету приписывалась метка –

«технический», «гуманитарный» или «естественный». Разметка делалась несколькими людьми, а затем для факультета выбиралась метка, которую чаще всего ставили для него люди. С помощью такого способа уменьшался человеческий фактор. Пользователи с «естественных» факультетов, например, с «Факультета ветеринарной медицины и технологии животноводства», не рассматриваются в данной работе, так как образуют отдельный класс.

Несколько примеров из факультетов «технического» класса: «Факультет компьютерных наук», «Факультет прикладной математики, информатики и механики», «Физико-математический факультет». Несколько факультетов «гуманитарного» класса: «Факультет журналистики», «Психолого-педагогический факультет», «Факультет дизайна, политологии и социологии».

3. Анализ данных

Прежде чем приступить к построению модели, необходимо провести анализ имеющейся выборки. Это делается для того, чтобы понять, какие признаки нужно учитывать, как они распределены, какие модели лучше использовать.

Из 150 названий факультетов были отфильтрованы те, которые указаны менее чем тремя разными людьми. После этого получилось 44 «технических» факультета и 92 «гуманитарных», а также 4120 и 4050 человек в двух классах соответственно. Хотя «технических» факультетов значительно меньше, общее количество людей в них немного больше, но это различие не существенно, поэтому можно считать выборки сбалансированными.

В выборке пользователей мужского пола оказалось 55%. В обществе существует стереотип, что среди мужчин больше «технарей», а среди женщин – «гуманитариев». Данные это подтверждают. Среди пользователей мужского пола больше «технического» класса, а среди пользователей женского пола – «гуманитарного». Для того, чтобы модель не переобучилась на этот признак и не стала предсказывать вместо целевых классов – пол, необходимо сбалансировать количество мужчин и женщин в двух классах.

В данной работе, после тестирования, было принято решение использовать технику oversampling. Объекты из меньшего класса (для «технического» класса – пользователи женского пола, для «гуманитарного» – мужского) копировались до того момента, пока количество пользователей в двух классах не становилось одинаковым. Это позволяло не терять данные из большего класса.

Перед дальнейшим анализом можно попробовать прочитать сами комментарии. Далее представлены несколько примеров из обеих групп.

«Технический» класс: «Антон, ты когда-нибудь на заблокированных колесах останавливался? Какой тормозной путь получается?», «Срочно подскажите контакты максимально дешёвых типографий.», «поддержим своих, конечно) 21-го на Труде)», «Кому интересно – почитайте: Воронеж признан городом с лучшим климатом среди областных и краевых центров.», «Первое фото очень оригинально и весьма красиво!».

«Гуманитарий» класс: «Очень симпатичные котятки(☺☺☺', '=)))», «По-моему, делается всё для того, чтобы дети знали, как можно больше о наркотиках.», «я не успеваю за скоростью твоей мысли))))я безнадежен», «У нас в 19 школе только буфет, и кормят только малышню и спортсменов=/', «вчера можно было в 00.30 увидеть 2 луны одна из них это марс,к несчастью у меня были облака и наблюдать это я не смогла(((».

Видно, что комментарии написаны на разные темы. Можно постараться найти различия в затрагиваемых тематиках. Слова нужно будет привести к начальной форме, чтобы избежать множества похожих по смыслу. Также большое внимание нужно уделить используемым смайлам и пунктуации. Еще можно заметить, что в некоторых словах допущены грамматические ошибки, это также может быть хорошим признаком для модели.

4. Получение признаков

После анализа данных в задачах машинного обучения начинается этап построения и отбора признаков. Качество финальной модели напрямую зависит от количества и качества полученных признаков.

Прежде чем начать работу с текстом, его нужно токенизировать – разбить на слова. Это можно сделать с помощью стемминга или лемматизации. Стеemming – это выделение основы слова, откидывание окончания. Например, слова: «бегающий», «бегущий», «бегать» – приведутся к слову «бег». Лемматизация – это приведение слова к начальной форме [3]. Например, слово «климатом» приведет к слову «климат». В данной задаче лучше сработала лемматизация.

После этого необходимо привести слова в набор векторов, потому что большинство моделей машинного обучения принимают на вход численные вектора. Самый простой подход называется Bag of Words: создается вектор длиной в словарь, состоящий из всех слов, встречающихся во всех комментариях. Далее для каждого слова считается количество вхождений в текст и подставляется это число на соответствующую позицию в векторе.

При таком подходе не учитывается порядок слов в предложении. Чтобы решить эту проблему, можно использовать идущие подряд токены – N-граммы. Для случая со словами это будут словосочетания.

Такие же векторы признаков можно строить и на уровне символов. Текст разбивается на N-граммы символов, где N обычно берется равное 3.

Слово «лесной» разобьётся на: «лес», «есн», «сно», «ной». Этот подход показывает себя чаще не хуже, чем разбиение на уровне слов.

Другими источниками признаков стали общие характеристики комментария. Они описывают комментарий в целом, обобщая его.

Пунктуационные признаки: наличие прямой речи; количество запятых, знаков вопроса, восклицательных знаков, троеточий; количество смайлов вида «)», «))», «((». Признаки содержания: процент слов, написанных CapsLock-ом; процент английских слов; наличие большой буквы в начале предложения. Общие признаки: количество слов, предложений; средняя длина слова, предложения.

5. Используемые модели

После получения векторов признаков можно приступить к обучению моделей. Использование большего количества моделей предпочтительнее, так как это повышает точность. Строится несколько моделей, затем они опрашиваются по каждому объекту, который нужно классифицировать, и выбирается наиболее частое предсказание. Данный метод называется – «ансамбль моделей» или «голосование моделей».

Пользователи были разделены на две части в пропорции 80 к 20. На первой группе проводилось обучение моделей, на второй – тестирование. За метрику была взята точность – это отношение числа верно классифицированных примеров к общему количеству примеров [4].

Итоговая задача – классифицировать пользователей, но так как классификация осуществляется по текстовым комментариям, то модели будут классифицировать каждый комментарий по отдельности. Финальным предсказанием для пользователя будет заниматься другая модель.

Каждая модель будет обучаться на своем наборе признаков. Может быть несколько моделей одного типа, которые будут учиться на разных признаках. Чем более разные получатся модели, тем в итоге окажется лучше точность. Входными данными для следующих трех моделей будут закодированные слова и закодированные символы. Каждой модели будет по два типа, один обучается только на словах, другой – только на символах.

Логистическая регрессия – линейная модель, которая каждому входному признаку присваивает вес. Предсказание делается, учитывая признаки с этими весами.

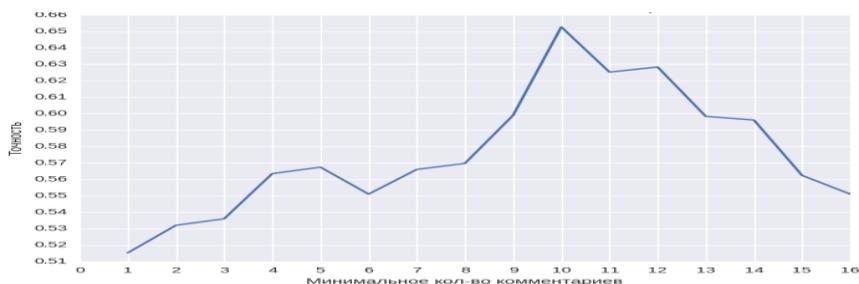
Многослойная нейронная сеть – модель, которая состоит из нескольких слоев с нейронами. Предсказания получаются нелинейными.

Рекуррентная нейронная сеть – сеть, которая была специально создана для моделирования последовательностей (слов, символов).

Последняя модель – градиентный бустинг деревьев. В отличие от предыдущих моделей, она обучается на общих характеристиках комментария (количество запятых, средняя длина слова...). Этот алгоритм показывает очень хорошие результаты в случае работы с множеством категориальных признаков.

Так как нужно строить предсказание для пользователя, а не для комментария, то финальная модель агрегирует предсказания этих моделей. Для пользователя строится матрица N на M , где N – количество комментариев у данного пользователя, M – количество моделей. Для всех комментариев отдельного пользователя делается предсказание с помощью каждой модели, результат записывается в матрицу. Финальный ответ для пользователя – наиболее частое предсказание.

Качество модели зависит от минимального количества комментариев, которое будет использоваться для классификации одного пользователя. В теории, чем больше комментариев, тем лучше. Однако в используемом наборе комментариев (описан выше) при увеличении кол-ва комментариев одного человека, уменьшается число человек, имеющих данное или большее кол-во комментариев, т.е. уменьшается размер обучающей выборки. Например, 2 и более комментария имеют примерно 4000 человек, а 10 и более комментариев – менее 1000. Этим объясняется снижение точности классификации при увеличении количества комментариев одного пользователя более 10-ти в проведенных экспериментах (см. Рис. 1).



с увеличением количества комментариев уменьшается количество пользователей в обучающей выборке

Рис. 1. Зависимость точности от количества комментариев одного пользователя

Заключение

В процессе работы были исследованы методы классификации текстов. В результате получилась модель, которая может с некоторой степенью точности классифицировать пользователей по типу полученного высшего

образования на основе их комментариев на два класса – «техническое» или «гуманитарное» образование. Наилучшая точность – 65% (+/- 2% – доверительный интервал) при использовании минимум 10 комментариев от одного пользователя. Полученная точность статистически значимо выше точности случайного гадания (50%). Можно предположить, что точность в дальнейшем можно несколько улучшить с помощью сбора большего количества данных и использования более глубоких моделей обучения.

Наиболее важными для модели градиентного бустинга, не использующей непосредственно слова комментариев, оказались следующие признаки:

- средняя длина слова;
- количество слов;
- процентное содержание английских слов;
- использование восклицательного знака.

Естественно, нельзя утверждать, что именно полученное высшее образование «делит» людей на классы, используемые в данной работе. Более вероятно, что люди при поступлении в ВУЗ выбирают специальность исходя из уже имеющегося типа мышления и личностных особенностей.

Список литературы

1. Типичный Воронеж [Электронный ресурс] : страница ВКонтакте. – Режим доступа : <https://vk.com/cityvrn>
2. Open API [Электронный ресурс] : Разработчикам | ВКонтакте. – Режим доступа : <https://vk.com/dev/openapi>
3. Manning C.D. Introduction to information retrieval / C.D Manning, P. Raghavan, H. Schutze. – Cambridge : Cambridge university press, 2008. – P. 496.
4. Дьяконов А.Г. Анализ данных, обучение по прецедентам, логические игры, системы WEKA, RapidMiner и MatLab / А.Г. Дьяконов. – М. : МАКСПресс, 2010. – 278 с.

КОНФОРМНЫЕ ОТОБРАЖЕНИЯ ПЛОСКИХ ФИГУР

Д. В. Козориз

Студент

А. В. Лобода

Профессор

Введение и постановка задачи

Задачей этой работы является построение взаимно-однозначного конформного отображения Г-образного «коридора» на прямоугольник (рис. 1).

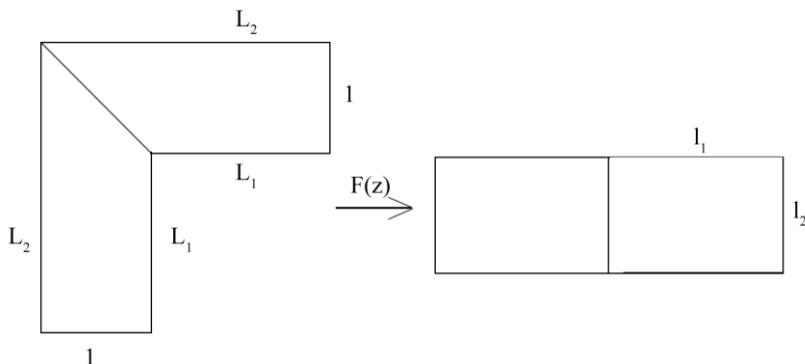


Рис. 1. Отображение Г-образного «коридора» на прямоугольник

По теореме Римана [1] любые две односвязные области на плоскости (границы которых содержат более одной точки) конформно эквивалентны. Поэтому наша задача заключается в конкретном описании (формулами и наглядным образом) соответствующего отображения.

Работа была выполнена в связи с известной задачей [2] о продвижении дивана через угол. Преобразования плоских фигур могут быть полезны также в различных задачах, связанных с распознаванием образов.

1. Принцип симметрии Римана-Шварца

Считая «коридор» симметричным, будем использовать принцип симметрии [1]. Следуя этому принципу, можно отобразить не «коридор»

на прямоугольник, а половину «коридора», т. е трапецию на половину прямоугольника (также являющуюся прямоугольником (рис.2)). Итоговое отображение получается в результате симметричного продолжения «половинного» отображения.

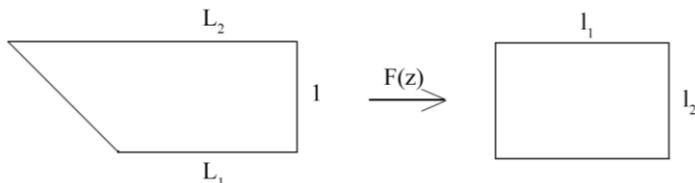


Рис. 2. Половинное отображение

Для построения такого отображения необходим промежуточный шаг – сначала трапеция отображается на полуплоскость, а затем полученная полуплоскость отображается уже на прямоугольник.

2. Формула Кристоффеля-Шварца

Для конформного отображения верхней полуплоскости на произвольный многоугольник используется формула Кристоффеля-Шварца, которая имеет общий вид:

$$F(z) = C_1 \int_0^z (t - a_1)^{\alpha_1 - 1} \cdot \dots \cdot (t - a_n)^{\alpha_n - 1} dt + C_2, \quad (1)$$

где a_1, \dots, a_n - точки действительной оси, соответствующие вершинам многоугольника, $\alpha_1\pi, \dots, \alpha_n\pi$ – внутренние углы многоугольника, C_1, C_2 – некоторые комплексные числа.

Для отображения верхней полуплоскости на прямоугольник можно использовать ту же формулу, но удобнее воспользоваться эллиптическим интегралом:

$$\zeta(z) = \int_0^z \frac{dt}{\sqrt{(1-t^2)(1-t^2k^2)}}, \quad (2)$$

где k – действительное число, удовлетворяющее условию $0 < k < 1$.

Подынтегральная функция содержит четыре особые точки $t = -1, t = -\frac{1}{k}, t = 1, t = \frac{1}{k}$, которые переходят при отображении (2) в вершины прямоугольника.

3. Компьютерные вычисления

Все вычисления этой работы выполнены в программном пакете Maple 17.

Отобразим верхнюю полуплоскость на нашу трапецию. Внутренние углы равны $\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}$, высота по условию исходной задачи равна 1. Зафиксируем длины оснований трапеции L_1 и L_2 . Пусть $L_1 = 3, L_2 = 4$ (в силу теоремы Римана эти длины можно выбирать произвольно, при различных L_1, L_2 изменяются лишь коэффициенты изучаемого отображения).

Подбором находим точки $a_1 = 0, a_2 = 0,001426, a_3 = 2,1, a_4 = 10$ – прообразы вершин трапеции на действительной оси.

Таким образом, функция, отображающая верхнюю полуплоскость на трапецию с вершинами $(0, 0), (0, 1), (4, 1), (3, 0)$ и углами $\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}$ примет вид:

$$F(z) = \int_0^z \frac{1}{t^2} \cdot (t - 0,001426)^{-\frac{1}{2}} \cdot (t - 2,1)^{-\frac{3}{4}} \cdot (t - 10)^{-\frac{1}{4}} dt. \quad (3)$$

Обратная к (3) функция F^{-1} отобразит трапецию на верхнюю полуплоскость. Граница трапеции отображается на действительную ось, где $t_1 = 0, t_2 = 0,001426, t_3 = 2,1, t_4 = 10$ – прообразы угловых точек – вершин трапеции.

Прообразы вершин прямоугольника симметричны относительно нуля, наши же, полученные после отображения трапеции на полуплоскость несимметричны. Следовательно, необходимо дробно-линейной функцией отобразить в себя верхнюю полуплоскость так, чтобы $t_1 \rightarrow -A, t_2 \rightarrow -1, t_3 \rightarrow 1, t_4 \rightarrow A$, где A – некоторое действительное число, $A = \frac{1}{k}, A > 0$.

Находим коэффициенты a, b, c, d дробно-линейного отображения.

$$L(z) = \frac{31,37110236z + 1.919776874}{29,58413500z + 1.832854705}. \quad (4)$$

Получаем $L(t_4) = A = 1,047424473$. Следовательно, $k = \frac{1}{A} = 0,9547$.

Мы отобразили трапецию с фиксированными длинами оснований на прямоугольник со сторонами $l_1 = 5,2$ и $l_2 = 1,6$.

Найдем линии внутри трапеции, на которые отобразятся горизонтальные и вертикальные линии внутри полученного (половинного) прямоугольника (рис.3).

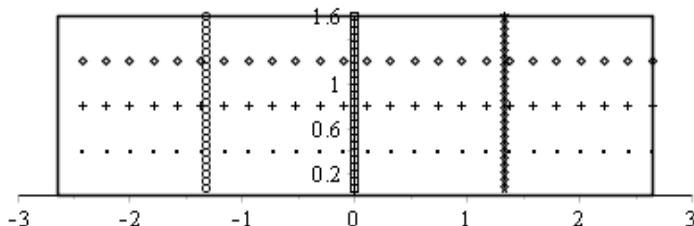


Рис. 3. Прямоугольная сетка

Для этого последовательно применим к прямоугольной сетке обратные к (2), (4) преобразования и преобразование (3). Полученную сетку внутри трапеции по принципу симметрии продолжаем для всего «коридора»:

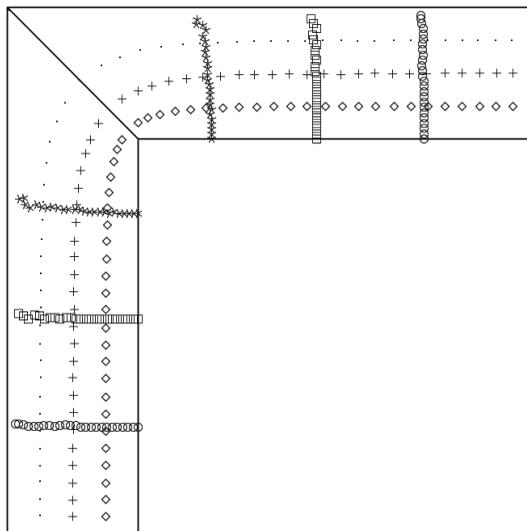


Рис. 4. Отображение прямоугольной сетки на линии внутри «коридора»

Для иллюстрации рассмотрим деформацию фигуры, расположенной в прямоугольнике, при отображении на «коридор» (рис.5).

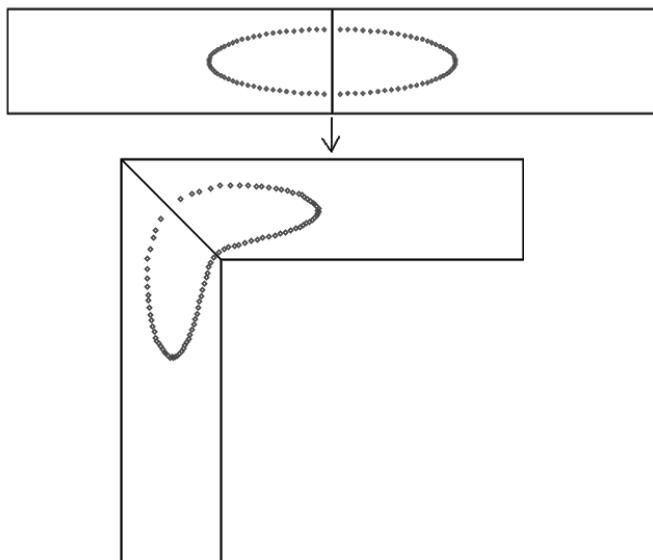


Рис. 5. Деформация фигуры

Заключение

С использованием пакета символьных вычислений Maple в работе построено и проиллюстрировано наглядным образом взаимно-однозначное отображение угловой области на прямоугольник, сохраняющее углы во всех внутренних точках. Работа представляет одно из направлений в исследовании известной нерешенной задачи о «движении дивана» и является, по сути, первым модельным шагом в этом направлении.

Вычисления и иллюстрации, связанные с построенным отображением, показывают, что возникающие деформации фигур можно достаточно жестко контролировать. Такая возможность, как ожидается, позволит получить новые оценки в «задаче о диване».

Список литературы

1. Привалов, И.И. Введение в теорию функций комплексного переменного : учебник для вузов / И.И. Привалов. – Изд. 13-е. – М.: Наука, 1984. – 432с.
2. Gerver, J. L. On Moving a Sofa Around a Corner / J. L. Gerver // Geometriae Dedicata. – 1992. – V.42. – №3. – P. 267-283.

ОПТИМИЗАЦИЯ РАСПИСАНИЯ ДВИЖЕНИЯ ГОРОДСКОГО АВТОТРАНСПОРТА. АВТОМАТИЗАЦИЯ ИЗМЕНЕНИЯ И СОСТАВЛЕНИЯ МАРШРУТОВ

А.В. Козырев

Магистрант

Н.К. Самойлов

Ассистент

Введение

Развитие и рост городов, а также населения в них порождают соответствующее увеличение количества городского автотранспорта. Составление различных маршрутов, оптимизация расписания движения маршрутных автобусов и снижение нагрузки на общую городскую транспортную сеть являются крайне актуальным научным исследованием в последние десятилетия [1].

В настоящее время не существует универсальной системы контроля за движением, временем отправления с конечных станций и прибытия на остановки. Используются простейшие таймеры, а системы контроля и управления со стороны диспетчерских служб находятся в базовом состоянии.

Интерес к решению данной проблемы продолжает расти. Это связано с большим количеством заинтересованных сторон. Частные компании заинтересованы в увеличении прибыли и системе управления собственным автопарком на маршруте. Органы власти стремятся к уменьшению нагрузки на городскую транспортную сеть. Конечный потребитель – житель города заинтересован в сокращении времени ожидания транспорта на остановках, а также в комфортном передвижении – число пассажиров равномерно распределяется по разным автобусам [2].

В данной работе рассматривается задача разработки целостной системы добавления и редактирования маршрутов городского транспорта. Управление интервалами прибытия на остановки и контроля передвижения каждого отдельного автобуса. Разработка системы контроля и управления данной системой для диспетчерских служб. Система тестирования и оценка динамики производимых изменений в транспортной сети города. Объектом исследования является городская автотранспортная сеть города Воронежа.

1. Методика решения поставленных задач

Для решения перечисленных задач разработана модульная система. Первый модуль служит для добавления и редактирования маршрутов городского транспорта. Маршруты используются для мониторинга транспорта. Общие для разных маршрутов участки движения и остановки должны быть одним объектом, это необходимо, чтобы можно было собирать статистику по каждому участку независимо от того на каком маршруте происходит набор статистических данных. Модуль включает в себя интерфейс пользователя для работы с системой. Полученные маршруты обрабатываются, анализируются на корректность [3].

Второй модуль, отвечает за равномерное распределение прибытия каждого автобуса на каждую остановку. Проводится мониторинг общих для разных маршрутов участков движения. Производится расчёт ориентировочного времени прибытия каждого из автобусов на данном участке (на основании данных о скорости, местоположении и азимуте конкретного автобуса, а также с учетом высчитываемого расстояния от текущего местоположения до ближайшей секции). В данном модуле учитывается соблюдение интервала в расписании прибытия автобусов внутри одного конкретного маршрута. Введен весовой коэффициент приоритета для разных автобусов. Модуль включает в себя интерфейс пользователя для водителей, а именно в форме android-приложения [4].

Третий модуль – это создание имитационной модели движения автотранспорта и генерации данных на основе, которой можно протестировать систему на работоспособность после каждого изменения.

2. Описание маршрута

Маршрут должен представлять из себя последовательность объектов, которые проезжает транспорт, совершая один рейс. Элементы маршрута должны храниться упорядоченно, как их проезжает транспорт. Выделено три типа объектов на маршруте: отрезок, остановка, перекресток [5].

Отрезок – максимально прямой участок дороги (ответственность за аппроксимацию относительно реальной кривизны лежит полностью на пользователе модуля) (рис. 1). Представляет собой направленный отрезок. Должен иметь точки начала и конца. Направление задается движением транспорта на данном участке дороги. Если дорога содержит более одной полосы в каждом направлении, то все они описываются одним отрезком. Отрезок не должен пересекать перекресток, если на этом перекрестке идёт разветвление секций (другие маршруты делают поворот в этом месте). Отрезок может проходить через остановку (реальную), в том случае, если маршруты, содержащие его, не содержат эту остановку.



Рис. 1. Пример отрезка

Перекресток – по сути, обычный перекресток (рис. 2). Добавлять перекресток следует, только если разные маршруты соединяются на нём, или же расходятся или и то и другое, если один единственный маршрут просто пересекает перекресток, то не обязательно его создавать, можно просто прокинуть через него отрезок. Описывается набором точек, к которым может примыкать отрезок или остановка.

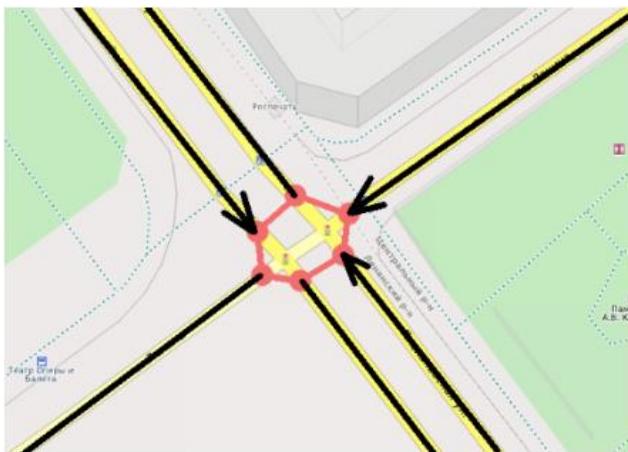


Рис. 2. Пример перекрестка

Остановка – это участок дороги, на котором останавливается маршрутный транспорт (рис. 3). Так же как и отрезок является направленным по движению отрезком, но который имеет имя.



Рис. 3. Пример остановки

Все объекты на маршруте должны быть соединены (от первого до последнего элемента можно провести линию, проходящую через все элементы маршрута “не отрывая ручки”). Реальный маршрут в прямом и обратном направлениях представляет собой два независимых маршрута.

3. Система управления движением автотранспорта

Входные данные о скорости, местоположении и азимуте конкретного автобуса берутся с установленных на каждом автобусе датчиков системы ГЛОНАСС. Это позволяет обеспечить [6]:

- Доступ из любой точки мира.
- Получение непрерывных отчетов по месту и времени стоянок и остановок, а также скорости автобусов на любом участке пути.
- Отображение местоположения и расстояния между участниками маршрута.
- Контроль над соблюдением необходимого интервала между участниками маршрута.

Меню для просмотра данных, поступающих с датчиков, установленных на автобусах, показано на рис. 4.

Данные приходят с интервалом в пределах одной минуты. На основании этих данных производится расчёт ориентировочного времени прибытия каждого из автобусов на данном участке, учитывая различные дорожные ситуации – «пробки» на дорогах, светофоры, аварии, технические неисправности самих автобусов.

Номер	Прием	Отправка	Коорд.	Отправка	Ост.	ЦДС ИД
	18:22:56	22.05.2018:22:55	22.05.20	22:38:43	19.05.20	
	06:41:35	22.05.20	06:41:27	22.05.20	06:41:27	22.05.20
	14:48:44	21.05.20	14:48:44	21.05.20	10:50:18	21.05.20
	18:22:40	22.05.2018:22:40	22.05.20	18:19:30	22.05.20	
	15:01:54	22.05.20	15:01:54	22.05.20	15:01:54	22.05.20
	18:21:50	22.05.2018:21:50	22.05.20	18:21:50	22.05.20	
	18:22:58	22.05.2018:22:58	22.05.20	18:21:08	22.05.20	
	14:48:07	22.05.20	14:48:08	22.05.20	12:35:50	22.05.20
	18:23:01	22.05.2018:22:41	22.05.20	18:20:57	22.05.20	
	18:22:41	22.05.2018:22:24	22.05.20	18:17:39	22.05.20	
	18:13:12	22.05.2018:13:12	22.05.20	18:09:11	22.05.20	
	18:22:00	22.05.2018:22:00	22.05.20	18:02:56	22.05.20	
	18:21:43	22.05.2018:21:43	22.05.20	18:17:44	22.05.20	
	18:22:27	22.05.2018:22:27	22.05.20	17:44:53	22.05.20	
	18:22:19	22.05.2018:21:42	22.05.20	18:22:19	22.05.20	
	18:22:19	22.05.2018:22:19	22.05.20	18:21:24	22.05.20	
	18:22:07	22.05.2018:22:07	22.05.20	20:06:27	21.05.20	
	18:22:55	22.05.2018:22:55	22.05.20	18:21:57	22.05.20	
	18:23:27	22.05.2018:23:27	22.05.20	18:22:22	22.05.20	
	18:23:00	22.05.2018:23:00	22.05.20	18:14:37	22.05.20	
	18:21:43	22.05.2018:21:43	22.05.20	20:44:22	22.05.20	
	18:05:16	22.05.2018:05:06	22.05.20	18:03:39	22.05.20	

Рис. 4. Web-интерфейс поступающих данных с автобусов

Расчёт происходит исходя и нескольких факторов:

- Персональный интервал движения автобусов внутри конкретного маршрута.
- Весовой коэффициент распределения инструкций для водителя (пример: если один автобус “притормаживал” четыре раза, а другой шесть, то вперед будет пропущен автобус с большим числом замедлений).
- Распределение по принципу наименьшего числа автобусов на остановке в каждый момент времени.
- Расчёт вероятности прибытия каждого автобуса к следующему объекту, исходя из дорожной обстановки.

Инструкции поступают на устройства, установленные в автобусе для водителей. Отражается общая дорожная информация, время в пути, рассчитанная скорость движения, также предупреждение обо всех изменениях. Выполнение данных инструкций контролируется посредством собираемой статистики (рис. 5). Указывается процент выполнения инструкций, «пропущенные» остановки и просчитывается безопасность стиля вождения (резкие, либо плавные старты и торможения, а также превышение скорости, либо отсутствие такого).

Информация для водителей поступает на android-приложения. Она подразделяется на несколько категорий [7]:

- Рассчитанная рекомендованная скорость
- Информация по остановкам

– Информация об интервалах времени прибытия и “простоя” на остановках

маршрут	количество остано	посещенные остановки	найденные пропущенные остановки
	31	15208	5132
	30	16525	5695
	48	12144	3019
	64	5292	1638
	85	1147	484
	51	17613	4561
	53	2910	644

Рис. 5. Пример Web-интерфейса системы контроля

Интерфейс android-приложения для водителей показан на рис. 6.

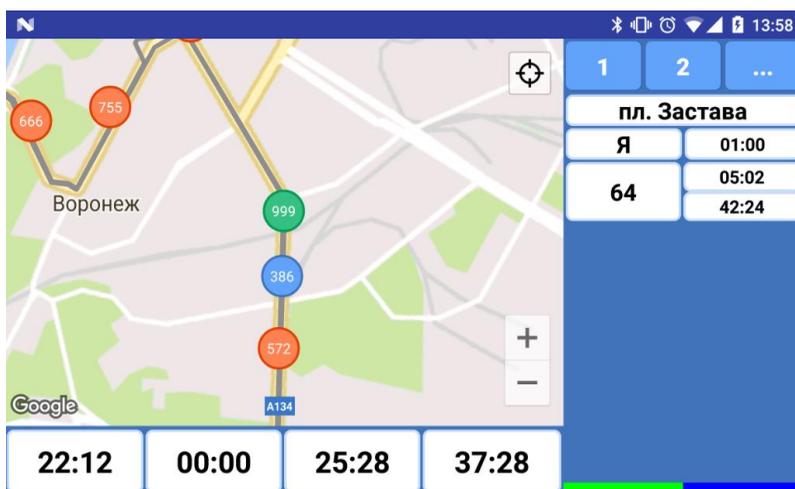


Рис. 6. UI для водителей

Заключение

В рамках данной статьи была рассмотрена реализация системы оптимизации расписания движения городского автотранспорта. Тема крайне актуальна в наше время и чрезвычайно востребована. Систему можно качественно развивать в множестве направлений. Разработка и доработка всех систем продолжается, также планируется добавление ряда дополнительных модулей. В качестве обработки всех системных событий выступает сервер, написанный на языке Java. Для взаимодействия системы с конечным пользователем используются смартфоны на базе операционной системы Android с централизованным управлением и работой в фоновом режиме.

Рассмотренный подход является достаточно универсальным и может применяться в условиях различных дорожных ситуаций и любых городов.

Список литературы

1. Глонассова А. Мониторинг транспорта: СНГ vs Запад / А. Глонассова [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/post/241109/>
2. Глонассова А. Мониторинг транспорта: СНГ vs Запад. Часть 2 / А. Глонассова [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/post/241367/>
3. Eckel B. Thinking in Java / В. Eckel. – New Jersey : Prentice Hall, 2006. – 1057 p.
4. Introduction to Android [Электронный ресурс] : Android Developers. – Режим доступа : <https://developer.android.com/guide/index.html>
5. GPS мониторинг – контроль “маршрутного” транспорта [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/company/m2m/blog/119071/>
6. Vistar. GPS/ГЛОНАСС мониторинг / (с) Copyright Vistar 2014 [Электронный ресурс]. – Режим доступа : <http://monitoring.vistar.su/>
7. Зарубежные сервисы GPS мониторинга [Электронный ресурс]. – Режим доступа : <https://habrahabr.ru/company/euler2012/blog/321750/>

ПРОЕКТИРОВАНИЕ И СБОРКА 3D-ПРИНТЕРА, АВТОМАТИЗАЦИИ ЭТАПОВ 3D-ПЕЧАТИ

А.С. Колесов

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Сравнительно недавно возникнув, технологии 3D-печати завоевывают все большую популярность в различных сферах деятельности. В настоящий момент 3D-печать уже доступна практически каждому – бытовой 3D-принтер можно купить в обычном компьютерном магазине за разумную цену. Однако помимо 3D-принтера необходимо наличие соответствующего программного обеспечения и, главное, умения с ним работать.

С целью подробного изучения технологий 3D-печати в данной работе предпринята успешная попытка разработки с нуля собственного 3D-принтера, а также подбор и доработка необходимого программного обеспечения для печати на собранном принтере.

1. Этапы 3D-печати

3D-печать – это процесс послойного создания физического объекта по цифровой 3D-модели с помощью 3D-принтера. В общем случае для печати на 3D-принтере готового изделия необходимо выполнить следующие шаги:

- разработка 3D-модели будущего изделия в 3D-редакторе;
- преобразование созданной 3D-модели в G-код для конкретного 3D-принтера;
- собственно печать.

G-code – это язык программирования устройств с числовым программным управлением (ЧПУ), представляет собой набор команд, позволяющих управлять различными узлами принтера. G-code чаще всего представлен в текстовом виде.

Программы управления процессом печати подключаются к 3D-принтеру, чаще всего по последовательному интерфейсу, и по одной команде передают сгенерированную на предыдущем шаге программу. Почти все подобные программы обладают функционалом визуализации процесса печати в реальном времени.

2. Конструкция 3D-принтера

В настоящий момент существует множество различных видов 3D-принтеров. В основном они различаются по технологии печати и виду кинематики – принципа механической конструкции 3D-принтера для позиционирования печатающего элемента принтера (или печатаемой детали) в процессе печати.

3D-печать может осуществляться различными способами и с использованием различных материалов, но в основе любого из способов лежит принцип послойного «выращивания» твёрдого объекта.

Наиболее популярные технологии создания слоев [1]:

- экструзия – слой формируется путем выдавливания какого-либо материала через сопло-дозатор;
- фотополимеризация – слой формируется путем засвечивания жидкого фотополимера в ультрафиолетовом диапазоне;
- послойное спекание – слой формируется путем плавления порошка под воздействием лазерного облучения.

Наиболее просты в реализации принтеры, использующие экструзию и фотополимеризацию. Однако учитывая, что пластик намного дешевле фотополимера, в процессе проектирования собственного 3D-принтера был выбран первый вариант.

Такой принтер должен уметь позиционировать сопло по трем осям. Основной вопрос, который стоит перед конструктором на данном этапе,

заключается в том, какой тип конструкции выбрать, чтобы добиться наилучшего соотношения сложности реализации и качества печати. Для построения 3D-принтеров было предложено большое количество самых разнообразных конструкций, наиболее удачными из которых являются:

- XZ Head Y Bed – одна из самых популярных механик, отличается большим количеством крепежных элементов, которые сложно держать настроенными;

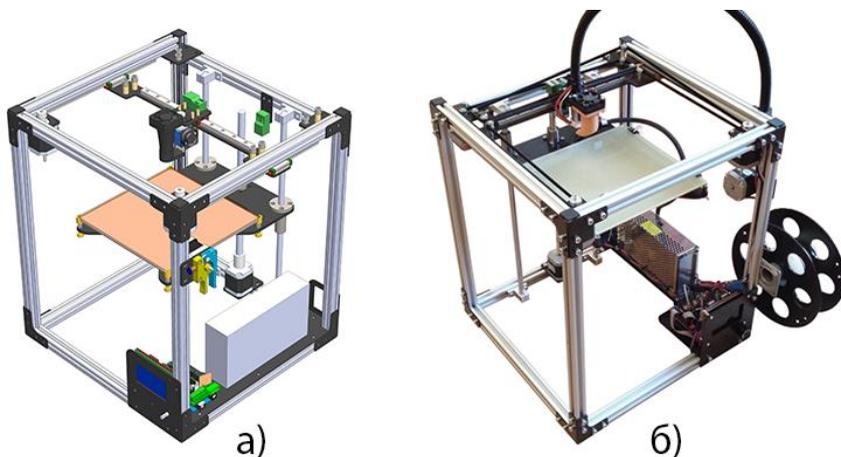
- XY Head Z Bed – используется, когда перемещать заготовку легче, чем перемещать инструмент;

- Delta – экструдер перемещается по всем трем осям, обеспечивается высокая скорость печати, но высока вероятность накопления ошибки позиционирования;

- H-Bot – все двигатели закреплены, высокая скорость печати, но возможны искажения на краях;

- CoreXY – все двигатели закреплены, высокая скорость печати, наиболее совершенная механика [2].

В силу простоты реализации и отсутствия видимых недостатков была выбрана кинематика CoreXY. На рис. 1 приведен проект разработанного 3D-принтера данной конструкции и уже собранное изделие.



а – проект, б – готовое изделие

Рис. 1. Разработанный 3D-принтер

3. Программное обеспечение для 3D-печати

Как было уже отмечено процесс подготовки к печати и собственно печати состоит из нескольких шагов. Для выполнения каждого шага

существуют различные программы, в том числе бесплатные и открытые. Однако поэтапная работа в нескольких программах, как минимум, неудобна, поэтому в рамках данного проекта 3D-принтера была предпринята попытка собрать интегрированное решение для выполнения всех этапов подготовки и 3D-печати из одной программы.

Полученное решение должно обладать следующей функциональностью:

- создание и редактирование 3D-моделей;
- преобразование модели в G-code;
- управление принтером и процессом печати;
- предпросмотр и визуализация процесса печати;
- автоматизация цикла 3D-печати до уровня «печать одной кнопкой».

Полностью самостоятельная разработка ПО с требуемой функциональностью является неподъемной задачей для одного и даже нескольких человек, поэтому итоговое решение должно быть построено на базе существующего свободного ПО.

Очевидно, что основой интегрированного программного решения для 3D-печати должен быть 3D-редактор. Из наиболее распространенных, а соответственно, знакомых пользователям редакторов, таких как 3ds Max Design, Cinema 4D, Maya и Blender, был выбран последний как единственный, являющийся свободным программным обеспечением, а также предоставляющий API для создания плагинов и пользовательских элементов графического интерфейса [3].

Для 3D-печати непосредственно из Blender для последнего было реализовано соответствующее расширение с использованием предоставляемого API.

В качестве генератора G-code было решено использовать Slic3r потому, что он также является свободным ПО и, кроме того, имеет интерфейс командной строки [4]. Интеграция со Slic3r выполнена следующим образом: подготовленная модель экспортируется в файл, который обрабатывается отдельным процессом Slic3r, запущенным с необходимыми настройками преобразования, переданными в командной строке; сгенерированный G-код затем загружается из полученного файла в память Blender.

Визуализация процесса печати была реализована встроенными средствами Blender. С помощью Blender Python API на сцену добавлялся объект типа «кривая», которая размещается рядом с печатаемым объектом и отображает текущее состояние печати, что показано на рис. 2.

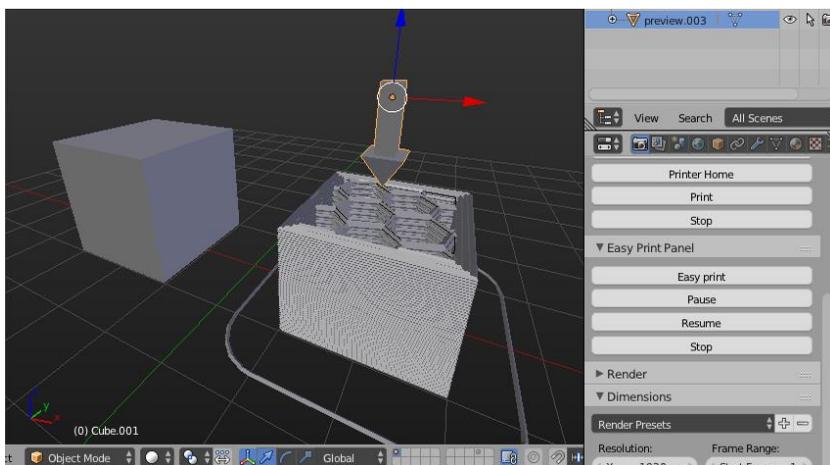


Рис. 2. Визуализация процесса печати и панель управления печатью

Управление принтером осуществляется через последовательный порт. Для поддержки кроссплатформенности использовалась библиотека PySerial. В процессе печати команды управления поочередно отправляются принтеру, при этом синхронно обновляется отображение процесса печати на сцене Blender.

К пользовательскому интерфейсу Blender была добавлена вкладка с панелями, каждая из которых отвечает за соответствующий компонент системы. На этой же вкладке находится панель, реализующая интерфейс «печать одной кнопкой» (рис. 2). В этом случае необходимые компоненты для 3D-печати будут сконфигурированы автоматически с параметрами по умолчанию.

Заключение

В результате проделанной работы был спроектирован и собран собственный 3D-принтер, построенный на базе CoreXY-кинематики.

Также было разработано полностью свободное интегрированное программное решение для 3D-печати, выполненное в виде расширения 3D-редактора Blender. Полученное решение может быть сконфигурировано для использования с другими 3D-принтерами различных конструкций.

Список литературы

1. Слюсар В.И. Фаббер-технологии. Новое средство трехмерного моделирования / В.И. Слюсар // Электроника: наука, технология, бизнес. – 2003. – № 5. – С. 54-60.

2. CoreXY | Cartesian Motion Platform [Электронный ресурс]. – Режим доступа : <http://corexy.com/theory.html>
3. Blender Documentation Contents – Blender 2.78.0 – API documentation [Электронный ресурс] : Документация. – Режим доступа : <https://docs.blender.org/api/2.78b/>
4. Slic3r Manual – Command Line Usage [Электронный ресурс]. – Режим доступа : <http://manual.slic3r.org/advanced/command-line>

ТЕОРЕТИКО-МНОЖЕСТВЕННЫЙ ПОДХОД К РАСПРЕДЕЛЕНИЮ ШТРАФОВ И ВОЗНАГРАЖДЕНИЙ В ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ

Е.А. Копытина

Студент

А.В. Копытин

Доцент

Д.И. Соломатин

Старший преподаватель

Введение

Одной из наиболее важных проблем управления проектами является проблема задержки проекта, которая неминуемо влечет за собой потерю значительной доли денежных средств, как со стороны исполнителя, так и со стороны заказчика. Во многих странах, при задержке общественного проекта, власти по закону имеют право оштрафовать осуществляющую его компанию. В контракте имеется пункт, по которому предусматривается штраф в том случае, если проект завершается позже запланированного времени. Важно знать за какую часть штрафа исполнители несут ответственность.

В тоже время имеет место дополнительное вознаграждение за досрочное выполнение проекта. В качестве примера можно привести разработку новых высокотехнологичных устройств. Компании борются за то, чтобы их новая продукция оказалась на рынке раньше продукции конкурентов, и, поэтому обещают бонус своим сотрудникам, если изготовление устройства завершится раньше запланированного времени. Следующий вопрос состоит в том, как разделить этот бонус между сотрудниками, чьими усилиями проект завершится раньше намеченного срока.

Впервые теоретико-игровой подход к анализу обозначенных проблем был предложен в [1]. Этот подход состоит в построении кооперативной игры, связанной с проектной проблемой и рассмотрении ее ядра как соответствующего множества решений. В [2] результаты [1] обобщаются путем рассмотрения произвольной неубывающей функции вознаграждения

1. Цель работы

В настоящее время в проектной деятельности существует острая необходимость в распределении штрафов и вознаграждений. Однако, недостаточно исследованы и внедрены системы распределения штрафов между участниками проекта, работа которых приводит к его срыву, и распределения вознаграждений для сотрудников, благодаря усилиям которых проект завершается раньше запланированного срока.

Из всего вышеуказанного следует актуальность исследования, которая подразумевает разработку универсального инструмента, позволяющего находить оптимальное распределение штрафов и вознаграждений в проекте.

Целью исследования является разработка веб-приложения, позволяющего:

- Находить оптимальное распределение штрафов и вознаграждений в проекте, используя теоретико-игровой подход;
- Редактировать граф проекта (добавлять или удалять работы и связи между ними).

2. Понятие проекта

Проект – это множество работ с известными взаимосвязями. Эти работы выполняются в течение определенного промежутка времени и направлены на достижение конкретной цели. Пусть N обозначает множество всех работ проекта. Для произвольной работы $i \in N$ пусть P_i обозначает множество всех предшествующих работ, т.е. тех работ, которые должны быть завершены до начала выполнения работы i , а F_i – множество всех следующих за i работ, т.е. тех, для начала выполнения которых, необходимо завершение работы i . Проект определяется как набор упорядоченных подмножеств N или путей, $\{N_1, \dots, N_m\}$, где биекция $\sigma_\tau : \{1, \dots, |N_\tau|\} \rightarrow N_\tau$ задает порядок в N_τ , $\tau \in \{1, \dots, m\}$, удовлетворяющий следующим условиям:

- $N = \bigcup_{\tau=1}^m N_\tau$;
- $F_{\sigma_\tau(i|N_\tau)} = \emptyset$, $P_{\sigma_\tau(1)} = \emptyset$, и $P_{\sigma_\tau(r)} = \{\sigma_\tau(1), \dots, \sigma_\tau(r-1)\}$ для каждого $\tau \in \{1, \dots, m\}$ и каждого $r \in \{2, \dots, |N_\tau|\}$;

– для $\tau, \nu \in \{1, \dots, m\}$, если $i, j \in N_\tau \cap N_\nu$ и $\sigma_\tau^{-1}(i) < \sigma_\tau^{-1}(j)$, то $\sigma_\nu^{-1}(i) < \sigma_\nu^{-1}(j)$.

Заметим, что проект может быть представлен в виде направленного графа, в котором множество дуг соответствует множеству работ. Для того чтобы избежать множественных дуг, в графе вводятся фиктивные работы, не расходующие ни времени, ни ресурсов. Фиктивные работы изображаются штрихованными дугами.

Свяжем с проектом $\{N_1, \dots, N_m\}$ функцию продолжительности. Свяжем с проектом $\{N_1, \dots, N_m\}$ функцию продолжительности $l: N \rightarrow \mathbb{R}_+$, где $l(i)$ обозначает длительность или продолжительность работы $i \in N$. Определим продолжительность $D(N_\tau, l)$ пути N_τ в соответствии с функцией l как сумму продолжительностей его работ, т.е. $D(N_\tau, l) = \sum_{i \in N_\tau} l(i)$. Продолжительность проекта $D(l)$ в соответствии с функцией l есть максимальная из продолжительностей его путей, т.е. $D(l) = \max_{1 \leq \tau \leq m} D(N_\tau, l)$. Путь N_τ будем называть критическим, если его продолжительность равна продолжительности проекта, т.е. $D(N_\tau, l) = D(l)$.

Введем в рассмотрение функцию $p: N \rightarrow \mathbb{R}_+$, представляющую запланированное или оцениваемое время выполнения работ и функцию $r: N \rightarrow \mathbb{R}_+$, представляющую реальное время выполнения работ после завершения проекта. Определим функцию задержки $d: N \rightarrow \mathbb{R}_+$ как $d(i) = (r(i) - p(i))_+ (= \max\{r(i) - p(i), 0\})$, т.е. $d(i)$ есть задержка работы i . Аналогично, определим функцию досрочного завершения $e: N \rightarrow \mathbb{R}_+$ как $e(i) = (p(i) - r(i))_+$, т.е. $e(i)$ есть время, на которое работа i завершилась раньше запланированного срока. Пусть $\mathbf{D} = \{i \in N \mid d(i) > 0\}$ и $\mathbf{E} = \{i \in N \mid e(i) > 0\}$ обозначают множества просроченных и выполненных досрочно работ соответственно.

3. Кооперативная игра расходов

Кооперативной игрой в форме характеристической функции называется пара (N, ν) , где N – конечное множество игроков, а $\nu: 2^N \rightarrow \mathbb{R}$ – характеристическая функция, сопоставляющая каждой коалиции, т.е. подмножеству $S \subseteq N$, действительное число $\nu(S)$, причем $\nu(\emptyset) = 0$. В общем случае, $\nu(S)$ представляет собой совместную выплату, которую может получить коалиция S , если ее члены решат объединиться.

Кооперативная игра может отражать расходы или вознаграждения. Игра, отражающая расходы, обозначается отображением c , тогда как игра, отражающая вознаграждения, обозначается отображением v .

Ядро игры (N, v) определяется как

$$\text{Core}(v) = \left\{ x \in \mathbb{R}^N \mid \sum_{i \in N} x_i = v(N), \sum_{i \in S} x_i \geq v(S) \quad \forall S \in 2^N \setminus \emptyset \right\},$$

т. е. ядро является множеством распределений $v(N)$, против которых не может обоснованно возражать ни одна из коалиций. Важным подклассом игр с непустым ядром является класс выпуклых игр (см. [3]). Игра (N, v) называется выпуклой, если $v(S) + v(T) \leq v(S \cup T) + v(S \cap T)$ для любых $S, T \subseteq N$.

Ядро игры (N, c) определяется как

$$\text{Core}(c) = \left\{ x \in \mathbb{R}^N \mid \sum_{i \in N} x_i = c(N), \sum_{i \in S} x_i \leq c(S) \quad \forall S \in 2^N \setminus \emptyset \right\},$$

и эквивалентом выпуклой игры для игры расходов является вогнутая игра. Игра (N, c) называется вогнутой, если $c(S) + c(T) \geq c(S \cup T) + c(S \cap T)$ для любых $S, T \subseteq N$.

Проблема банкротства (bankruptcy problem) определяется тройкой (N, E, c) , где $N = \{1, \dots, n\}$ – множество игроков, $E \geq 0$ – имущество, которое должно быть разделено между игроками, а $c \in \mathbb{R}_+^N$ – вектор требований игроков, причем $\sum_{i \in N} c_i \geq E$. Соответствующая кооперативная игра $(N, v_{E,c})$ определяется как

$$v_{E,c}(S) = \left(E - \sum_{i \in N \setminus S} c_i \right)_+$$

для всех $S \subseteq N$. Как показано в [4], игра $(N, v_{E,c})$ является выпуклой.

Проблема налогообложения (taxation problem) может рассматриваться как двойственная к проблеме банкротства. Проблема налогообложения определяется тройкой (N, E, c) , где $N = \{1, \dots, n\}$ – множество игроков, $E \geq 0$ – налог, который должен быть собран с игроков, а $c \in \mathbb{R}_+^N$ – вектор платежеспособностей игроков, причем $\sum_{i \in N} c_i \geq E$. Соответствующая кооперативная игра расходов $(N, c_{E,c})$ определяется как

$$c_{E,c}(S) = \min \left\{ E, \sum_{i \in S} c_i \right\}$$

для всех $S \subseteq N$. Как показано в [5], игра $(N, c_{E,c})$ является вогнутой.

4. Проектная проблема

Проектная проблема возникает тогда, когда реальное время выполнения работ отличается от запланированного, что приводит к задержке или ускорению проекта и связанным с этим штрафами или вознаграждениями. Будем считать, что вознаграждение (штраф) равно времени, на которое проект был выполнен досрочно (просрочен). Таким образом, проектная проблема может быть описана тройкой $(\{N_1, \dots, N_m\}, p, r)$.

Пусть дана проектная проблема $(\{N_1, \dots, N_m\}, p, r)$. Предположим, что все работы, выполненные досрочно, на самом деле завершились по плану, т.е. функция $\max\{p, r\}$ задает реальное время выполнения работ после завершения проекта. Тогда разность $D(\max\{p, r\}) - D(p)$ определяет максимальную сумму, на которую могут быть оштрафованы все просроченные работы. Простейший способ задания вектора $b \in \mathbb{R}_+^N$ максимальных индивидуальных ответственностей работ проекта состоит в том, чтобы положить $b_i = d(i)$ для каждой работы $i \in N$. Однако, такой подход неадекватен, поскольку любая работа $i \in D$ не может быть оштрафована больше, чем на неотрицательную часть разности между максимальной продолжительностью пути, содержащем i , в соответствии с функцией $\max\{p, r\}$ и плановой продолжительностью проекта. Поэтому определим вектор b следующим образом:

$$b_i = \max_{\tau: N_i \ni i} \min \{ d(i), (D(N_\tau, \max\{p, r\}) - D(p))_+ \} \quad \text{для всех } i \in N.$$

Предположим теперь, что все просроченные работы на самом деле завершились по плану, т.е. функция $\min\{p, r\}$ задает реальное время выполнения работ после завершения проекта. Тогда разность $D(p) - D(\min\{p, r\})$ определяет максимальное возможное вознаграждение всех работ, выполненных досрочно. Простейший способ задания вектора $f \in \mathbb{R}_+^N$ максимальных индивидуальных вознаграждений работ проекта состоит в том, чтобы положить $f_i = e(i)$ для каждой работы $i \in N$. Однако, такой подход снова неадекватен, так как любая работа $i \in E$ не может получить больше, чем неотрицательная часть разности между максимальной плановой продолжительностью пути, содержащем i , и

продолжительностью проекта в соответствии с функцией $\min\{p, r\}$. Поэтому определим вектор f следующим образом:

$$f_i = \max_{\tau, N, \varepsilon_i} \min\{e(i), (D(N_\tau, p) - D(\min\{p, r\}))_+\} \text{ для всех } i \in N \text{ [6].}$$

Рассмотрим сначала случай, когда $D(p) \leq D(r)$. В этом случае просроченные работы должны быть оштрафованы как минимум на величину $D(r) - D(p)$. Вспомним, что $D(\max\{p, r\}) - D(p) -$ максимальная величина штрафа. Разность между этими величинами $D(\max\{p, r\}) - D(r)$ является вкладом работ, завершившихся досрочно, в уменьшение общей задержки проекта. С другой стороны, $D(p) - D(\min\{p, r\})$ есть максимальное возможное вознаграждение для работ, завершившихся досрочно. Как показано в [6], в лучшем случае работы, завершившиеся досрочно, могут получить вознаграждение равное $E := \min\{D(p) - D(\min\{p, r\}), D(\max\{p, r\}) - D(r)\}$. Генеральный менеджер проекта вправе решить, какую часть этого вознаграждения должны дополнительно компенсировать просроченные работы. Пусть $\alpha \in [0, 1]$ обозначает эту часть. Тогда величина $C(\alpha) := \alpha E + D(r) - D(p)$ составляет итоговый штраф просроченных работ. Вспоминая (1), имеем

$$\sum_{i \in N} b_i \geq D(\max\{p, r\}) - D(p) \geq C(\alpha).$$

Для того чтобы распределить штраф $C(\alpha)$ между просроченными работами, рассмотрим проблему налогообложения $(N, C(\alpha), b)$. Соответствующая кооперативная игра $(N, c_{C(\alpha), b})$ расходов определяется как

$$c_{C(\alpha), b}(S) = \min\left\{C(\alpha), \sum_{i \in S} b_i\right\} \text{ для всех } S \subseteq N.$$

Для того чтобы разделить вознаграждение E между работами, завершившимися досрочно, рассмотрим проблему банкротства (N, E, f) .

Соответствующая кооперативная игра $(N, v_{E, f})$ определяется как

$$v_{E, f}(S) = \left(E - \sum_{i \in N \setminus S} f_i\right)_+ \text{ для всех } S \subseteq N.$$

Определим, наконец, соответствующую проектную игру (N, u_α^+) как

$$u_\alpha^+(S) = v_{G(\alpha), f}(S) - \alpha c_{H, b}(S) \text{ для всех } S \subseteq N.$$

Заметим, что $u_\alpha^+(N) = D(p) - D(r)$.

5. Практическая реализация

Разработано веб-приложения для расчета оптимальных штрафов и вознаграждений в проекте.

В качестве средства реализации использован прототипно-ориентированный язык JavaScript, обладающий рядом следующих преимуществ:

- Возможность работы с действиями пользователей (клики, перемещения курсора);
- Быстродействие;
- Возможность запуска программ, написанных на JavaScript, на сервере и в браузере;
- Взаимодействие с приложением может осуществляться через текстовые редакторы – Microsoft Office и Open Office.

При нажатии на кнопку «Нормализация» (рис. 1) граф проекта приобретает вид, показанный на рис. 2, таким образом работы проекта выстраиваются в соответствии с реальным временем их начала.

Расчет проекта

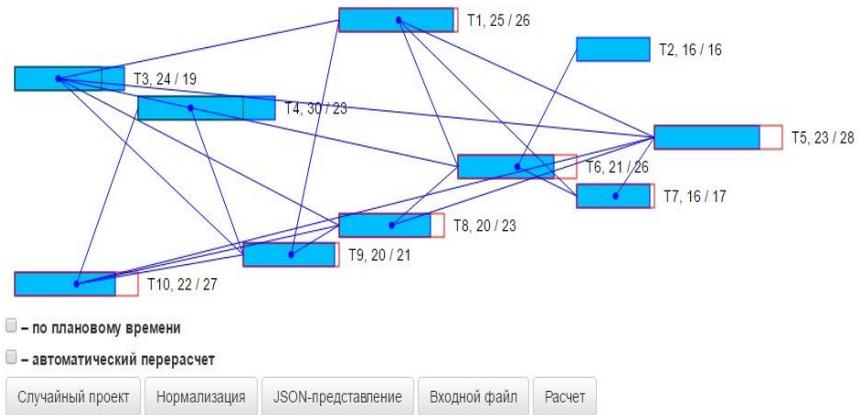


Рис. 1. Граф проекта

Расчет проекта

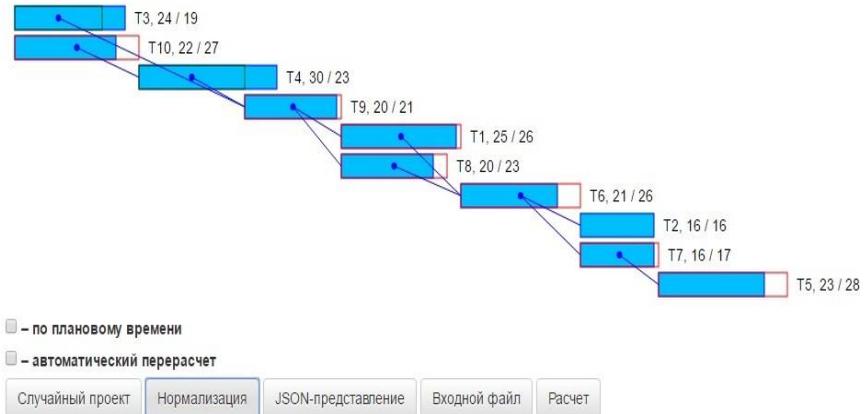


Рис.2. Граф проекта в нормализованной форме

При установке флажка «По плановому времени» работы проекта выстраиваются в соответствии с плановым временем их начала.

При установке флажка «Автоматический перерасчет» происходит расчет штрафов и вознаграждений работ проекта. Зеленым цветом показаны вознаграждения, а красным, соответственно, штрафы (рис. 3).

Расчет проекта

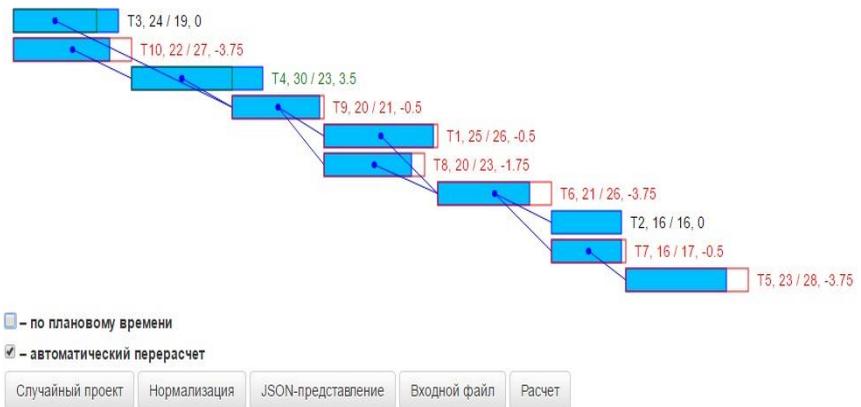


Рис.3. Расчет штрафов и вознаграждений

Заключение

Предложен теоретико-игровой подход к задаче распределения штрафов и вознаграждений в проектах. В рамках этого подхода указанная проблема сведена к известным проблемам банкротства и налогообложения, что позволяет построить семейство соответствующих выпуклых кооперативных игр. Последнее обстоятельство гарантирует наличие справедливых распределений штрафов и вознаграждений между работами проекта. Разработано веб-приложение для расчета оптимальных штрафов и вознаграждений, которое позволяет редактировать граф проекта.

Список литературы

1. Estévez-Fernández A. Project games / A. Estévez-Fernández, P. Borm, H. Hamers // *International Journal of Game Theory*. – 2007. – Vol. 36. – No. 2. – P. 149-176.
2. Estévez-Fernández A. A game theoretical approach to sharing penalties and rewards in projects / A. Estévez-Fernández // *European Journal of Operational Research*. – 2012. – Vol. 216. – No. 3. – P. 647-657.
3. Shapley L. S. Cores of convex games / L. S. Shapley // *International Journal of Game Theory*. – 1971. – Vol. 1. – No. 1. – P. 11-26.
4. Curiel I. J. Bankruptcy games / I. J. Curiel, M. Maschler, S. Tijs // *Zeitschrift für Operations Research*. – 1987. – Vol. 31. – No. 5. – P. 143-159.
5. Branzêi R. Two approaches to the problem of sharing delay costs in joint projects / R. Branzêi, G. Ferrari, V. Fragnelli, S. Tijs // *Annals of Operations Research*. – 2002. – Vol. 109. – No. 1. – P. 359–374.
6. Копытин А. В. О распределении вознаграждений в проектах / А. В. Копытин, Д. И. Соломатин, Е. А. Копытина // *Вестник Воронежского государственного университета. Сер. Системный анализ и информационные технологии*. – Воронеж, 2016. – № 1. – С. 72-76.

МОДЕЛИРОВАНИЕ ЭМОЦИЙ ВИРТУАЛЬНЫХ ПЕРСОНАЖЕЙ

В.В. Коротков

Студент

А.А. Вахтин

Доцент

Введение

Исследования в области искусственного интеллекта являются одним из доминирующих направлений развития современных информационных технологий. Как правило, применяемые на практике алгоритмы и системы, в том или ином смысле обладающие свойством интеллектуальности, сконцентрированы на эффективном решении прикладных задач. Однако в ряде областей правдоподобное воссоздание интеллектуального человеческого поведения гораздо важнее эффективности и рациональности. К таким системам можно отнести видеоигры, виртуальные тренажёры, интерактивные учебные курсы. Интерес к подобным виртуальным мирам особенно возрос в связи с развитием технологий виртуальной и дополненной реальности. Другими примерами могут служить различные машинные интерфейсы, копирующие отдельные человеческие способы коммуникации для общения с пользователем, например, интерфейсы систем «умный дом», виртуальные рассказчики и экскурсоводы.

В отличие от рациональной машины человек обладает эмоциями и чувствами, играющими существенную роль в коммуникации, принятии решений и многих других аспектах поведения. Правдоподобное воссоздание этих явлений позволит значительно повысить степень «погружения» пользователя в виртуальную среду. Именно моделированию эмоций у искусственных персонажей и посвящена данная статья.

1. Цель работы

В данной статье рассматривается создание модели эмоций и чувств виртуальных человекоподобных персонажей, а также реализующей её программной библиотеки. Условимся называть субъектов, обладающих эмоциями эмоциональными агентами.

К разрабатываемой модели и её компонентам предъявлялись следующие требования:

- индивидуальность поведения различных агентов;
- адаптивность поведения агентов к происходящим изменениям;

- интегрируемость модели в существующие системы и независимость от контекста;
- эффективность с точки зрения используемых ресурсов.

2. Формализация эмоций

Наиболее подходящими с точки зрения простоты формализации теориями возникновения эмоций являются так называемые оценочные теории. Рассматриваемая модель опирается на модель OCC (Ortony, Clore & Collins), которая рассматривает эмоции как результат оценки событий, действий и объектов мира с точки зрения целей, стандартов и знаний агента [1].

Для удобства всякое состояние мира представлено в простой форме: в виде конъюнкции литералов (предикатов), которые истинны в данный момент времени. Всякий литерал, не принадлежащий состоянию, считается ложным. Изменение состояния в результате наступления какого-либо события задаётся списком литералов, которые следует добавить к состоянию и списком литералов, которые следует исключить.

Модель эмоционального агента включает следующие компоненты:

- представление о текущем состоянии мира;
- целевые состояния и степень их важности (значения от 0 до 1);
- знания о возможных событиях, их необходимых предусловиях и возможных последствиях;
 - вероятности наступления событий (при удовлетворении их предусловий) и их последствий (значения от 0 до 1);
 - стандарты, определяющие, насколько то или иное действие порицаемо агентом (значения от -1 до 1);
 - отношение к другим агентам (значения от -1 до 1);
 - индивидуальные черты личности (вектор значений от -1 до 1).

На основе модели агента и информации о произошедшем событии определяются оценочные переменные. Среди них:

- полезность или желательность (Desirability) события;
- априорная вероятность появления события (Likelihood);
- степень уверенности в происшествии события (Certainty), сообщается с информацией о событии;
- отношение к связанному с событием агенту (Liking);
- полезность события для другого агента (DesireOther), узнаётся с помощью коммуникации;
- отношение к событию в соответствии со стандартами агента (Praiseworthiness).

В таблице указаны зависимости между эмоциями и оценочными переменными. Интенсивности эмоций в зависимости от причины и

условия их возникновения (ожидание события, подтверждение события, восприятие объекта и т.д.) вычисляется как произведение соответствующих переменных.

Таблица

Зависимость эмоций от оценочных переменных

Эмоция	Переменные
Радость – Горе	Desirability
Удивление – Шок	Likelihood
Радость за другого – Обида	Liking, DesireOther>0
Жалость – Злорадство	Liking, DesireOther<0
Надежда – Страх	Desirability, Likelihood, Certainty
Удовлетворение – Подтверждение страхов Облегчение – Разочарование	Подтверждение или отмена ожидаемого события, интенсивность надежды или страха ожидаемого события
Гордость – Стыд	Praiseworthiness
Восхищение – Порицание	Praiseworthiness, Liking

Полезность произошедших изменений состояния мира относительно целей агента определяется как разность вероятностей их достижения после и до изменений с учётом важности. Вероятность достижения цели определяется как вероятность выполнения плана – приводящей к ней последовательности действий:

$$Desirability(S \rightarrow S') = \sum_{G \in Goals} [P(Plan(S' \rightarrow G)) - P(Plan(S \rightarrow G))] Importance(G),$$

где S – исходное состояние, S' – состояние после изменений.

Так как модель предполагает возможность различных последствий событий, план состоит из пар действие-последствие:

$$P(Plan) = \sum_{(A,E) \in Plan} P(A) \cdot P(E | A), \quad Plan = [(A_1, E_1), \dots, (A_n, E_n)].$$

Полученный в результате оценки события вектор эмоций добавляется во множество активных эмоций. Текущее эмоциональное состояние агента в каждый момент времени определяется суммой активных эмоций. При этом интенсивность эмоций уменьшается линейно с течением времени до полного исчезновения и удаления из множества.

3. Индивидуальность

Личность каждого агента определяется в соответствии с теорией «Большой пятёрки» [2]. На основе четырёх из пяти измерений, предусмотренных данной теорией (экстраверсия, доброжелательность,

нейротизм, открытость опыту), вычисляются шесть коэффициентов, определяющих склонности агента:

- склонность к положительным эмоциям;

$$C_{Pos. Em} = 1 + \text{Экстраверсия} / 4$$

- склонность к отрицательным эмоциям;

$$C_{Neg. Em} = 1 + \text{Нейротизм} / 4$$

- склонность к дружелюбию;

$$C_{Pos. Feel} = 1 + (\text{Экстраверсия} + \text{Доброжелательность}) / 8$$

- склонность к антипатии;

$$C_{Neg. Feel} = 1 + (\text{Нейротизм} - \text{Доброжелательность}) / 8$$

- склонность к эмпатии;

$$C_{Oth. Em} = 1 + \text{Доброжелательность} / 4$$

- коэффициент обучения.

$$C_{Leam. Rate} = (1 + \text{Открытость_опыту}) / 10$$

Коэффициенты эмоциональных наклонностей выступают дополнительными множителями при вычислении соответствующих эмоций. Коэффициент обучения определяет вклад новых знаний в процессе обучения.

4. Обучение

При восприятии события *Event* и переходе из состояния *S* в состояние *S'* вероятность появления события обновляется в соответствии с выражением:

$$P(Event) = \begin{cases} P(Event) \cdot (1 - C_{Leam. Rate}) + C_{Leam. Rate}, & S' \neq S \\ P(Event) \cdot (1 - C_{Leam. Rate}), & S' = S \end{cases}$$

Если переход осуществился при наступлении последствия *CurEffect*, то для всех возможных последствий $Ef \in Effects(Event)$ их вероятности обновляются согласно формуле:

$$P(Ef | Event) = \begin{cases} P(Ef | Event) \cdot (1 - C_{Leam. Rate}) + C_{Leam. Rate}, & Effect = CurEffect \\ P(Ef | Event) \cdot (1 - C_{Leam. Rate}), & Effect \neq CurEffect \end{cases}$$

Коррекция стандартов агента происходит при получении оценки *Feedback* (число от -1 до 1) своего или чужого действия *Action*. При этом учитывается отношение к оценившему действие агенту:

$$\begin{aligned} \text{Praiseworthiness}(Action) = & \text{Praiseworthiness}(Action) \cdot (1 - C_{Leam. Rate}) + \\ & + C_{Leam. Rate} \cdot \text{Feedback}, \end{aligned}$$

$$C_{Leam.Rate}^i = C_{Leam.Rate} \cdot (Liking + 1).$$

Отношение к другому агенту обновляется всякий раз, когда тот совершает действие, затрагивающее цели рассматриваемого агента. Это происходит в соответствии с формулой:

$$Liking = Liking \cdot (1 - C_{Leam.Rate}) + Pleasure \cdot C_{Leam.Rate},$$

где *Pleasure* – интенсивность радости/горя, вызванного действиями другого агента.

5. Детали реализации

Описанная модель была реализована в виде программной библиотеки на языке C++ с использованием следующих сторонних средств:

- библиотек из пакета boost (алгоритмы, структуры данных);
- nlohmann::json (работа с данными в формате JSON).

При использовании библиотеки пользователь или внешняя система описывает все сущности модели (агентов, цели, события и т.д.) с помощью соответствующих методов API или загрузив сохранённое в JSON состояние мира. Созданные экземпляры соответствующих классов в дальнейшем идентифицируются по заданным в описании уникальным строковым идентификаторам. Внешняя система общается с эмоциональной подсистемой с помощью асинхронных сообщений, информирующих о происходящих событиях или необходимости изменения каких-либо данных модели. Текущее эмоциональное состояние агента доступно напрямую, результат эмоциональной оценки какого-либо отдельного события доступен через экземпляр класса `std::future`, возвращаемый асинхронным вызовом.

При изменении представлений агента о текущем состоянии мира происходит перестроение планов по вычислению всех активных целей и вычисление вероятностей их достижения. Планирование различных целей по возможности происходит параллельно, увеличивая эффективность работы системы на устройствах с многоядерными процессорами. В качестве планировщика был реализован алгоритм YAHSP3, который отличается высокой скоростью работы, но более низким качеством построенных планов, что приемлемо при моделировании таких неточных и субъективных процессов, как человеческое мышление [3]. Планировщик реализует прямой эвристический поиск в пространстве состояний на основе алгоритма A* и оценки так называемой аддитивной эвристики [4]. При этом помимо прямых наследников каждого узла обходятся «перспективные» – достижимые из текущего посредством выполнения некоторого известного плана, а не единственного действия. Такие «перспективные» планы получаемы из последовательностей действий, которые обходятся при вычислении аддитивной эвристики [5].

Заключение

Данная статья посвящена моделированию эмоций виртуальных персонажей с целью повышения их правдоподобности. Описана математическая модель эмоционального, обучаемого агента на основе модифицированной оценочной теории эмоций ОСС и пятифакторной модели личности человека. Разработана библиотека, пригодная для внедрения в сторонние программные системы.

Список литературы

1. Ortony A. The Cognitive Structure of Emotions / A. Ortony, G.L. Clore, A. Collins. – New York: Cambridge University Press, 1988. – 224 p.
2. Первин Л. Психология личности: Теория и исследования / Л. Первин, О. Джон. – М.: Аспект Пресс, 2001. – 607 с.
3. Vidal V. A Lookahead Strategy for Heuristic Search Planning / V. Vidal // Proceedings of the 14th International Conference on Automated Planning and Scheduling (Whistler, June 3-7 2004). – Whistler, 2004. – P. 150-159.
4. Bonet B. Planning As Heuristic Search / B. Bonet, H. Geffner // Artificial Intelligence. – 2001. – №129. – P. 5-33.
5. Vidal V. YAHSP2: Keep It Simple, Stupid [Electronic resource] / V. Vidal. – Electronic text data. – Toulouse, 2011. – Mode of access: <http://v.vidal.free.fr/onera/publis/ipc11-yahsp2.pdf>

РАЗРАБОТКА УСТРОЙСТВА ДЛЯ МОНИТОРИНГА СОСТОЯНИЯ ОКРУЖАЮЩЕЙ СРЕДЫ ВНУТРИ ПОМЕЩЕНИЯ

В.И. Кулаков
Студент
Д.Н. Борисов
Доцент

Введение

С массовым внедрением высоких технологий в быт человека коренным образом изменился формат получения информации. Большое количество контента потребляется с мобильных устройств. Таким образом, актуальность и способ получения данных является ключевым моментом при построении информационной системы любого уровня.

Устройство, которое измеряет некоторые характеристики окружающей среды внутри помещения такие, как влажность воздуха, температура, количество угарного газа в воздухе, а также обнаруживает громкие звуки и движение, должно также своевременно уведомлять пользователя. В современном мире человек иногда сталкивается с отсутствием доступа к сети интернет через Wi-Fi или Ethernet подключение, но доступность данных и возможность их передачи являются неотъемлемой функцией устройства. Поэтому основной канал передачи данных — GSM сеть, взаимодействие в которой осуществляется с помощью sms-сообщений. Этот способ передачи данных наиболее предпочтителен, т. к. сотовая связь обладает огромным покрытием территорий, а sms-сообщения имеют низкую стоимость.

1. Постановка задачи

Разработать компактное устройство, которое способно:

- измерять температуру окружающей среды;
- измерять влажность воздуха;
- детектировать наличие угарного газа в воздухе;
- регистрировать движение и громкие звуки внутри помещения;
- поддерживать оповещение посредством SMS сообщений;
- предоставлять возможность гибко настраивать функциональность устройства и его чувствительность;
- отправлять полученные значения на сервер.

Основная задача разработки устройства заключается в детектировании внешних параметров и оповещение пользователя с помощью SMS сообщение. Устройство в режиме реального времени считывает показания с различных датчиков, сравнивает их с эталонными (запрограммированными) значениями и при расхождении полученных данных с эталонными уведомляет пользователь. По запросу пользователя устройство способно сообщить показания датчиков на текущий момент времени.

2. Физическая структура устройства

Разработанное устройство представляет собой конструктивно законченный корпус с установленными и подключенными внутри всеми необходимыми датчиками и модулями. Внешний вид устройства показан на рис. 1.

Пользователю нет необходимости вскрывать корпус с целью дополнительной настройки устройства. Конфигурирование устройства осуществляется с помощью GSM-команд, которые заранее предустановлены и сообщаются пользователю в документации устройства.

Для первичного запуска необходимо установить сим-карту и присоединить устройство к сети электроснабжения.



Рис. 1. Собранное устройство

Для активации устройства необходимо отправить sms-сообщение с кодовым словом на номер сим-карты, установленной в устройстве, и дальнейшее оповещение, взаимодействие с пользователем будет осуществляться по номеру телефона, отправившего sms-сообщение с кодовым словом. В архитектуре устройства имеется несколько датчиков и модулей соединенных между собой, перечень которых отображен на рис. 2.



Рис. 2. Компоненты устройства

В устройстве установлено четыре различных датчика для мониторинга различных параметров окружающей среды внутри помещения, а также два модуля связи и сирена для звукового оповещения.

В качестве микропроцессора был выбран Arduino Uno, к нему были подключены GSM-модуль SIM-900 и необходимые датчики [1].

Wi-Fi модуль обеспечивает подключение к одной из доступных сетей с целью передачи показаний датчиков и метаданных на сервер [2].

3. Входные/выходные данные

Входными данными для устройства являются параметры окружающей среды, эти параметры считываются датчиками устройства, данные обрабатываются и, при необходимости, отправляются пользователю или на сервер. Также входными данными считаются GSM-команды пользователя. При получении команды устройство интерпретирует и проверяет ее правильность, после чего производит действие, соответствующее полученной команде. Входные и выходные данные устройства показаны на рисунке 3.



Рис. 3. Входные и выходные данные устройства

Входные данные включают:

- Показания датчиков. Эти данные находятся в исходном, не преобразованном и не понятном для потенциального пользователя виде [3].

- GSM-команда от пользователя системы. Получив одну из предопределенных команд, устройство предпринимает действие, определенное в коде программы.

На выходе устройства – данные (т.е. показания датчиков, но уже в преобразованном виде, в совокупности со служебной информацией, передаваемой на сервер для последующей обработки и хранения), а также sms-оповещения, отправляемые пользователю.

4. Взаимодействие пользователя с устройством

Устройство взаимодействует с пользователем с помощью sms-сообщений. Пример входящего сообщения показан на рис. 4.

H: 21 T: 28.2 / MOTIONS!
NO SMOKE

Рис. 4. Пример приходящего сообщения

Сообщение, показанное на рис. 4, сообщает пользователю о том, что влажность в помещении 21%, температура 28.2°C, обнаружено движение, а уровень угарного газа ниже критического порога.

Пользователь взаимодействует с устройством с помощью специальных GSM-команд, перечень которых приведен в таблице. Устройство отвечает пользователю текстовыми сообщениями, содержимое которых варьируется в зависимости от полученной команды.

Таблица

Sms-команды

GET	
STOP	STOP T
	STOP PIR
	STOP SMOKE
STATUS	
RESUME	RESUME T
	RESUME PIR
	RESUME SMOKE

Типичный вариант использования (пользователю необходимо узнать температуру внутри помещения) выглядит следующим образом:

- Пользователь отправляет GSM-команду “GET T” на сотовый номер устройства.
- Устройство регистрирует текущие показания датчиков и формирует уведомление.
- Сформированное уведомление отправляется пользователю в виде короткого sms-сообщения.
- После отправки сообщения осуществляется дублирование данной информации: отправка данных на сервер для дальнейшей обработки и сохранения в качестве журнала логирования.

Команда GET позволяет узнать показания всех подключенных к устройству датчиков.

Команда STOP позволяет отключить датчики, как все вместе, так и по отдельности.

Команда STATUS позволяет узнать статус датчиков: включены они или отключены, имела ли место ошибка получения данных и т. п.

Команда RESUME включает отключенные датчики.

Если к команде STOP или RESUME дописать индекс необходимого датчика, то она будет применена только касательно этого датчика.

Данные команды позволяют легко настроить устройство, а также получать информацию о текущих показаниях датчиков через sms-общения.

Перечень команд, перечисленных в таблице, не является исчерпывающим, т. к. имеются команды для более точной, детальной настройки как всего устройства, так и отдельно взятых датчиков.

Регистрация показаний датчиков осуществляется в режиме реального времени постоянно. Оповещение пользователя происходит только по запросу или при выходе показаний за установленные рамки. Данные функции относятся к углубленной настройке устройства, т. к. позволяют не только получать уведомления, но и задавать допустимый коридор значений, при выходе за который будет формироваться уведомление в автоматическом режиме.

Заключение

Данная статья посвящена разработке устройства для считывания некоторых параметров окружающей среды внутри помещения. Описан принцип работы устройства и его физическая модель. Приведена модель взаимодействия пользователя с устройством, а также описана функциональность устройства, которое на сегодняшний день уже работает в полном объеме и оправдывает ожидания пользователей.

Список литературы

1. МакРобертс М. Начала Arduino / М. МакРобертс. – London : CUP, 2010. – 459 с.
2. Исакова Д.В. Разработка серверной части системы для мониторинга состояния окружающей среды внутри помещения / Д.В. Исакова // Сборник студенческих научных работ факультета компьютерных наук ВГУ; под редакцией Д.Н. Борисова. – 2017. – Выпуск 11. – С. 136-141.
3. Настройка модуля sim900 [Электронный ресурс] : GSM/GPRS модуль SIM900. Часть вторая. – Режим доступа : <http://we.easyelectronics.ru/part/gsm-gprs-modul-sim900-chast-vtoraya.html>

АЛГОРИТМ ЗАЩИТЫ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ СТЕГАНОГРАФИЧЕСКОГО «САМОВЛОЖЕНИЯ» КОНТЕЙНЕРА

А.Г. Ласкин

Студент

М.А. Дрюченко

Доцент

Введение

В наши дни информационные технологии являются одним из наиболее развивающихся направлений человеческой деятельности. Вместе с их развитием растет и значимость информационных ресурсов. Информационные ресурсы являются объектом права собственности и не менее ценным ресурсом, чем материальные или энергетические. Поэтому их защита признана сегодня наиболее актуальной проблемой. При этом защита от несанкционированного доступа это один из главных аспектов обеспечения безопасности информационных ресурсов. Еще в Древнем мире сформировалось два основных направления решения этой задачи: криптография и стеганография. Их общая цель состоит в скрытии секретной информации за счет шифрования, но стеганография, в отличие от криптографии, при этом позволяет скрыть сам факт наличия такой секретной информации.

В настоящее время особенно популярны исследования именно в области стеганографии. Можно выделить две причины становления такой ситуации: ограничение на использование криптосредств в ряде стран мира и появление проблемы защиты прав собственности на информацию, представленную в цифровом виде [1]. Например, довольно часто возникает такая ситуация, когда собственнику какого-либо информационного ресурса (книги, документа, фотографии и т.п.) необходимо предоставить другому лицу ограниченный доступ (просмотр в демо-режиме) к этому ресурсу с возможностью получения полного доступа в будущем по паролю или ключу (например, просмотр всего содержимого книги после оплаты ее стоимости). Для такого рода задач было решено разработать алгоритм защиты цифровых изображений на основе стеганографического «самовложения» контейнера.

Цель работы, которая рассматривается в рамках данной статьи, – анализ предметной области стеганографии и разработка программного прототипа, использующего предложенный алгоритм.

1. Анализ предметной области стеганографии

Стеганография – наука о способах передачи или хранения информации с учётом сохранения в тайне самого факта такой передачи (хранения). Как правило, сообщение будет выглядеть как что-то неприметное, например, изображение или документ. Стеганография по факту это встраивание в некоторый контейнер слабого шума, несущего скрытую информацию, с использованием стегоключа. Контейнер – любая информация, предназначенная для встраивания тайных сообщений. Стегоключ или просто ключ – секретный ключ, необходимый для скрытия (шифрования) информации [2]. Стегоконтейнер (или стего) – контейнер, содержащий встроеное сообщение.

Дадим также определение некоторых введенных нами понятий. Под самовложением будем понимать процесс получения искаженного изображения, в котором значащая информация визуально скрыта и незаметно внедрена в это же изображение. Наглядным примером защиты цифровых изображений путем их самовложения может служить предоставление неавторизованным пользователям лишь части исходного изображения, содержащей стеганографически скрытые недостающие фрагменты исходного растра. Под самореконструкцией будем понимать обратный процесс восстановления исходного или приближенного к нему цифрового изображения на основе его искаженного представления и имеющегося ключа шифрования или пароля. В терминах стеганографии исходное изображение является пустым контейнером, а искаженное изображение – заполненным контейнером.

В ходе работы были рассмотрены следующие методы встраивания скрытой информации в цифровые изображения [3]:

- метод наименьших значащих бит (LSB),
- метод сокрытия с использованием таблиц квантования,
- метод сокрытия с использованием дискретного косинусного преобразования (ДКП),
- ВРС и ABCDE.

Суть метода LSB заключается в замене последних, наименее значащих бит в контейнере на биты скрываемого сообщения. Методы этого класса обладают слабой статистической незаметностью и робастностью, но позволяют внедрить большое количество информации, являются простыми в реализации и быстрыми по скорости внедрения/извлечения информации.

Во втором методе для внедрения информации используются младшие биты коэффициентов квантования из таблиц квантования, которые используются, например в JPEG. Но данный метод обеспечивает очень малую пропускную способность.

Третий метод позволяет представлять изображение в виде набора коэффициентов дискретного косинусного преобразования и встраивать информацию в эти коэффициенты. Данный метод обладает относительно низкой пропускной способностью, но зато высокой робастностью.

Суть последних двух рассмотренных алгоритмов ВРС и АВСЕ очень схожа: изображение разбивается на битовые плоскости, затем шумоподобные блоки этих плоскостей заменяются на блоки скрываемого сообщения. Это позволяет добиться очень высокой пропускной способности, но робастность и статистическая незаметность остаются на низком уровне.

Одним из способов повысить пропускную способность методов встраивания информации является предварительное сжатие данных. В ходе работы были рассмотрены следующие базовые алгоритмы:

- RLE,
- алгоритм Хаффмана,
- JPEG.

Алгоритм RLE позволяет сжать данные за счет наличия длинных последовательностей повторяющихся элементов (например, нулей или единиц в двоичном представлении чисел).

Алгоритм Хаффмана использует для сжатия префиксные коды, которые формируются на основе вероятности появления элемента в последовательности.

Основные этапы алгоритма сжатия JPEG будут подробно рассмотрены в следующем разделе данной статьи.

2. Алгоритм «самовложения» контейнера

Перед описанием основных этапов разрабатываемого алгоритма введем следующие понятия. Пусть I – исходное изображение размером $W \times H$ пикселей, $I_{xy} \subset I$ – скрываемый фрагмент прямоугольной формы размером $w \times h$, $w \ll W, h \ll H$. Тогда $x = \overline{1, W - w}$, $y = \overline{1, H - h}$ – координаты скрываемого фрагмента в исходном изображении.

Алгоритм можно разделить на несколько этапов. На первом этапе происходит внесение визуальных искажений в области фрагмента I_{xy} (это может быть, например, заливка черным цветом):

$$I' = F_1(I, x, y, w, h). \quad (1)$$

На следующем этапе происходит стеганографическое встраивание фрагмента I_{xy} в искаженное изображение I' с использованием ключа k :

$$J = F_2(I', I_{xy}, k). \quad (2)$$

Для восстановления исходного представления изображения \tilde{I} необходимо знание ключа k :

$$\tilde{I} = F_3(I, k). \quad (3)$$

При этом для процедуры восстановления важно, чтобы восстановленное изображение \tilde{I} было визуально максимально близко к исходному I :

$$\|I - \tilde{I}\| \rightarrow \min. \quad (4)$$

А также желательно обеспечить максимизацию коэффициента эффективного использования, т.е. отношения площади скрываемого фрагмента к площади всего исходного изображения:

$$\frac{S(I_{xy})}{S(I)} = \frac{wh}{WH} \rightarrow \max. \quad (5)$$

Разработка программного прототипа, использующего предложенный алгоритм, была выполнена на языке C++ в среде Qt Creator версии 4.0.3 с использованием библиотеки libjpeg, содержащей большое количество функций для работы с изображениями формата JPEG.

В данной программе используется пример обработки изображений формата JPEG со встраиванием скрытой информации в младшие биты коэффициентов дискретного косинусного преобразования. Для реализации предложенного алгоритма был выбран именно такой способ, ввиду того что JPEG является одним из наиболее распространенных форматов цифровых изображений, а способ встраивания информации, содержащей фрагмент изображения, в коэффициенты ДКП обладает хорошими свойствами робастности. Процесс кодирования/декодирования JPEG с учетом встраивания данных включают в себя следующие этапы [4]:

- дискретизация,
- субдискретизация,
- дискретное косинусное преобразование,
- квантование,
- встраивание/извлечение информации,
- сжатие по алгоритмам RLE и Хаффмана.

На этапе дискретизации происходит преобразование цветового пространства RGB в пространство яркости/цветности. В виду того, что человеческое зрение более чувствительно к перепадам яркости, чем цветности, то на следующем этапе (этапе субдискретизации) происходит уменьшение цветового разрешения. Затем выполняется дискретное косинусное преобразование, позволяющее перейти из пространственного представления в спектральное. На этапе квантования происходит устранение избыточных высокочастотных коэффициентов. После этого

квантованные коэффициенты ДКП могут быть использованы для внедрения скрытой информации. Завершающим этапом служит сжатие при помощи алгоритмов RLE и Хаффмана.

Рассмотрим более подробно процесс самовложения (рис. 1) в данной реализации.

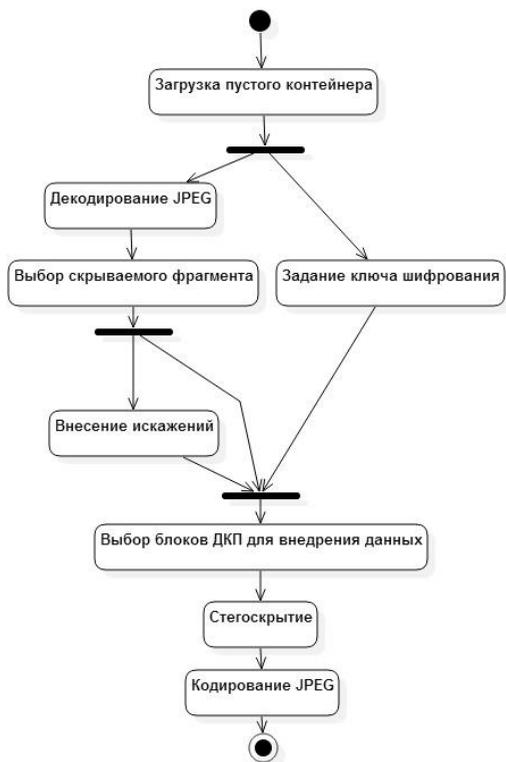


Рис. 1. Процесс самовложения изображения

После загрузки исходного изображения в формате JPEG происходит его декодирование, чтобы получить доступ к спектральному представлению изображения и его коэффициентам ДКП. Затем выбирается фрагмент, который необходимо скрыть, и задается ключ или пароль. Выбранный фрагмент закрашивается белым цветом. После этого генератор псевдослучайных чисел инициализируется ключом или паролем и псевдослучайным образом выбираются блоки ДКП для внедрения данных. Стегоскрытие происходит следующим образом: из коэффициентов ДКП

выбранного фрагмента изображения строится последовательность бит, в эту последовательность включается информация о расположении и размере скрываемого фрагмента, а также контрольная сумма, по которой проверяется правильность извлечения данных на этапе самореконструкции. Затем эти биты встают на место наименьших значащих бит коэффициентов ДКП в блоках, которые были определены на предыдущем шаге псевдослучайным образом. Последним этапом является кодирование обратно в формат JPEG.

На рис. 2 представлен процесс самореконструкции, позволяющий восстановить исходное представление изображения.

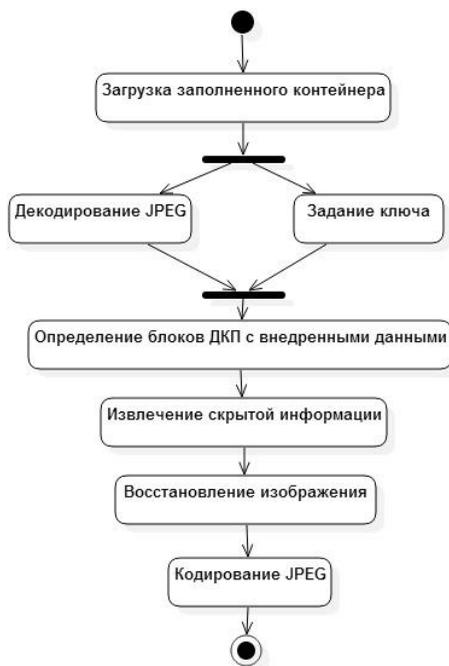


Рис. 2. Процесс самореконструкции изображения

На рис. 3 представлен результат работы программы на тестовом изображении «tiffany.jpg» из набора Kodak Lossless True Color Image Suite. Как видно, искажения за счет внедрения данных практически незаметны. Программный прототип, использующий предложенный алгоритм, в данной реализации позволяет скрыть и незаметно внедрить до 10% площади изображения.



а

б

в

а – исходное изображение; б – заполненный контейнер, содержащий стеганографически скрытый искаженный фрагмент; в – восстановленное изображение

Рис. 3. Пример работы алгоритма стеганографического самовложения

Заключение

В данной статье проанализирована предметная область стеганографии и описан разработанный на основе этого анализа программный прототип, использующий алгоритм защиты цифровых изображений на основе стеганографического «самовложения» контейнера. Предложенный алгоритм может быть использован для предоставления доступа к каким-либо документам или изображениям в режиме демо-просмотра с возможностью получения полного доступа при помощи ключа или пароля.

В перспективе другим вариантом использования является восстановление поврежденных или утраченных фрагментов изображений, документов. Таким образом, для развития предложенного алгоритма в будущем планируется добавить сжатие и шифрование скрываемого фрагмента, а также определить новые способы повышения пропускной способности.

Список литературы

1. Грибунин В.Г. Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев. – М. : СОЛОН-ПРЕСС, 2009. – 272 с.
2. Генне О.В. Основные положения стеганографии. / О.В. Генне // Защита информации. Конфидент. – 2000. – №3. – С. 20-24.
3. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. – М. : «МК-Пресс», 2006. – 288 с.
4. Аграновский А.В. Стеганография, цифровые водяные знаки и стеганоанализ / А.В. Аграновский, А.В. Балакин, В.Г. Грибунин, С.А. Сапожников. – М. : Вузовская книга, 2009. – 220 с.

МАЛОГАБАРИТНЫЙ БРАСЛЕТ ДЛЯ СНЯТИЯ ЭКГ

П.О. Левчук

Студент

С.А. Зуев

Ассистент

Введение

В настоящий момент времени для предупреждения пороков сердца, обнаружения аритмии и экстрасистол, а также других проблем с сердцем, необходимо снимать электрокардиограмму. Одним из вариантов, как это можно сделать, является периодическое посещение клиник или больниц для съёма ЭКГ, используя находящееся там оборудование. Более удобный вариант – использование портативного устройства, это позволяет следить за состоянием сердца на больших промежутках времени. Существует большое количество таких устройств, выполненных в форм-факторе пояса, чехла для телефона, отдельного устройства, но не в форме браслета.

1. Получение сигнала

Цель эксперимента в данной статье – создать необходимое ПО и аппаратную часть для сбора браслета, который сможет снимать электрокардиограмму.

Для снятия сигнала на кисть руки устанавливаются три хлоридсеребряных электрода. Они подключаются к усилителю на базе усилителя AD8232 (рис. 1 и рис. 2), в котором установлен аналоговый фильтр низких частот (35 Гц) для удаления помех связанных с активностью мышц. Затем усилитель подключается к микропроцессору с АЦП, где происходит цифровая фильтрация.

2. Описание цифровой фильтрации

Фильтр среднего частот 50-60 Гц позволяет убрать помехи электронной сети. Реализуется на основе массива, который содержит в себе (кол-во полученных значений за последнюю секунду / 60) элементов. Каждое последующее значение измерения суммируется со значениями в массиве и делится на их кол-во.

Фильтр Калмана представляет собой рекурсивный фильтр, оценивающий вектор состояния динамической системы, используя ряд неполных и зашумленных измерений [1]. Алгоритм фильтра состоит из двух этапов: прогнозирование и коррекция.



Рис. 1. Внешний вид усилителя с электродами

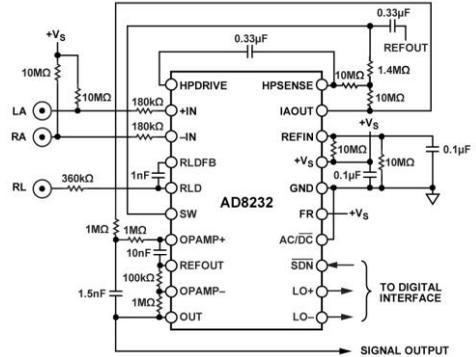


Рис. 2. Принципиальная схема усилителя

На этапе прогнозирования происходит экстраполирование вектора состояния системы и его ковариационной матрицы.

$$x = F * x + B * u;$$

$$P = (F * P * FT) + Q;$$

На этапе коррекции происходит вычисление отклонение полученного измерения от предсказанного на этапе прогнозирования, вычисление ковариационной матрицы вектора отклонения, вычисление матрицы с коэффициентами Калмана для текущего состояния системы, вычисление значения вектора состояния системы.

$$y = z - H * x;$$

$$S = (H * P * HT) + R;$$

$$K = (P * HT * Invariant(S));$$

$$x = x + K * y;$$

$$I_{KH} = I - K * H;$$

$$P = (I_{KH} * P * I_{KH}T) + (K * R * KT);$$

Схема подключения электродов для снятия сигналов ЭКГ с браслета показана на рис. 3.

3. Обучаемая нейронная сеть

Для улучшения процесса анализа сигналов ЭКГ, снимаемых с браслета, была предложено использовать нейронную сеть. В качестве алгоритма обучения будет использовано обучение с учителем. Для получения обучающей выборки и тестовых данных используются два канала съёма сигналов (рис. 4). В рабочем режиме – один (рис. 3). В качестве входных данных используется сигнал с руки, а в качестве учителя

– стандартное ЭКГ по Эйнтховену. Для обучения используется библиотека keras [2] на языке Python и Tensorflow [3].

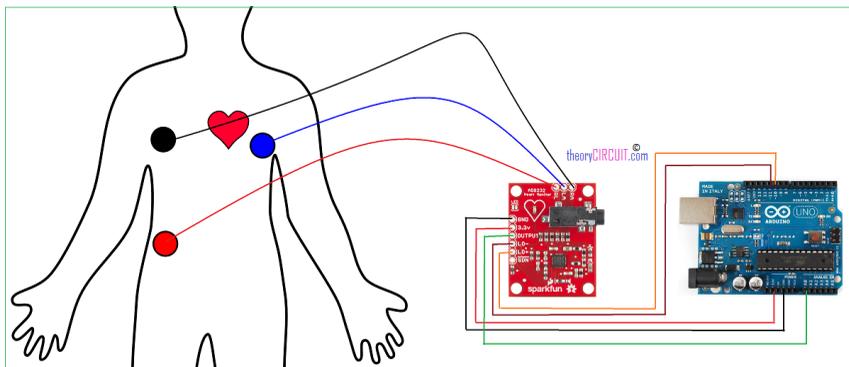


Рис. 3. Схема подключения электродов для снятия сигналов ЭКГ с браслета

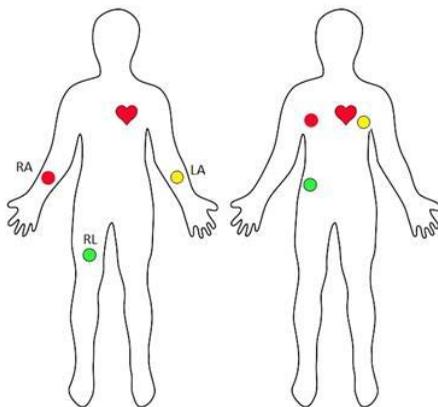


Рис. 4. Схема съёма кардиосигналов в режиме обучения нейронной сети

Заключение

Данная статья посвящена созданию малогабаритного браслета для снятия электрокардиограммы. Описано написание цифровых фильтров, которые используются для удаления шумов из получаемого сигнала. В итоге данного исследования, мы получили ПО для создания такого браслета. Использование такого устройства может помочь более эффективно следить за активностью сердца человека.

Список литературы

1. Фильтр Калмана [Электронный ресурс]: статья из Википедии. – Режим доступа : https://ru.wikipedia.org/wiki/Фильтр_Калмана
2. Keras: Deep Learning library for Theano and TensorFlow [Электронный ресурс] : – Режим доступа : <https://keras.io/>
3. An open-source software library for Machine Intelligence [Электронный ресурс] : – Режим доступа : <https://www.tensorflow.org/>

ОРГАНИЗАЦИЯ ПРОЕКТНОГО ОФИСА В КОМПАНИИ «WIZART ANIMATION»

А.В. Лоскутова

Студент

И.В. Абрамов

Доцент

Введение

На сегодняшний день проектные методы управления компаниями становятся всё более актуальными для достижения стратегических целей различных организаций, работающих над проектами. Проект — это временное предприятие, направленное на создание уникального продукта, услуги или результата. Итог проекта может быть, как материальным (осязаемым), так и нематериальным (неосязаемым).

Для успешной реализации проекта необходима четкая последовательность процессов, операций и процедур, т.е. проектом необходимо управлять. Управление проектом — это приложение знаний, навыков, инструментов и методов к работам проекта для удовлетворения требований, предъявляемых к проекту [1].

1. Проектный офис

Всё больше и больше организаций применяют в решении задач управления проектный офис. Офис управления проектами (ОУП) — организационная структура, стандартизирующая процессы руководства проектами и способствующая обмену ресурсами, методологиями, инструментами и методами [2]. Проектный офис – подразделение, отвечающее за методологическое и организационное обеспечение проектного управления в организации, планирование и контроль портфеля проектов, внедрение и развитие информационной системы планирования и мониторинга проектов, формирование сводной отчетности по программам/проектам. С введением проектного офиса становится

возможным не только анализ каждого проекта в отдельности, но и общей картины состояния всех проектов организации, что влечет за собой оптимизацию принятия решений. Так же с организацией проектного офиса, применением методов управления проектами, системой своевременного сбора и анализа информации о ходе исполнения проектов организации становится возможным принимать просчитанные и своевременные решения, больше сил уделяя вопросам трагического развития.

Проектные офисы классифицируются по различным типам выполняемых функций, но ни одна из них не является идеальной. Можно разделять функции ОУП на основные и дополнительные. Основные функции используются всеми проектными офисами, расширенные подбираются в зависимости от поставленных задач перед ОУП [3].

1. Основные функции:
2. Контроль соблюдения методологии управления, а также разработка и внедрение методологии.
3. Ведение и формирование сводной отчетности по проектам.
4. Помощь в планировании и контроль реализации отдельных проектов.

Дополнительные функции:

1. Обеспечение работы проектного комитета.
2. Централизованное выделение руководителей или администраторов проектов.
3. Выбор, внедрение, поддержка и развитие информационной системы планирования и мониторинга проектов.
4. Внедрение и поддержка системы стимулирования участников проектной деятельности.
5. Организация обучения проектному управлению или непосредственно обучение.
6. Аудит проектов.
7. Управление знаниями в области управления проектами.
8. Портфельное управление.

2. Объект исследования

Воронежская компания «Wizart Animation» анимационная студия, которая работает над созданием высококачественный семейных анимационных мультфильмов. Компания организует свою работу на основании оценки потребностей рынка. Выполнение работ по производству любого вида анимации осуществляется в виде проекта. В данный момент компания работает над тремя проектами одновременно. Для улучшения управления деятельности в компании предлагается организовать проектный офис, который будет осуществлять функцию

обратной связи и позволит более эффективно контролировать работу над проектами.

Ожидаемым результатом от организации проектного офиса в анимационную студию является: создание единой системы по управлению разными направлениями проектной деятельности компании, включающей описание и регламентацию основных процессов управления ими, структуру проектного офиса, с распределением функций, задач и ролей каждого структурного элемента, единое информационное пространство, методику инициации проекта, систему оценки и контроля. Важная роль проектного офиса заключается в координации рабочей группы и этапах поиска, разработки, а также внедрения и реализации проектных управленческих решений.

Для разработки единой системы управления и реализации проектов необходимо решить следующие задачи:

- разработать регламентацию всех процессов внутри проектного офиса;
- разработать стандарты ведения проектов;
- разработать систему контроля и оценки всей проектной деятельности;
- создать единую информационную среду.

3. Организационная структура студии

Для начала была произведена формализация деятельности компании. Разработаны функциональные и технологические модели студии, отражающие все бизнес-процессы.

Обычно, процесс создания фильма проходит в несколько фаз: проектирование (создание концепции фильма), предварительная подготовка (написание сценария, прорисовка персонажей, раскадровка, черновая анимация), сборка и монтаж (редактирование анимации, подготовка цветовых схем, постановка музыки и звука, монтаж, финальный вывод на носители). Поэтому организационная структура компании «Wizart Animation» включает в себя множество производственных отделов, управление которыми является затруднительным. Организационная структура представлена на рис. 1. Смешанная отчетность не позволяет вместить состояние всех проектов. Необходимость сосредоточить всю информацию, сохранить, стандартизировать ее – вот одна из задач проектного офиса.

Организация проектного офиса позволяет:

- организовать управление по целям и исключение дублирования целей различных проектов;

- сформировать выверенный портфель проектов, планировать проекты с учетом их значимости и наличия необходимых ресурсов; оперативно перераспределять ресурсы;
- комплексно планировать сроки и результаты;
- повысить квалификацию специалистов в области управления и реализации проектов.

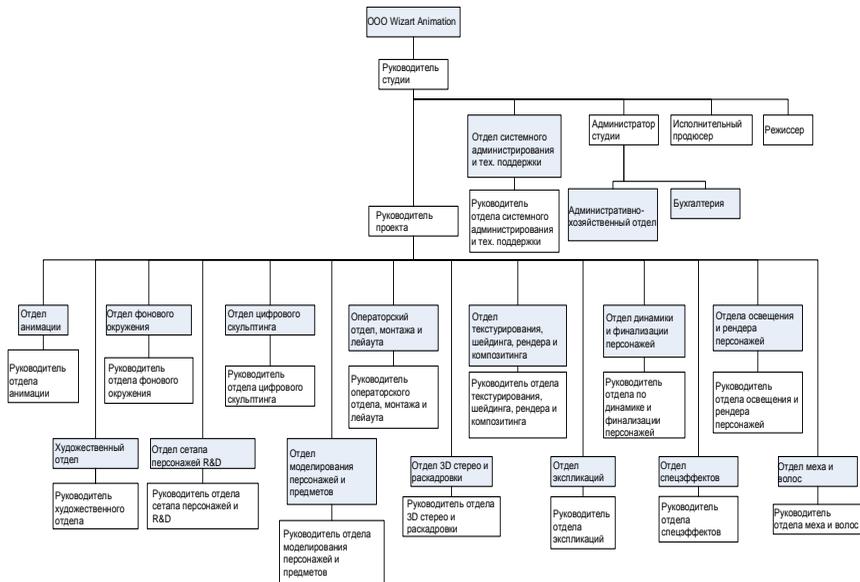


Рис. 1. Орг. структура «Wizard Animation»

4. Ожидаемые результаты внедрения ОУП в «Wizard Animation»

Возможность вести анализ всей работы студии через портфель проектов. До организации проектного офиса разработка исполнения и ведения отчетности, единого расписания осуществлялась в разных формах и часто не совпадала по срокам и возникали накладки. Возможен был анализ каждого проекта в отдельности, а не портфеля проектов. Благодаря созданию проектного офиса и стандартизации проектов становится возможным планирование всей работы компании; состыковки проектов по времени проведения и использованию ресурсов.

Качество и количество информации. Ранее процесс эффективного принятия решений тормозился либо избыточной, либо недостаточной информацией. С применением методов управления проектами и архивирования проектов становится возможной работа с максимально выверенной информацией, что позволяет оптимизировать управление

проектом; при смене менеджера проекта, работа над ним не останавливается как было ранее.

Заключение

В статье актуализируется проектный метод управления в мультипликационной студии. Проектный способ направлен на формирование командного ресурса для поиска инициативных совместных управленческих решений. Офис управления связывает данные и информацию, взятые из корпоративных стратегических проектов, и определяет степень выполнения стратегических задач более высокого уровня. В целях оптимизации временных, финансовых ресурсов особую роль отводится работе проектного офиса, как координатора между участниками проекта. От организации работы проектного офиса зависит успешность выполнения проекта.

Список литературы

1. Мартин, П. Управление проектами [Текст] / П. Мартин, К. Тейт; пер. с англ. – СПб. : Питер, 2006. – 224 с.
2. Руководство к своду знаний по управлению проектами (Руководство PMBOK®). – 5-е издание / Project Management Institute, Inc. 2013. – 614 с.
3. PMPHelp. Всё об управлении проектами [Электронный ресурс]. – Режим доступа: <http://www.pmphelp.net>

ПРОЕКТИРОВАНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ВЫБОРА УЧЕБНЫХ ЭЛЕМЕНТОВ LMS MOODLE

В.А. Мелихов

Студент

А.П. Толстобров

Доцент

Введение

В настоящее время проблема организации электронного курса в системе Moodle становится все более актуальной. Важнейшими компонентами электронного курса являются так называемые деятельностные учебные элементы (англ. activity), которые предоставляют возможность преподавателю задавать студентам различные учебные поручения, проверять результат выполнения поручений студентами и давать им оценку.

В Moodle существует 14 различных типов деятельностных учебных элементов [1, 2], предоставляющие очень широкие возможности по реализации в электронной среде самых разнообразных видов учебных поручений, используемых в практике учебного процесса. Они отличающихся по виду выполняемых обучаемым действий, по форме взаимодействия преподавателя и обучаемого, по форме представления обучаемым результатов выполнения учебного поручения, по способу оценивания этих результатов и по целому ряду других факторов.

Однако, обратной стороной разнообразия типов деятельностных учебных элементов Moodle и предоставляемых ими функциональных возможностей является то, что перед преподавателем, являющимся разработчиком электронного курса (в особенности имеющим небольшой опыт) встает очень непростая проблема правильного выбора конкретного типа элемента курса Moodle, наилучшим образом подходящего для реализации нужного ему учебного поручения. Это при том, что названия этих элементов (или их перевод) весьма слабо отражают их назначение и возможности, а многие, предоставляемые различными элементами возможности, являются или выглядят очень схожими. Например, возможность представления обучаемыми результатов выполнения какой-либо работы в форме текста или файла поддерживается несколькими типами таких элементов, при том, что другие возможности этих элементов могут существенно отличаться.

Каждый тип оцениваемого деятельностного элемента Moodle представляет собой программный модуль, набор реализуемых разнообразных возможностей которого обеспечиваются соответствующей настройкой многочисленных параметров модуля [3].

Целью данной работы является проектирование экспертной системы Moodle, которая позволит преподавателю – разработчику электронного курса на основе своих пожеланий, предъявляемых к учебному поручению, которое он хочет реализовать, подобрать наиболее подходящий вид деятельностного учебного элемента соответствующую настройку его параметров.

1. Экспертные системы

Экспертные системы (ЭС) – прикладные программы ИИ искусственного интеллекта, в которых база знаний представляет собой формализованные знания высококвалифицированных специалистов (экспертов) предметной области [4]. В нашем случае, специалиста в области использования и настройки учебных элементов системы Moodle.

В целом процесс функционирования разрабатываемой ЭС можно представить следующим образом: преподаватель, желающий получить определенный функционал учебного элемента, через пользовательский

интерфейс посылает запрос к ЭС; решатель ЭС, пользуясь базой знаний, предлагает преподавателю оцениваемый элемент Moodle, который реализует требуемый преподавателю функционал наиболее подходящим образом, объясняя ход своих рассуждений при помощи подсистемы объяснений. На рис. 1 представлена структура экспертной системы.

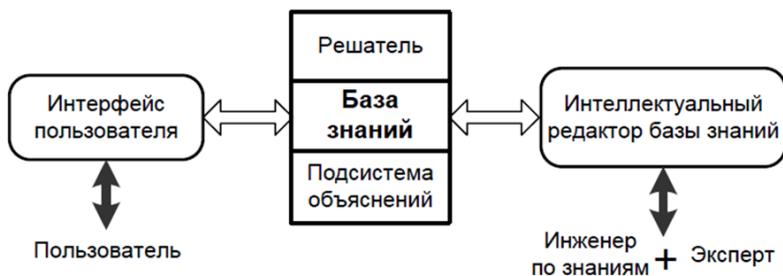


Рис. 1. Структура экспертной системы

Диалог конечного пользователя с ЭС как на стадии ввода информации, так и при получении результатов осуществляется через интерфейс пользователя. В базе знаний содержится совокупность знаний предметной области, записанная на машинный носитель в форме, понятной эксперту и пользователю. Решатель – это программа, моделирующая ход рассуждений эксперта на основании информации, имеющейся в БЗ. Подсистема объяснений — программа, позволяющая пользователю получить ответы на вопросы: «Как была получена та или иная рекомендация?», т.е. описание всего процесса получения решения с указанием использованных фрагментов БЗ. Интеллектуальный редактор базы знаний — программа, предоставляющая инженеру по знаниям возможность создавать БЗ в диалоговом режиме. Инженер по знаниям — специалист в области искусственного интеллекта, выступающий в роли посредника между экспертом и базой знаний [5].

2. Модель представления знаний

Так как в качестве условия и действия в правилах разрабатываемой системы является предположение о наличии или отсутствии того или иного функционала у учебного элемента, в качестве модели представления знаний была выбрана продукционная модель.

На рис. 2 представлена архитектура продукционной экспертной системы.

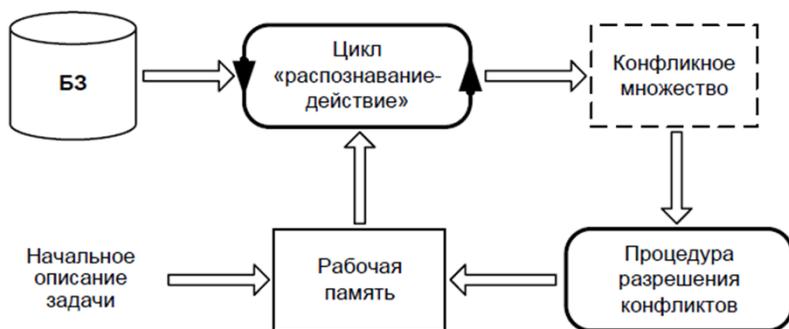


Рис. 2. Архитектура продукционной экспертной системы

В БЗ содержится множество продукционных правил или просто продукций (productions), являющихся парами условие–действие, которые определяют элементарные шаги решения задачи. Условная часть (IF–part) правила является шаблоном (образцом), по которому можно определить, в какой момент необходимо использовать (активировать) данное правило для выполнения очередного этапа решения задачи. Часть действия (THEN–part) описывает соответствующий шаг решения. Условную часть правила также называют антецедентом (antecedent), а часть действия — консеквентом (consequent).

Рабочая память (working memory) содержит текущее описание модели мира в процессе рассуждений. В этой модели содержится набор образцов, инициализируемый начальным описанием задачи.

В управляющем цикле распознавание–действие (the recognize — act cycle) осуществляется сравнение образцов из рабочей памяти с условными частями правил в БЗ. Если условие какого-либо правила соответствует образцу, то это правило помещается в конфликтное множество (conflict set). Продукции, содержащиеся в конфликтном множестве, называют допустимыми, так как они согласованы с текущим состоянием рабочей памяти. После того, как закончит работу цикл распознавание–действие, осуществляется процесс разрешения конфликтов (conflict resolution), в ходе которого выбирается и активизируется (возбуждается) одна из допустимых продукций. Наконец, в соответствии с частью действия активированного правила осуществляется модификация рабочей памяти. Весь этот процесс повторяется до тех пор, пока образцы в рабочей памяти не будут соответствовать ни одному из правил БЗ.

Заключение

Данная статья посвящена проектированию экспертной системы для выбора учебных элементов LMS Moodle. Предполагается что разрабатываемая экспертная система позволит преподавателям, связавшим свою профессиональную деятельность с электронным обучением, более эффективно и точно осуществлять выбор и настройку необходимых им при создании электронных курсов элементов системы Moodle.

Список литературы

1. Документация LMS Moodle [Электронный ресурс]: официальный сайт Moodle. – Режим доступа : https://docs.moodle.org/30/en/Main_page
2. Russian Moodle [Электронный ресурс]: официальный сайт Moodle – Режим доступа : <https://moodle.org/login/index.php>
3. Толстобров А.П. Методика выбора типа деятельностного элемента электронного курса LMS Moodle для реализации в курсе оцениваемого учебного поручения / А.П. Толстобров. Информатика : проблемы, методология, технологии: 7-я Международная школа-конференция «Информатика в образовании» (Воронеж 11-12 февраля 2016 г.). – Воронеж, 2016. – С. 677-685.
4. Болотова Л.С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях / Л.С. Болотова. – М. : Информика, 2012. – 664 с.
5. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб. : Питер, 2014 – 384 с.

ИССЛЕДОВАНИЕ СИММЕТРИЙ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Е.Ю. Морозов
Студент
А.В. Лобода
Профессор

Введение

Симметрии различного вида представляют в современной науке немалый интерес. В данной работе исследуются симметрии одного семейства quadro-кубических поверхностей в трехмерном комплексном пространстве. При этом используется теория групп преобразований Ли.

Также работа тесно связана с изучением свойств однородности поверхностей в \mathbb{C}^3 .

1. Постановка задачи

Пусть дано семейство вещественных гиперповерхностей в пространстве \mathbb{C}^3 :

$$v x_2 + Q(x_1, y_1, x_2, y_2) = x_2(\mu x_2^2 + v y_2^2), \quad (1)$$

где $x_1 = \Re(z_1)$, $y_1 = \Im(z_1)$, $x_2 = \Re(z_2)$, $y_2 = \Im(z_2)$, $u = \Re(w)$, $v = \Im(w)$ – компоненты координат в трехмерном комплексном пространстве, $\mu, v \in \mathbb{R}$ и одновременно не равны нулю, а $Q(x_1, y_1, x_2, y_2)$ — некоторая квадратичная форма. Симметрией поверхности из данного семейства называется любое аффинное преобразование, сохраняющее эту поверхность. Симметрией определяющего уравнения для заданной поверхности будем называть преобразование, сохраняющее это уравнение с точностью до ненулевого множителя. Задача этой работы состоит в изучении групп аффинных преобразований, сохраняющих конкретные поверхности из данного семейства, и размерностей таких групп.

Отметим, что можно изучать поставленную задачу с точностью до аффинных преобразований исходных объектов. На размерности и некоторые другие свойства изучаемых групп это не повлияет.

2. Метод решения

Видно, что любая поверхность вида (1) сохраняется сдвигом по переменной $u = \Re(w)$. Таким образом, размерность каждой из изучаемых групп преобразований не меньше единицы. Для изучения групп больших размерностей удобно использовать однопараметрические группы преобразований.

Обозначим через F_t однопараметрическую группу аффинных преобразований в \mathbb{C}^3 , сохраняющих уравнение поверхности $\Phi = 0$. Группа F_t зависит от параметра t аналитически, что возможно в группе Ли. Уравнения, определяющие эту группу, будут иметь вид:

$$z_i = \sum_{j=1}^3 A_{i,j}(t) z_j + S_i(t), \quad i = 1, \dots, 3, \quad (2)$$

где $A_{i,j}(t)$, $S_i(t)$ – некоторые аналитические комплексные функции от вещественного аргумента, а $z_3 = w = u + iv$. Также эта группа должна содержать тождественное преобразование, т.е. $F_t = \text{Id}$.

Факт сохранения некоторой поверхности M из заданного семейства (1) такой группой можно записать в виде

$$F_t(\Phi) = \Phi(z, z, v) \cdot \psi(z, z, w, t), \quad (3)$$

где $\psi(z, z, w, t)$ – некоторая ненулевая функция, а $\Phi = \Phi(z, z, v) = vx_2 + Q(x_1, y_1, x_2, y_2) - x_2(\mu x_2^2 + v y_2^2)$ – определяющая функция поверхности M .

Продифференцируем (3) по параметру t в точке $t = 0$:

$$\frac{d}{dt} F_t(\Phi) \Big|_{t=0} = \Phi(z, z, v) \cdot \frac{d}{dt} \psi(z, z, w, t) \Big|_{t=0}. \quad (4)$$

Левая часть выражения (4) называется действием соответствующего группы F_t инфинитезимального преобразования на поверхность M . Инфинитезимальное преобразование ставит в соответствие каждой точке p поверхности M вектор, касательный к p . Иными словами, оно показывает направление движения точки p в результате действия группы F_t для бесконечно малых t [1].

В сужении (4) на поверхность M правая часть обращается в ноль, так как $\Phi|_M \equiv 0$. Таким образом получаем тождество

$$\left\{ \frac{d}{dt} F_t(\Phi) \Big|_{t=0} \right\} \Big|_M \equiv 0. \quad (5)$$

Подставляя в (5) переменную $v = Q(x_1, y_1, x_2, y_2) / x_2 + (\mu x_2^2 + v y_2^2)$ из (1) и умножая на возникающий знаменатель, получаем полиномиальное уравнение:

$$P(x_1, y_1, x_2, y_2, u) = 0.$$

Коэффициентами при мономах в полиноме P являются линейные комбинации значений производных функций из уравнения (2) в точке $t = 0$. Обозначим их следующим образом:

$$\begin{aligned} A_{1,j}'(t) &= \alpha_{j,1} + i\alpha_{j,2}, \quad A_{2,j}'(t) = \beta_{j,1} + i\beta_{j,2}, \quad A_{3,j}'(t) = \gamma_{j,1} + i\gamma_{j,2}, \\ S_j'(t) &= \sigma_{j,1} + i\sigma_{j,2}, \quad j = 1, \dots, 3. \end{aligned} \quad (6)$$

Тождественное равенство нулю полинома P означает равенство нулю всех коэффициентов при его одночленах. Так получается однородная система линейных уравнений относительно 24 неизвестных (6). Каждое уравнение в этой системе отвечает одному моному, общее число которых для произвольной поверхности (1) равно 83 (т.к. полином P содержит одночлены, имеющие с третьей по шестую степень). Обозначим через W матрицу, соответствующую этой системе. Искомая размерность группы аффинных преобразований, сохраняющих такую поверхность, описывается равенством

$$\dim G = 24 - \text{rank } W.$$

Таким образом, исследование размерности группы аффинных преобразований, сохраняющих поверхность M , сводится к исследованию ранга полученной матрицы W .

Следует заметить, что коэффициенты (6) являются элементами инфинитезимального преобразования [1], которое удобно записывать в виде матрицы:

$$S = \begin{pmatrix} \alpha_{1,1} + i\alpha_{1,2} & \alpha_{2,1} + i\alpha_{2,2} & \alpha_{3,1} + i\alpha_{3,2} & \sigma_{1,1} + i\sigma_{1,2} \\ \beta_{1,1} + i\beta_{1,2} & \beta_{2,1} + i\beta_{2,2} & \beta_{3,1} + i\beta_{3,2} & \sigma_{2,1} + i\sigma_{2,2} \\ \gamma_{1,1} + i\gamma_{1,2} & \gamma_{2,1} + i\gamma_{2,2} & \gamma_{3,1} + i\gamma_{3,2} & \sigma_{3,1} + i\sigma_{3,2} \end{pmatrix}. \quad (7)$$

Интегрируя отдельные инфинитезимальные преобразования, полученные в результате решения системы уравнений, получаем однопараметрические группы аффинных преобразований, сохраняющих поверхность M . Подробнее об интегрировании инфинитезимальных преобразований можно узнать из [2].

3. Общий случай

Для произвольной quadro-кубической поверхности вида (1), квадратичная форма Q может быть записана в явном виде:

$$Q(x_1, x_2, y_1, y_2) = k_1 x_1^2 + k_2 x_1 x_2 + k_3 x_2^2 + k_4 x_1 y_1 + k_5 x_2 y_1 + k_6 y_1^2 + k_7 x_1 y_2 + k_8 x_2 y_2 + k_9 y_1 y_2 + k_{10} y_2^2. \quad (8)$$

За счет применения невырожденных аффинных преобразований поверхности вида (1) можно привести эту поверхность к некоторому каноническому виду с меньшим числом коэффициентов в форме (8), но с большим числом рассматриваемых случаев.

Для исследования системы, описанной в предыдущем пункте, использовался пакет компьютерной алгебры Wolfram Mathematica. Результатом исследования является оценка ранга матрицы W полученной системы: $19 \leq \text{rank } W \leq 23$. Следствием этой оценки является следующая теорема:

Теорема 1: размерность группы Ли аффинных преобразований, сохраняющих любую поверхность из семейства (1), удовлетворяет неравенству:

$$1 \leq \dim G \leq 5.$$

Пример 1: единичная размерность группы достигается на поверхности

$$vx_2 + |z_1|^2 + x_1 y_1 + y_2^2 = x_2 |z_2|^2.$$

Эта поверхность сохраняется лишь одним типом аффинных преобразований – сдвигом по переменной u .

Пример 2: $\dim G = 2$ достигается на поверхности

$$vx_2 + |z_1|^2 + y_2^2 = x_2 |z_2|^2.$$

У этой поверхности, помимо сдвига по переменной u , имеется еще один тип движений – повороты в плоскости z_1 : $z_1 \rightarrow e^{it} z_1^*$.

Поверхности, которым соответствуют пятимерные группы аффинных преобразований, представляют особый интерес, например, потому, что эти поверхности могут являться *однородными*. Поверхность называется аффинно-однородной, если под действием аффинных преобразований можно сдвигаться в любом направлении вдоль этой поверхности.

4. Частный случай

Детально был рассмотрен случай, в котором квадратичная форма (8) не зависит от x_2 и y_2 , т. е. :

$$v x_2 + k_1 x_1^2 + k_2 x_1 y_1 + k_3 y_1^2 = x_2 (\mu x_2^2 + \nu y_2^2). \quad (9)$$

Замечание: за счёт аффинных преобразований в плоскости z_1 , набор k_1, k_2, k_3 может быть фактически сокращен до одного параметра.

Для этого случая справедлива следующая оценка размерности группы преобразований G :

Теорема 2: размерность группы Ли аффинных преобразований, сохраняющих любую поверхность вида (9), удовлетворяет неравенствам $3 \leq \dim G \leq 5$, причем

- $\dim G = 3$ достигается на поверхностях вида

$$v x_2 = k x_1^2 + y_1^2 + x_2 (\mu x_2^2 + \nu y_2^2), \quad k \neq 0, \nu \neq 1; \quad (10)$$

- $\dim G = 4$ достигается на поверхностях вида

$$v x_2 = |z_1|^2 + x_2 (\mu x_2^2 + \nu y_2^2); \quad (11)$$

- $\dim G = 5$ достигается на аффинно-однородных поверхностях вида

$$v x_2 = x_1^2 + x_2 (\mu x_2^2 + \nu y_2^2). \quad (12)$$

Таким образом, для произвольной поверхности вида (9) существует три основных типа движений, которые сохраняют эту поверхность:

1. *сдвиг* по переменной u : $w \rightarrow w^* + t$;
2. *масштабирование*: $z_1 \rightarrow e^{3t} z_1^*$, $z_2 \rightarrow e^{2t} z_2^*$, $w \rightarrow e^{4t} w^*$. Такой тип аффинных преобразований привносит в обе части уравнения (9) ненулевой множитель e^{6t} ;
3. *поворот со сдвигом* (“скользящий поворот”): $z_1 \rightarrow z_1^*$, $z_2 \rightarrow z_2^* + it$, $w \rightarrow w^* + 2tv \cdot z_2^* + it^2 v$. Это движение добавляет в обе части (9) слагаемые $2tv x_2 y_2 + t^2 v x_2$, которые взаимно уничтожатся, в результате чего получается исходное уравнение.

У поверхностей (11) имеется еще одно движение – *повороты в плоскости* z_1 , описанные в примере 2 предыдущего раздела. Поверхности (10) и (11) представляют интерес в рамках классификации действий групп аффинных преобразований на многообразиях [3].

Поверхности (12), в дополнение к трем основным, имеют еще два типа движений:

1. *сдвиг* по переменной y_1 : $z_1 \rightarrow z_1^* + it$;
2. *поворот*: $z_1 \rightarrow z_1^* + t \cdot z_2^*$, $z_2 \rightarrow z_2^*$, $w = 2it \cdot z_1^* + it^2 \cdot z_2^* + w^*$. В

результате такого движения в обе части (9) добавляются слагаемые $2t x_1 x_2 + t^2 x_2$, после взаимного уничтожения которых получается исходное уравнение.

Семейство аффинно-однородных поверхностей вида (12) известно, оно было описано в работе

Еще один результат данной работы расширяет класс аффинно-однородных поверхностей, представленный в [4], за счет более богатых кубических частей, содержащихся в уравнениях этих поверхностей

Теорема 3: для любой тройки коэффициентов μ, ν, λ , одновременно не равных нулю, поверхность

$$\nu x_2 + x_1^2 = x_2(\mu x_2^2 + \lambda x_2 y_2 + \nu y_2^2),$$

является аффинно-однородной, а размерность группы Ли аффинных преобразований, сохраняющих такую поверхность, равна 5.

5. Заключение

В данной работе были изучены симметрии одного семейства квадратно-кубических поверхностей в пространстве S^3 . В ходе работы была получена оценка для размерностей групп Ли аффинных преобразований, сохраняющих данные поверхности. Также был произведен разбор частного случая исходного семейства поверхностей, что дало конкретные примеры поверхностей для различных размерностей. В ходе решения было произведено обобщение известных классов аффинно-однородных поверхностей из работы [4].

Список литературы

1. Ли, С. Теория групп преобразований: в 3-х частях. Часть 1 / С. Ли; пер. с нем. Л.А. Фрай; под ред. А.В. Болсинова. – М. : Ижевск: Институт компьютерных исследований, 2011. – 693 с.
2. Лобода, А.В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства / А. В. Лобода, А. С. Ходарев // Известия вузов. Матем. – 2003. – № 10. – С. 38-50.

3. Isaev, A. On the symmetry algebras of 5-dimensional CR-manifolds [Электронный ресурс] / A. Isaev, B. Kruglikov // ArXiv e-prints. – 2016. – Электрон. журн. – Режим доступа : <https://arxiv.org/abs/1607.06072>

4. Atanov, A.V. Affine homogeneous strictly pseudoconvex hypersurfaces of the type $(1/2,0)$ in C^3 [Электронный ресурс] / A.V. Atanov, A.V. Loboda, A.V. Shipovskaya // ArXiv e-prints. – 2014. – Электрон. журн. – Режим доступа : <https://arxiv.org/abs/1401.2252>

РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ КЛАСТЕРНОГО АНАЛИЗА ДАННЫХ ДЛЯ ОБЪЕКТИВНОЙ ОЦЕНКИ ЧИСЛА ПОРОЖДАЮЩИХ КЛАССОВ

Д.Э. Мубаракшин

Студент

А.А. Сирота

Профессор

Введение

В настоящее время кластерный анализ является одним из наиболее распространенных средств, используемых в сфере машинного обучения.

История развития данного направления берет свое начало с незапамятных времен: на протяжении всего своего существования человечество стремилось искать и собирать знания об окружающей действительности. Впоследствии эти знания и опыт могли быть использованы в частности и для упорядочивания групп объектов по каким-либо признакам [1].

Кластеризация является частным случаем классификации и относится к группе алгоритмов обучения без учителя, особенностью которых является то, что обучающие объекты рассматриваются как однородные («непомеченные»).

На настоящий момент выделяются следующие основные области применения алгоритмов кластеризации:

- Биоинформатика;
- Экология;
- Социология;
- Информатика.

Также одним из наиболее перспективных направлений развития кластеризации является медицина. В частности, такие алгоритмы могут использоваться для выделения подгрупп пациентов, имеющих как разные

заболевания, так и различные патологии одного и того же заболевания. Однако нередко возникают ситуации, когда число возможных состояний больных не может быть определено однозначно по ряду объективных причин. В таких случаях количество кластеров обычно задается «вручную».

Основной целью данной работы является сравнение критериев качества оценки кластеризации и применение одного из них для анализа реальных данных.

1. Решение задачи в случае неизвестного числа кластеров

Случай, когда число кластеров изначально не задано, считается одним из самых трудных в области кластерного анализа. Решение может быть основано на применении одно из двух стандартных подходов. Первый предполагает последовательный перебор возможных значений числа кластеров. В итоге выбирается тот вариант, который наилучшим образом удовлетворяет заранее выбранному критерию. Второй подход основан на предварительной оценке числа групп без проведения кластеризации и подстановке полученного результата в готовый алгоритм. В случае если не нужно ограничивать объем проводимых вычислений, обычно выбирают первый подход как более эффективный.

Рассмотрим основные критерии, которые могут быть использованы для оценки числа кластеров:

– Индекс Калинского-Харабаша [2].

Его основная идея – максимизация отношения показателей межклассового и внутриклассового разброса. Формула для вычисления значения индекса использует перечисленные показатели и имеет вид:

$$I_{CH}(\tilde{M}) = \frac{E_B(X^N, Q_{\tilde{M}})(N - \tilde{M})}{E_W(X^N, Q_{\tilde{M}})(\tilde{M} - 1)}. \quad (1)$$

– Индекс Кржановского-Лая [3].

Этот критерий выявляет максимальный скачок внутриклассовой дисперсии в процессе последовательного изменения числа кластеров. Формула для расчета:

$$I_{CL}(\tilde{M}) = \left| \frac{\Delta E_W(\tilde{M})}{\Delta E_W(\tilde{M} + 1)} \right|, \quad (2)$$

$$\Delta E_W(\tilde{M}) = (M - 1)^{1/n} E_W(X^N, Q_{\tilde{M}-1}) - M^{1/n} E_W(X^N, Q_{\tilde{M}}).$$

– Индекс Дэвиса-Болдуина [4].

Для его вычисления сначала находят среднюю схожесть между каждым кластером и другим, расположенным к нему ближе всего. На каждом шаге изменения числа кластеров вычисляется степень схожести

элементов в каждой группе и степень отличия между группами в целом (в качестве метрики может использоваться как Евклидово расстояние, так и другая метрика):

$$e_{wi} = \left[\frac{1}{N_i} \sum_{k=1}^{N_i} \|x^{(k,i)} - m_i\|^p \right]^{1/p}, \quad h_{ij} = \left[\sum_{k=1}^n |m_{i,k} - m_{j,k}|^q \right]^{1/q} \quad (3)$$

В итоге индекс вычисляется по формулам:

$$I_{DB}(\tilde{M}) = \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} R_i, \quad R_i = \max_j R_{ij}, \quad (4)$$

$$R_{ij} = (e_{wi} + e_{wj}) / h_{ij}. \quad (5)$$

– GAP-индекс [5].

Проводится анализ статистики расхождений («гар» – скачок, промежутки) между значением внутрикласового разброса и средними значениями этой величины, которые получают при кластеризации унифицированных выборок, сгенерированных при помощи метода Монте-Карло на основе стандартного распределения:

$$I_{GAP}(\tilde{M}) = \frac{1}{B} \sum_{k=1}^B \log E_w^*(X^N, Q_{\tilde{M}}) - \log E_w(X^N, Q_{\tilde{M}}), \quad (6)$$

Особенностью данного критерия является возможность расчета даже в случае одного кластера.

– Индекс оценки силуэта [6].

Для каждого образа вычисляется величина его «силуэта» $r_{i,k}$, определяющая его схожесть с образами своего кластера и отличие от образов других кластеров:

$$r_{i,k} = (b_{i,k} - a_{i,k}) / \max(b_{i,k}, a_{i,k}), \quad (7)$$

$$a_{i,k} = \frac{1}{N_i - 1} \sum_{s=1}^{N_i} d(x^{(k,i)}, x^{(k,s)}), \quad (8)$$

$$b_{i,k} = \min_{j \neq i} \left[\frac{1}{N_j} \sum_{t=1}^{N_j} d(x^{(k,i)}, x^{(t,j)}) \right], \quad (9)$$

где $a_{i,k}$ определяет среднее расстояние от образа до образов своего кластера, $b_{i,k}$ – минимальное среднее расстояние от образа до образов других кластеров.

В итоге индекс может быть вычислен по формуле:

$$I_{SV} = \frac{1}{N} \sum_{i=1}^{\tilde{M}} \sum_{k=1}^{N_i} r_{i,k} = (b_{i,k} - a_{i,k}) / \max(b_{i,k}, a_{i,k}), \quad (10)$$

2. Сравнение критериев

Для проведения сравнения в среде разработки MATLAB проводится генерация случайных тестовых выборок с размером признакового пространства $n = 2$. Каждая выборка содержит от 2 до 7 кластеров, степень пересечения между которыми задается с помощью параметра dm (чем меньше значение параметра, тем ближе расположены центры кластеров и больше вероятность ошибки распознавания).

Для каждой выборки происходит расчет оптимального числа кластеров в соответствии с одним из используемых критериев, затем полученный результат сравнивается с изначальным числом кластеров. В итоге формируется суммарная ошибка E и исследуется ее зависимость от величины dm .

На рис. 1 представлены результаты анализа критериев оценки качества кластеризации (отражена зависимость величины ошибки E от степени пересечения кластеров dm). При этом сама кластеризация проводится с помощью алгоритма K средних.

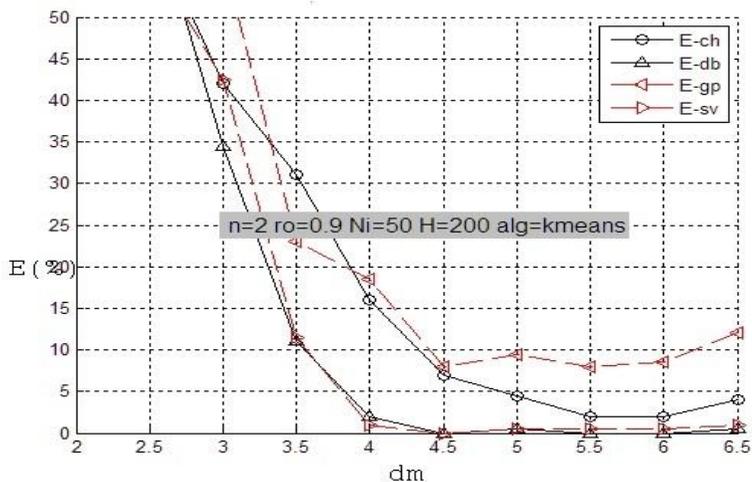


Рис. 1. Зависимость частоты возникновения ошибки от степени пересечения кластеров при применении алгоритма K средних

3. Анализ исходных данных

В данной работе проводится исследование спектров пациентов с подозрением на лейкоплакию.

Исходные данные заданы в виде матрицы 3645×60 , где $n = 3645$ – размерность признакового пространства (количество спектральных характеристик, представляющих одного пациента), а $m = 60$ – число

объектов, составляющих обучающую выборку (количество пациентов, прошедших исследование). При этом у первых 30 пациентов подозревается лейкоплакия, остальные предположительно здоровы.

В таблице приводится фрагмент исходных данных, используемых для анализа в рамках данной работы.

Таблица

Пример обучающих данных (фрагмент выборки)

n \ m	1	2	...	60
1	43,52	-11,77		-19,96
2	35,33	10,75		10,75
...				
3645	-155,13	-85,5		-112,12

4. Проведение кластеризации и визуализация данных

В результате проведения кластеризации на основе выбранного алгоритма и критерия оценки качества кластеризации в обучающей выборке было выделено 3 кластера, при этом все предположительно здоровые пациенты были строго выделены в единую группу, а две другие группы представили пациентов с подозрением на лейкоплакию.

На рис. 2 представлено визуальное отображение полученного разбиения, которое было сформировано с помощью метода главных компонент с числом главных компонент $N_{pc} = 2$.

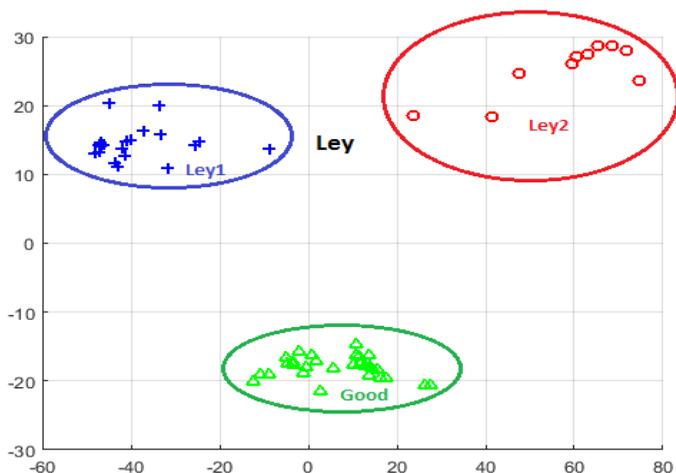


Рис. 2. Визуальное отображение кластеризованных данных, полученное с помощью метода главных компонент

Таким образом, в результате проведенной обработки данных было доказано, что в данном случае двух кластеров (Leu и Good) недостаточно для описания всех возможных состояний пациентов, необходимо выделять три кластера (Leu1, Leu2, Good). Это может сыграть ключевую роль при постановке точного диагноза, выборе методов и способов лечения и назначении специальных средств и препаратов.

Заключение

Главный вывод, который может быть извлечен в результате выполнения данной работы, можно сформулировать так: рассматривая различные алгоритмы и методы анализа данных, необходимо четко осознавать, на основании каких характеристик должно проводиться их сравнение. К таким характеристикам могут относиться точность алгоритма анализа, уровень ошибок, допускаемых алгоритмом, скорость работы алгоритма, его вычислительная сложность, устойчивость по отношению к аномалиям, которые могут содержаться в обучающих данных, и многие другие. В случае если оценка этих критериев является трудно осуществимой или ее недостаточно для выбора нужного способа обработки информации, задача перестает быть тривиальной и ответственность за ее решение возлагается на разработчика или исследователя. Помочь в ее решении может проведение многофакторных модельных экспериментов с применением современных инструментальных средств компьютерного моделирования [7].

Список литературы

1. Сокэл Р.Р. Нумерическая таксономия: методы и современное развитие / Р.Р. Сокэл // Журн. общ. биол. – 1968. – Т. 29. – №. 3. – С. 297-315.
2. Caliński T. A dendrite method for cluster analysis / T. Caliński, J.A. Harabasz // Communications in Statistics-theory and Methods. – 1974. – Т. 3. – №. 1. – С. 1-27.
3. Krzanowski W.J. A criterion for determining the number of groups in a data set using sum-of-squares clustering / W.J. Krzanowski, Y.T. Lai // Biometrics. – 1988. – С. 23-34.
4. Davies D.L. A cluster separation measure / D.L. Davies, D.W. Bouldin // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1979. – №. 2. – С. 224-227.
5. Tibshirani R. Estimating the number of clusters in a data set via the gap statistic / R. Tibshirani, G. Walther, T. Hastie // Journal of the Royal Statistical Society: Series B (Statistical Methodology). – 2001. – Т. 63. – №. 2. – С. 411-423.

6. Kaufman L. Finding groups in data: an introduction to cluster analysis / L. Kaufman, P.J. Rousseeuw – John Wiley & Sons, 2009. – 344 p.

7. Yeung K.Y. Model-based clustering and data transformations for gene expression data / K.Y. Yeung et al. // Bioinformatics. – 2001. – Т. 17. – № 10. – С. 977-987.

МЕТОДИКА КОМПЬЮТЕРНОГО АНАЛИЗА ПРОИЗВОДСТВЕННЫХ БИЗНЕС-ПРОЦЕССОВ НА ПРИМЕРЕ СКЛАДА ПРОИЗВОДСТВЕННЫХ РЕСУРСОВ

В.А. Назарьева

Студент

М.Г. Матвеев

Профессор

Введение

Важным звеном логистической цепочки для производственных предприятий, дистрибьюторских компаний и розничных сетей является склад распределения, эффективность функционирования которого в конечном итоге существенно влияет на эффективность бизнеса в целом.

Для организации эффективной работы склада можно использовать современные компьютерные методы. При этом наиболее подходящим методом является имитационное моделирование. В индустриально развитых странах имитационное моделирование довольно часто используется в проектах, направленных на создание новых или реконструкцию существующих логистических систем [1]. На данный момент в России такое моделирование является инновационным подходом к организации работы логистических систем.

Цель исследования в данной статье: построение модели складской системы и проведение имитационного моделирования и функционально-стоимостного анализа с целью повышения эффективности функционирования моделируемой системы.

Для достижения цели необходимо выполнить следующие задачи:

- провести описание и построить модель процесса работы склада для проведения имитации;
- задать параметры модели;
- выполнить имитацию;
- провести ФСА результатов моделирования;
- провести оптимизацию модели.

© Назарьева В.А., Матвеев М.Г., 2017

1. Описание процесса

Графически структуру склада можно представить с помощью топологической схемы, состоящей из множества логистических территорий, называемых зонами (рис. 1): «Зона приемки», «Зона разгрузки транспортного средства», «Зона хранения», «Зона комплектации» и «Зона отгрузки материала» [2].

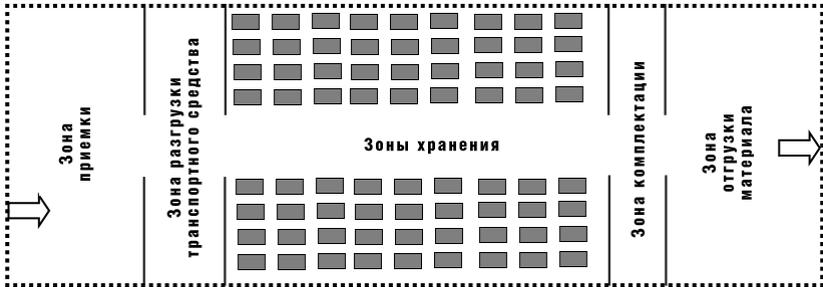


Рис. 1. Графическая модель склада

Для формального представления модели бизнес-процессов склада была использована методология функционального моделирования IDEF0. В основу построения модели легли три базовые функции: приемка и разгрузка товара, хранение, комплектация и отгрузка товара в производство (рис. 2), их дальнейшая декомпозиция производилась на диаграмме EPC, на которой была проведена имитация.

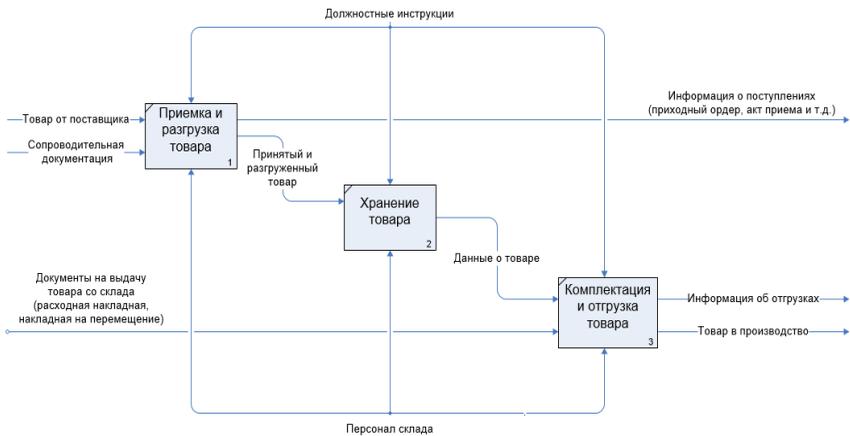


Рис. 2. Основные бизнес-процессы склада

На склад производственного предприятия по согласованному с поставщиком календарю поставок прибывают машины с товаром одного типа. Машины с грузом на склад прибывают не в определенное время, а в случайные моменты времени, то есть интервал между прибытием машин является случайной величиной. Количество груза, которое находится в машине – тоже величина случайная, следовательно, время разгрузки товара для каждой прибывшей на склад машины является случайным. Товар поступает, хранится и передается в производство в упаковках, количество к отгрузке указано в накладной.

На рассматриваемом складе распределения было принято решение о необходимости проведения исследования, направленного на организацию эффективной работы склада. Поставленную задачу возможно решить с помощью метода имитационного моделирования и оптимизации.

Основные задачи моделирования:

- увеличить кол-во обслуживаемых машин с товаром;
- сократить затраты на обслуживание склада;
- сократить очереди в пунктах обработки материального потока на складе;
- сократить дефицит, связанный с недостатком товаров в отдельные моменты времени.

2. Параметры модели

Предположим, что на рассматриваемом в качестве примера складе предусмотрено 2 поста для разгрузки, 3 места для парковки, работают 4 грузчика, один кладовщик. Для упрощения в модели точно определены вместимость склада – 300 складских мест, и точка перезаказа – 120 единиц товара.

Имитационную модель склада можно представить в следующем обобщенном виде (рис. 3), в котором множество входных параметров модели после проведения имитации преобразуется во множество выходных показателей функционирования моделируемой системы [3].

3. ФСА результатов моделирования

После построения модели было проведено имитационное моделирование, и в результате найдены временные и стоимостные показатели работы склада.

Результаты моделирования:

- из 13 машин, прибывших на склад, было обслужено всего 8;
- товар, привезенный только 7-ю машинами был размещен на хранение на складе;
- из 13 машин 7 ожидали на парковке (среднее время ожидания 42 минуты);

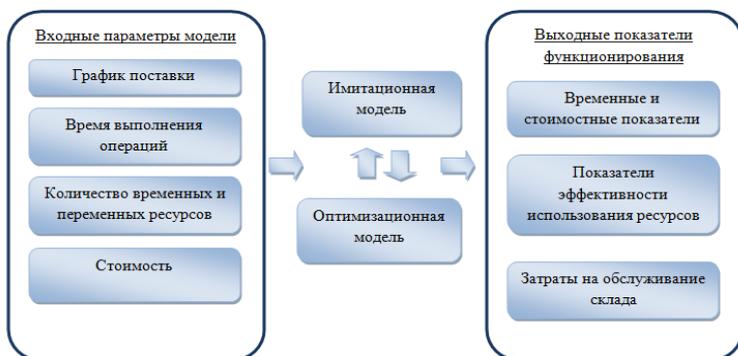


Рис 3. Имитационная модель склада

- поступило 7 накладных на отгрузку, из них товары по 6 накладным были отгружены в производство, а по одной накладной возник дефицит, вызванный нехваткой товаров на складе из-за медленного обслуживания приезжающих машин;
- всего от поставщика поступило 489 единиц товара, запас на складе в конце дня составляет 218, в производство было отгружено 271 единица товара (рис. 4);
- затраты на з/п персонала за день составили 2 467 руб.

Переменная	Значение
<input checked="" type="checkbox"/> Переменные глобальных объектов	
Место разгрузки.Количество	0
Товар.Поставка	0
Место парковки.Количество	2
Товар.Запас	218
Поступило.Количество	489
Склад.Вместимость	300
Склад.ТочкаПерезаказа	120

Рис. 4. Результаты моделирования

Таким образом, в конце рабочего дня 4 машины остались не обслуженными, из них 2 машины ожидают на парковке и 2 машины ожидают обслуживания в зоне приемки-разгрузки.

Проведя функционально-стоимостной анализ результатов имитации, можно сделать следующий вывод: существующей пропускной способности склада недостаточно для обслуживания текущего грузопотока. Необходимо провести оптимизацию модели, которая позволит улучшить показатели работы склада.

4. Оптимизация модели

Алгоритм взаимодействия оптимизационной и имитационной моделей представлен на рис. 5. Процесс оптимизации включает в себя следующие шаги: задаем входные параметры (вектор X) и выполняем имитацию модели, в результате получаем вектор выходных показателей модели $Y(X)$, далее рассчитываем значения целевой функции (величина $F(Y(X))$), если необходимо улучшить значение целевой функции, то изменяем значения входных параметров и заново выполняем имитацию модели. Эти шаги повторяются до тех пор, пока целевая функция не достигнет своего максимума [4].

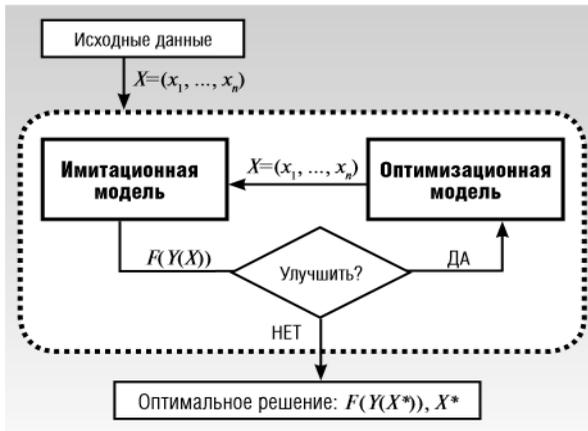


Рис. 5. Алгоритм взаимодействия имитационной и оптимизационной модели

В рассматриваемой задаче ключевой показатель эффективности работы склада (целевая функция) – это прибыль от реализации товаров (Π). Прибыль вычисляется как разница между выручкой от реализации отгруженной продукции (B) и затратами на производство и реализацию продукции ($З$). Таким образом прибыль растет, когда затраты уменьшаются.

$$\text{Прибыль } (\Pi) = \text{Выручка}(B) - \text{Затраты } (З)$$

В состав общих затрат по созданию и поддержанию запасов входят:

- стоимость закупки материала, включая доставку материала;
- штрафы за простой машин с товаром от поставщика;
- общая стоимость операций по приемке, разгрузке, размещению и отгрузке товаров на складе, которая включает в себя заработную плату сотрудников склада;

- затраты на хранение.

После выполнения оптимизации модели показатели работы склада значительно улучшились:

- все прибывшие на склад машины были разгружены;
- весь товар, привезенный машинами был размещен на складе;
- снизилось время ожидания машины на парковке до 25 минут;
- увеличилось количество товаров, отгруженных в производство.

В таблице показано, как изменялось значение KPI в ходе выполнения оптимизации модели, длительность моделируемого процесса один день.

Таблица

Значение KPI

№	Прибыль
1	14300
2	18200
3	28000

Заключение

В ходе выполнения исследования была построена имитационная модель склада производственных ресурсов в программной среде Business Studio, позволяющая находить наилучший режим работы склада.

Таким образом, имитационное моделирование и функционально-стоимостной анализ позволили оптимизировать бизнес-процессы предприятия, не проводя эксперименты над реальной компанией и сотрудниками [5].

Список литературы

1. Сидоренко, В. Н. Имитационное моделирование в науке и бизнесе: подходы, инструменты, применение / В. Н. Сидоренко, А. В. Красносельский // Бизнес-информатика. – 2008. – № 2. – С. 52-57.
2. Гаджинский, А.М. Современный склад. Организация, технологии, управление и логистика: учеб.-практ. пособие / А.М. Гаджинский. – Москва: Проспект, 2005. – 176 с.
3. Чеботарев, В.Г. Анализ и моделирование бизнес-процессов / В.Г. Чеботарев. – Москва : Логика бизнеса IDS Scheer Group, 2005. – 245 с.
4. Бабина, О.И. Имитационная модель склада промышленного предприятия по производству бетона / О.И. Бабина // Бизнес-информатика. – 2015. – № 1. – С. 41-50.
5. Имитационное моделирование и функционально-стоимостной анализ. Методика / Группа компаний «Современные технологии управления». – Самара : Альфа-Пресс, 2010. – 81 с.

СРАВНЕНИЕ МЕТОДОВ ОБНАРУЖЕНИЯ СКРЫТЫХ ТЕМ В ПОТОКЕ ТВИТОВ ИЗ СОЦИАЛЬНОЙ СЕТИ TWITTER

М.А. Напрасникова

Студент

А.В. Сычев

Доцент

Введение

В настоящее время наблюдается стремительный рост популярности социальных сетей, и Twitter не является исключением. Миллионы людей по всему миру пишут в Twitter сообщения о том, что происходит вокруг. Twitter становится ведущим источником для исследования появляющихся событий, последних новостей и общих тем, которые важны для широкой аудитории Twitter. Но чтобы получить ответ на свой запрос в Twitter, пользователь сталкивается с огромным количеством нерелевантной информации. Рассмотрим методы интеллектуального анализа данных, применяемые для легкого доступа к ключевой информации, скрытой в массиве твитов из социальной сети Twitter.

1. Проблемы, возникающие при исследовании данных, полученных из Twitter

При работе с данными, полученными из Twitter, существуют следующие проблемы:

- Ограниченная длина твита. На данный момент объем символов в твите равен 140, не считая URL.
- Огромный объем и скорость появления сообщений.
- Высокая зашумленность данных. Т.к. пользователи Twitter часто пишут сообщения на ходу, то в них есть масса опечаток, и, конечно, очень много сообщений являются просто спамом.
- «Взрывная» природа твитов. Т.е. если в мире происходит какое-нибудь важное событие, то вместе с ним следует огромная волна твитов от пользователей, обсуждающих это событие.

2. Видоизмененная скрытая марковская модель

Один из подходов анализа данных, полученных из Twitter, основывается на применении скрытой марковской модели для сегментирования события на подсобытия, для чего вводятся дополнительные вероятности для выделения подсобытий в событиях [1]. Скрытая марковская модель может сегментировать события по ходу их

появления в зависимости от распределения слов, а так же с учетом внезапности их появления. Каждый полученный сегмент представляет подсобытие – семантически отдельную часть всего события. Затем выбираются ключевые сообщения в каждом сегменте и на их основе составляются резюме события.

3. Вейвлет-анализ сигналов-хештэгов

Идея обнаружения событий с использованием вейвлет-анализа заключается в том, что из потока твитов выбираются хештеги и время появления каждого твита. Затем для каждого момента времени подсчитывается количество появлений заданного хештега и строится частотно-временная характеристика сигнала, используя непрерывное Вейвлет-преобразование [2]. Процесс извлечения скрытых тем делается для каждого построенного хештег-сигнала, путем нахождения локальных максимумов. Последующее применение LDA-модели улучшает резюмирование событий. Каждый временной интервал рассматривается как документ для обучения LDA модели.

4. Метод на основе графа для резюмирования событий в Twitter

Данный метод основан на построении графа из супер-твитов, в которых хранится обработанная информация. Супер-твит представляет собой кортеж, включающий в себя текст твита, количество дубликатов данного твита, список авторов, размещавших данный твит и время поста главного первоначального твита. Такой подход позволяет уменьшить объем для хранения входных данных. С помощью построенного графа выполняется идентификация событий, важных фаз событий, нахождение тем подсобытий, учитывая социальный аспект. Фазы события моделируются как сообщества в графе схожести (similarity graph), резюмирование подсобытий происходит путем аппроксимации минимально-взвешенного доминирующего множества.

5. Подход для нахождение скрытых тем в потоке твитов из социальной сети Twitter, использующий LDA модель.

Данный метод основан на построении LDA модели. Данная модель принимает на вход массив документов, а на выходе предоставляет множество скрытых тем и их привязку к каждому документу [3].

Рассматриваемая модель описывается следующим образом.

Модель обрабатывает данные в пакетном режиме и периодически обновляется и при этом не разрастается в размерах со временем, т.е. она остается чувствительной к изменению темы с течением времени. Темы представляются в виде полиномиального распределения слов, что является более описательным, чем метод представления с помощью хештегов или строкой. Время в модели разделено на отсчеты, в соответствии с которыми

все документы также разбиваются на группы. Для использования наработок предыдущей модели вводим коэффициент вклада. Он определяет степень влияния параметров предыдущей модели на параметры распределения Дирихле в текущей LDA модели. Таким образом, модель, с приходом новых документов, генерирует набор постоянно развивающихся тем. Ключевым параметром в модели является количество тем, который влияет на их подробность. Большое значение количества тем позволяет получать специализированные темы, а малое – более общие темы.

6. Сравнительная таблица методов

В приведенной ниже таблице перечислены преимущества и недостатки рассмотренных методов обнаружения скрытых тем.

Таблица

Сравнительная таблица методов

Метод	Преимущества	Недостатки
SummHMM	Значение переменной состояния может стать частью одной из существующих цепей, а может сформировать новую цепь, независимую от других цепей. Данная особенность может быть очень полезной для выделения подсобытий, поскольку они могут быть разбросаны во времени.	Метод не учитывает взаимосвязи между словами. Подходит только для структурированных событий. Модель, построенная по одному типу события, не сработает эффективно на события другого типа.
Вейвлет-анализ сигналов-хештэгов	Метод позволяет обрабатывать данные в режиме реального времени, т.к. учитывает время и частоту появления твитов, может идентифицировать всплески упоминаний хештэгов, относящихся к изолированным событиям или событиям, повторяющимся в течение времени.	Метод применяется для обнаружения событий, которые происходили в течение нескольких дней, а идентифицировать последовательные события, происходившие в течение одного дня, не удается.

Метод	Преимущества	Недостатки
Метод на основе графа для резюмирования событий в Twitter	Метод учитывает социальные аспекты Twitter. Создает линию сюжета развития события, учитывая взаимосвязь между подсобытиями.	Метод не подходит для потокового анализа данных. Отсутствует моделирование временной динамики контента.
LDA модель	Модель использует динамический словарь, эффективно обрабатывает потоковые данные, не разрастаясь в размерах со временем, и может быть применена к событиям любого типа.	Возможны проблемы с масштабируемостью метода

Заключение

В ходе данной работы были рассмотрены методы интеллектуального анализа данных, применяемые для обнаружения скрытых тем в массиве данных, полученных из социальной сети. В итоге был выбран алгоритм, основанный на LDA модели, т.к. он работает с динамическим словарем, способен обрабатывать твиты в режиме реального времени, учитывает результаты предыдущего шага работы и может быть применен к нахождению событий любого типа.

Список литературы

1. Nichols J. Summarizing Sporting Events Using Twitter / J. Nichols, J. Mahmud, C. Drews // ACM international conference on Intelligent User Interfaces – 2012. – P. 189-198.
2. Babcock B. Models and issues in data stream systems / B. Babcock, S. Babu, M. Datar // ACM Symposium on Principles of Database System. – 2002. – P. 1-16.
3. Canini K.R. Online inference of topics with latent dirichlet allocation / K.R. Canini, L. Shi, T.L. Griffiths // Journal of Machine Learning Research. – 2009. – P. 65-72.

МЕХАНИЗМ ОБМЕНА ДАННЫМИ МЕЖДУ ТИПОВЫМИ КОНФИГУРАЦИЯМИ 1С И МОБИЛЬНЫМ ПРИЛОЖЕНИЕМ 1С ПОД ANDROID

В. В. Нижневский

Студент

А. И. Другалев

Ассистент

Введение

Современное общество пришло к тому, что передача всех функций, связанных с производством какого-то материального блага или услуги, к одному человеку оказалась невозможной. Попросту говоря, мы живем в мире, где для производства гвоздя необходимы усилия сотен людей: тех, кто добывает руду, тех, кто выплавляет из неё металл, а из металла делает гвозди, тех, кто транспортирует и продаёт их. И, конечно, тех, кто руководит всеми этими процессами.

Актуальность решения задачи оперативного обмена информацией между руководителем и сотрудниками обусловлена тем, что в современном мире почти у всех есть смартфоны, позволяющие устанавливать различные приложения и постоянно поддерживать связь с начальством и коллегами.

1. Анализ задачи

Наше время – самый ценный ресурс, которым мы обладаем. Потерянные деньги можно заработать, но потеряно время потеряно навсегда.

Рассмотрим методики планирования времени двух авторов: Глеба Архангельского и Стивена Кови.

Глеб Архангельский предлагает делить задачи на жесткие и гибкие. Жесткие задачи, например, встречи, привязаны к конкретному времени, а гибкие – нет.

Также автор выделяет несколько приемов выделения отдельных дел:

Выделять главное дело дня, способствующее продвижению к главным целям.

Выполнять часть большой и сложной задачи, требующей большого количества времени для реализации – «съесть кусочек слона». Например, написать главу дипломной работы.

Выполнять одно неприятное дело, которое постоянно откладывается, но рано или поздно его необходимо сделать – «съесть лягушку». Например, купить и вкрутить лампочку.

Получить приятную награду за достижения успеха по делам из пунктов 2 и 3. Например, чтение книги, просмотр фильма, досуг с близкими или друзьями.

Фиксировать в конце каждого дня главное событие дня для определения личных ценностей [1].

Методика Стивена Кови позволяет выявлять в потоке ежедневных дел наиболее важные и правильно расставлять приоритеты. Автор предлагает распределять все дела и обязанности по четырем квадрантам (рис. 1).

	Срочные	Не срочные
Важные	<p>I Квадрант «Необходимость»</p>	<p>II Квадрант «Эффективность и баланс»</p>
Неважные	<p>III Квадрант «Разочарования»</p>	<p>IV Квадрант «Расточительство и излишества»</p>

Рис. 1. Методика Стивена Кови

Квадрант I – «важное и срочное». В него помещаются дела, которые необходимо решить «здесь и сейчас». Фактически, это ситуация аврала. Держать этот квадрант пустым получится не всегда, однако к этому необходимо стремиться.

Квадрант II – «важное, но несрочное». В идеале, именно в нем должна быть сосредоточена основная масса текущих дел. Важность показывает, что выполнение этих обязанностей приближает вас к достижению главных целей, а отсутствие срочности свидетельствует о том, что вы не допустили аврала. Этот квадрант необходимо постоянно разгружать, иначе дела из него будут переходить в квадрант I.

Квадрант III – «неважное, но срочное». Такие дела отвлекают от движения к поставленным целям, но требуют срочного внимания. Это могут быть личные поручения, которые далеко не всегда можно отбросить.

Квадрант IV – «неважное и несрочное». Необходимо тщательно фильтровать дела из этого квадранта. Некоторые из них могут принести пользу в будущем, но некоторые могут просто отнять много времени. Если

эти дела связаны с работой, а не с развлечениями или с личной жизнью, возможно, их стоит порекомендовать другим [2].

Видно, что в настоящее время остро стоит вопрос планирования рабочего времени, причем не только планирования времени руководителя, но и времени и сотрудников.

2. Цель работы

Цель работы в рамках данной статьи – реализация механизма обмена данными между типовыми конфигурациями 1С или уникальными решениями и мобильным приложением 1С под Android.

Реализация такого механизма обмена способствует решению более крупной задачи – создание индивидуального модуля «Менеджер задач», позволяющего руководителю назначать своим сотрудникам или самому себе задачи на день или другой период времени, а также получать отчеты от сотрудников о проделанной работе. Возможность пользоваться модулем должна быть как на персональном компьютере, так и на смартфоне.

1. Реализация

Создадим конфигурацию 1С, в свойствах которой в поле «Назначение использования» укажем «Персональный компьютер» и «Мобильное приложение».

Добавим в конфигурацию следующие объекты:

- Справочник «Задачи»;
- План обмена «ПланОбменаМобильный»;
- XDTO-пакет «WebОбмен»;
- Web-сервис «WebОбмен»;
- Общую команду «Обмен» [3].

Так как целью работы, проводимой в рамках написания этой статьи является реализация механизма обмена данными, не будем детализировать реквизиты, табличные части и формы справочника «Задачи». В дальнейшем это можно будет создать для реализации механизма выставления задачи и отчета о проделанной работе. Создадим только реквизит «Наименование».

В состав созданного плана обмена необходимо включить справочник «Задачи».

В свойствах XDTO-пакета «WebОбмен» в поле URI пространства имен укажем адрес web-сервиса, в данном случае «<http://localhost/ws1>».

В web-сервисе «WebОбмен» создадим операцию «ВыполнитьОбмен», в которую добавим параметры «КодУстройства», необходимый для идентификации устройства, с которым будет проходить обмен и «ДанныеУстройства».

Для операции «ВыполнитьОбмен» создадим одноименную процедуру, в которой реализуем получение и выгрузку данных в мобильное устройство.

В общем команде «Обмен» создадим процедуру «ОбработкаКоманды», с помощью которой происходит обмен данных с центральной базой и обработка полученных данных.

Запустим конфигурацию в режиме 1С:Предприятие, а также мобильное приложение 1С.

В приложении на персональном компьютере создадим элемент справочника «Задача 1» (рис. 2), в мобильном приложении – элемент справочника «Задача 2» (рис. 3).

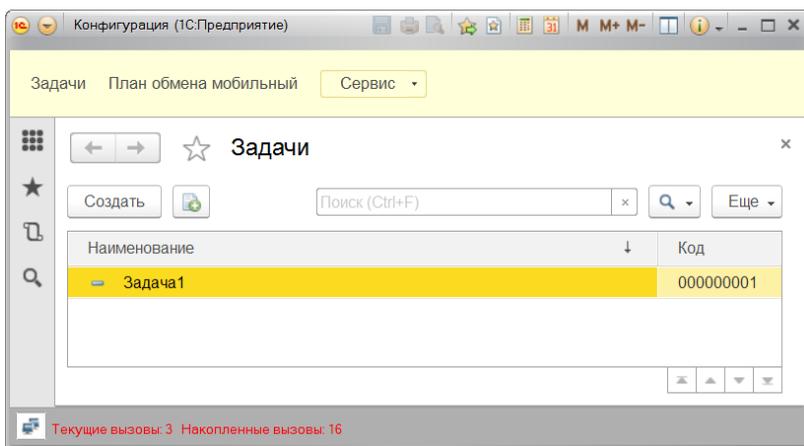


Рис. 2. Созданная задача в приложении для ПК

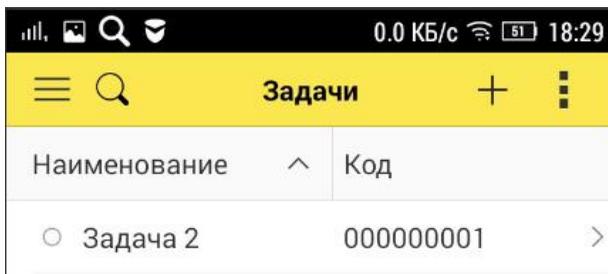


Рис. 3. Созданная задача в мобильном приложении

Затем на мобильном приложении выполним команду «Обмен». После этого произойдет обмен данными между конфигурацией 1С на ПК и

мобильным приложением 1С. В нашем примере элемент «Задача 1» добавится в мобильное приложение, а элемент «Задача 2» – в приложение для ПК. Результат показан на рис. 4 и 5.

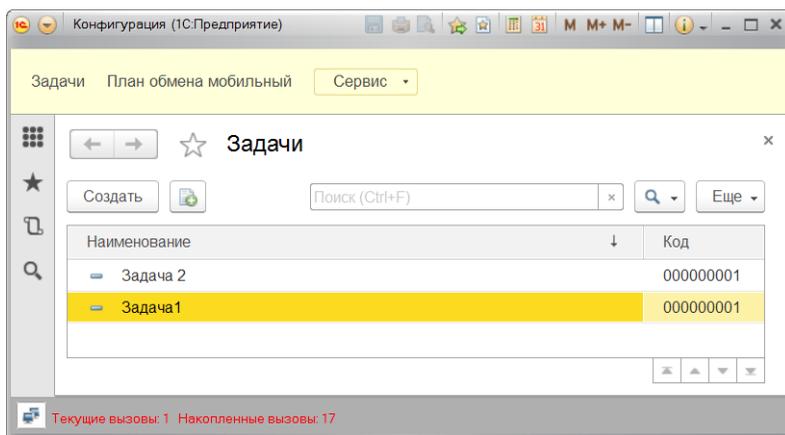


Рис. 4. Результат обмена данными на ПК

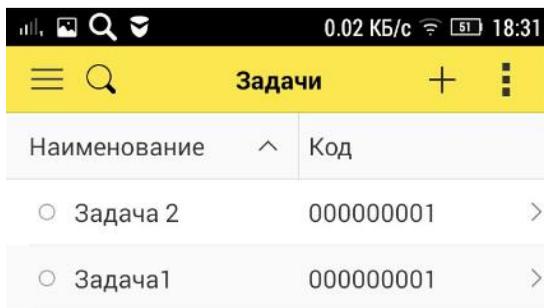


Рис. 5. Результат обмена данными в мобильном приложении

Заключение

Данная статья посвящена реализации механизма обмена данными между типовыми конфигурациями 1С или уникальными решениями и мобильным приложением 1С под Android. Такой механизм необходим для реализации индивидуального модуля «Менеджер задач», позволяющего руководителям выставлять задачи сотрудникам и получать оперативные отчеты о выполнении.

Список литературы

1. Корпоративный тайм-менеджмент : Энциклопедия решений / Г.А. Архангельский. – М. : Альпина Бизнес Букс, 2008. – 160 с.
2. Семь навыков высокоэффективных людей : Мощные инструменты развития личности / Кови Стивен Р.; Пер. с англ., М.: Альпина Бизнес Букс, 2008. – 276 с.
3. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / М. Г. Радченко, Е. Ю. Хрусталева. – М. : 1С-Публишинг, 2013. – 964 с.

РАЗРАБОТКА УЧЕБНОГО КОМПИЛЯТОРА ДЛЯ ПРОЦЕДУРНОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ

В.А. Новиков

Студент

Д.И. Соломатин

Старший преподаватель

Введение

На факультете компьютерных наук ВГУ студентам кафедры программирования и информационных технологий на третьем курсе преподается дисциплина «Теория компиляторов». В рамках изучения данной дисциплины студенты знакомятся с основами синтаксического и семантического анализа кода, оптимизацией и кодогенерацией под некоторые платформы. В качестве практической задачи студенты должны разработать собственный компилятор для некоторого подмножества выбранного языка программирования под выбранную платформу.

Учитывая изучаемые ранее дисциплины, для выполнения данной задачи, как правило, используются следующие инструменты:

- C# в качестве языка реализации компилятора;
- Antlr 3 [1] для построения лексического и синтаксического анализаторов;
- .NET Framework в качестве целевой платформы компилятора (кодогенерация в Microsoft IL).

Для простоты освоения курса необходим пример готового компилятора, построенного с помощью перечисленных выше инструментов, который был бы достаточно прост для изучения студентами, т. е. обладал достаточно простой и прозрачной архитектурой,

но при этом демонстрировал бы основные приемы построения компилятора для полноценного языка программирования.

Примеров компиляторов, удовлетворяющих вышеописанным требованиям, найдено не было, что привело к появлению данной работы.

1. Описание учебного языка

В качестве основы входного языка для разработанного компилятора был выбран язык PL/SQL [2].

Входной язык разработанного учебного компилятора поддерживает следующие типы данных:

- числовые типы: int, char, double, float;
- строковый тип string;
- логический тип bool;
- контейнеры: массивы, списки, ассоциативные массивы.

Поддерживаются основные операторы и операции:

- условный оператор;
- циклы: for, while, do-while

Помимо встроенных типов данных, допускается определение пользовательских классов.

Учебный компилятор позволяет использовать не только базовые или определенные в программе типы, но и типы из динамически подключаемых библиотек. Написана стандартная библиотека языка, содержащая все основные математические функции, классы консольного и файлового ввода-вывода.

2. Архитектура компилятора

Учебный компилятор состоит из следующих модулей:

- лексический и синтаксический анализатор;
- семантический анализатор;
- модуль оптимизации;
- модуль кодогенерации.

На рис. 1 продемонстрирован процесс компиляции исходного кода.

2.1. Лексический и синтаксический анализ

В учебном компиляторе лексический и синтаксический анализ реализован при помощи библиотеки Antlr 3, позволяющей генерировать классы анализаторов из формально описанной грамматики языка. В результате работы данных анализаторов из исходного кода программы формируется абстрактное синтаксическое дерево. Особенностью генерируемого абстрактного синтаксического дерева является типизация его узлов: различные узлы представлены экземплярами различных классов.

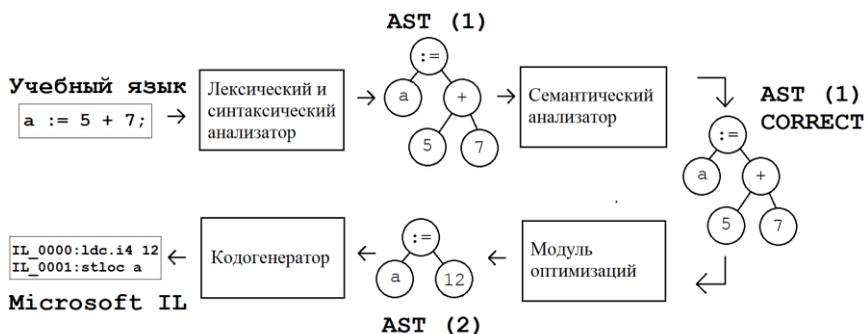


Рис. 1. Процесс компиляции исходного кода

Пример абстрактного типизированного синтаксического дерева представлен на рис. 2.

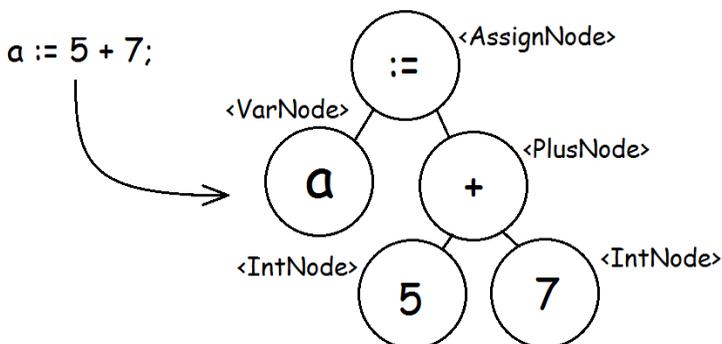


Рис. 2. Абстрактное синтаксическое дерево (типизированное)

2.2. Семантический анализ

В процессе семантического анализа выясняется смысловая корректность исходного кода. Помимо этого, на данном этапе собирается информация об объявленных в программе типах – для каждого из них создается объект-описатель, содержащий информацию о процедурах, функциях и полях, объявленных в типе.

Семантический анализ осуществляется в процессе обхода абстрактного синтаксического дерева. Обход дерева реализуется при помощи модифицированного паттерна «Посетитель» [3]. Для каждого узла определен метод *Visit*, аргументом принимающий объект типа данного узла и контекст семантического анализа, в котором содержится информация о определенных типах и переменных, доступных в данной

области видимости. На листинге 1 представлен пример посещающего метода для узла операции отрицания.

Листинг 1

Метод анализа узла NotNode в семантическом анализаторе

```
private TypeDescriptor Visit(
    NotNode node,
    Context context) {
    var operand = Visit(node.Operand as dynamic, context);
    if (!IsBoolType(operand.Type)) {
        Log(
            String.Format(
                "Оператор {0} не может применяться для типа {1}",
                node.Text, operand.Type
            ),
            node
        );
        return TypeDescriptor.Undefined;
    }
    return operand;
}
```

Методы обхода возвращают объект типа *TypeDescriptor*, который описывает тип поддерева, представленный проанализированным узлом.

2.3. Оптимизации

Оптимизации в учебном компиляторе реализованы как эквивалентные преобразования абстрактного синтаксического дерева. Реализованы следующие простейшие оптимизации:

- свертка констант;
- распространение констант.

В результате свертки констант те выражения, которые могут быть вычислены на этапе компиляции, вычисляются, и узлы-операции заменяются на вычисленное значение. Процесс свертки констант продемонстрирован на рис. 3.

Алгоритм распространения констант является расширением свертки констант. В данной оптимизации учитываются и неконстантные переменные, которые, однако, имеют определенное на момент компиляции значение, и могут быть заменены на константу.

Алгоритм оптимизации заключается в обходе дерева, поиске константных выражений, их вычислении и замене на узлы константы. Обход дерева также осуществляется при помощи модифицированного паттерна «Посетитель».

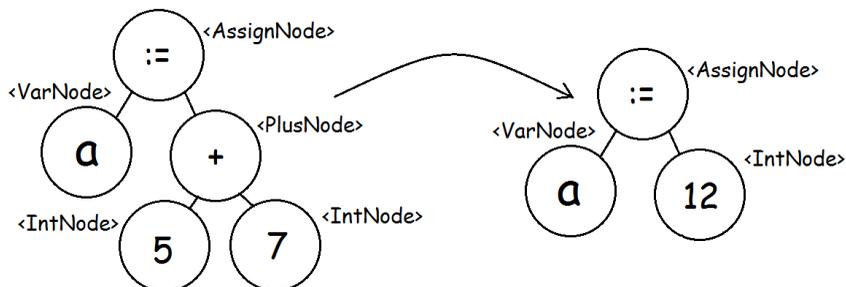


Рис. 3. Свертка констант

2.4. Кодогенерация

В процессе кодогенерации из абстрактного синтаксического дерева формируется код целевой платформы.

Для каждого структурного элемента программы определен отдельный класс, генерирующий соответствующий фрагмент кода целевой платформы. Данные классы содержат шаблон для генерации кода, а также методы, порождающие объекты-строители для вложенных элементов. В процессе генерации кода объект-строитель запросит сгенерированный код у его вложенных объектов-строителей и подставит полученный результат в шаблон.

Отдельно выделим класс *CommandBuilder*, который ответственен за генерацию кода операций. В данном классе определены методы, соответствующие основным операциям, свойственным стековым машинам. При вызове данных методов класс *CommandBuilder* генерирует соответствующий MSIL-код [4].

Процесс кодогенерации заключается в обходе абстрактного синтаксического дерева и вызове соответствующих методов на объектах-строителях кода. Обход, как и ранее, осуществляется при помощи модифицированного паттерна «Посетитель». Диаграмма классов генераторов кода продемонстрирована на рис. 4.

Далее представлен пример посещающего метода для узла сложения (листинг 2) и метода обработки бинарных операций (листинг 3).

Листинг 2

Метод обработки узла сложения в кодогенераторе

```
private TypeInfo Visit(
    PlusNode node,
    MethodBuilder builder,
    CodegenContext ctx) {
    return BinaryOperatorVisit(node, builder,
        (x) => x.Plus(), ctx);
}
```

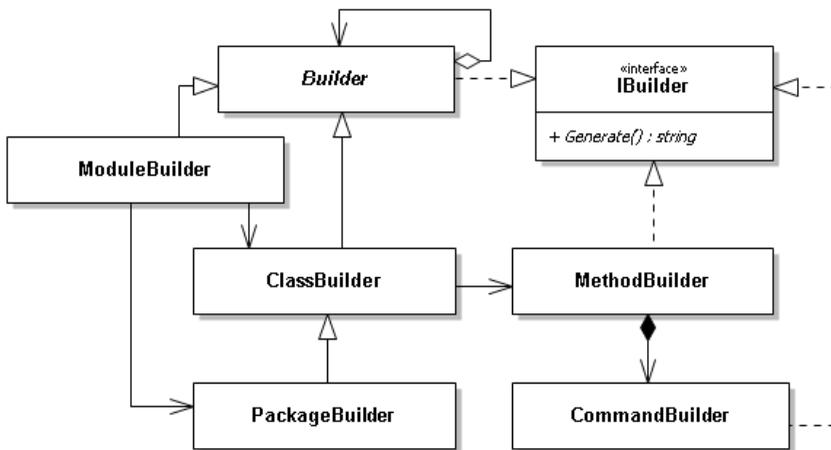


Рис. 4. Диаграмма классов-генераторов кода

Листинг 3

Метод обработки бинарных операций в кодогенераторе

```

private TypeInfo BinaryOperatorVisit(
    BinaryOperator binaryOperator,
    MethodBuilder builder,
    Action<MethodBuilder> action,
    CodegenContext context) {
    Visit(
        (dynamic) binaryOperator.LeftOperand,
        builder, context
    );
    var res = Visit(
        (dynamic) binaryOperator.RightOperand,
        builder, context
    );
    action(builder);
    return res;
}

```

3. Пример программы на учебном языке

На листинге 4 представлен пример быстрой сортировки, реализованный на учебном языке.

Также для демонстрации возможностей компилятора на входном языке компилятора была написана простейшая реализация нейронной сети, на основе которой построен пример распознавания рукописные цифры из

набора MNIST Image Dataset. Полученный пример работает корректно, что косвенно подтверждает корректность реализации компилятора.

Листинг 4

Быстрая сортировка на входном языке разработанного компилятора

```
create package body Sort is
  procedure QuickSort(
    arr array(int),
    first int, last int
  )
  b int; e int;
  opor int; tmp int;
begin
  b := first; e := last; opor := arr[(b + e) / 2];
  while b <= e loop
    while arr[b] < opor loop
      b := b + 1;
    end loop;
    while arr[e] > opor loop
      e := e - 1;
    end loop;
    if b <= e then
      tmp := arr[b]; arr[b] := arr[e]; arr[e] := tmp;
      b := b + 1; e := e - 1;
    end if;
  end loop;
  if e > first then
    Sort.QuickSort(arr, first, e);
  end if;
  if b < last then
    Sort.QuickSort(arr, b, last);
  end if;
end;
end;
```

Заключение

В рамках данной работы был разработан учебный компилятор, демонстрирующий основные принципы лексического, синтаксического и семантического анализов кода, его оптимизации и кодогенерации для виртуальной машины Microsoft CLR. Компилятор написан на языке C# и обладает достаточно простой для понимания архитектурой. Данный компилятор может быть использован студентами как пример при выполнении практического задания в процессе изучения курса «Теория компиляторов».

Список литературы

1. Terence, P. The Definitive ANTLR Reference / P. Terence. – Dallas : Pragmatic Bookshelf, 2007. – 376 p.
2. Database PL/SQL Language Reference [Электронный ресурс]. – Режим доступа : <https://docs.oracle.com/cloud/latest/db112/LNPLS/toc.htm>
3. Shawn, H. Visitor and multiple dispatch via C# ‘dynamic’ / H. Shawn. [Электронный ресурс]. – Режим доступа : <https://blogs.msdn.microsoft.com/shawnhar/2011/04/05/visitor-and-multiple-dispatch-via-c-dynamic/>
4. Rishikesh, S. Understanding Common Intermediate Language (CIL) / S. Rishikesh. [Электронный ресурс]. – Режим доступа : <https://www.codeproject.com/Articles/362076/Understanding-Common-Intermediate-Language-CIL>

АВТОМАТИЗАЦИЯ ПРОЦЕССА ОБНАРУЖЕНИЯ ВАНДАЛИЗМА В РУССКОЯЗЫЧНОМ РАЗДЕЛЕ ВИКИПЕДИИ

А.А. Ногих

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Википедия – крупнейший в мире проект, основанный на открытом и свободном взаимодействии участников. Являясь пятым по посещаемости сайтом в сети Интернет [1] и не накладывая ограничений на возможность участия, Википедия сильно подвержена вандализму – порядка 7% всех правок являются вандальным [2]. Существует масса публикаций, посвященных автоматизации борьбы с этим явлением в английском разделе Википедии [2, 3]. Однако русскоязычный раздел (на момент 2017 года являющийся 7-м по величине) остался в стороне от такого внимания со стороны исследователей. Отсутствуют и размеченные вручную выборки правок, опираясь на которые можно проводить подобные исследования. Данная статья имеет своей целью восполнить данный пробел и проверить возможности автоматизации борьбы с вандализмом в русскоязычном разделе Википедии.

1. Значимые особенности Википедии

Структурно Википедия состоит из страниц, расположенных в различных пространствах имен (namespaces). Существуют пространства имен статей, шаблонов, категорий, личных страниц участников и т.д.

После каждого совершенного изменения статья приобретает новый вид, называемый версией (ревизией) статьи. Для каждой статьи хранится список всех состояний, через которые она прошла, включая [4]:

- дату и время изменений;
- авторов изменений (как анонимных, так и зарегистрированных);
- комментарии к изменениям;
- различные флаги (например, флаг небольших изменений);
- полный текст содержимого статьи после каждого изменения и т.д.

По результатам публичного обсуждения, участникам проекта могут быть предоставлены полномочия по администрированию. Формально, в сообществе определены группы пользователей, за каждой из которых закреплены определенные привилегии.

В отличие от англоязычного раздела, в разделе Википедии на русском языке введен механизм патрулирования статей:

Патрулированные версии – версии страниц Википедии с отметкой о том, что они прошли первичную проверку качества, имеют приемлемый вид и не содержат серьезных нарушений правил Википедии.

Проверку статей осуществляют пользователи, имеющие флаг патрулирующего. Дополнительно был введен флаг автопатрулируемого, при котором правки уже отпатрулированных правок сохраняют эту отметку. Данные два флага назначаются добросовестным участникам по результатам открытого обсуждения их кандидатуры сообществом.

Существенным для данной работы также является механизм «отката» (rollback) правок – предназначенный для быстрой отмены последних изменений в статье и нацеленный на борьбу с вандализмом. Чтобы получить возможность использовать «откат», пользователь также должен пройти одобрение сообществом.

2. Постановка задачи

Важным этапом для данной работы является как можно более четкое определение вандальных изменений. В соответствии с правилами и руководствами Википедии [5]:

Вандализм в Википедии – явно вредительское добавление, удаление или изменение содержания, совершённое умышленно в целях скомпрометировать достоверность и авторитетность энциклопедии.

В данном определении сделан акцент на злом умысле вносящего изменения человека. В соответствии с той же страницей правил, вандализмом не является:

- тест новичка (когда пользователи проверяют, действительно ли можно редактировать Википедию);
- изучение возможностей Википедии;
- нарушения нейтральности;
- внесение оригинальных исследований (не подтвержденных в других источниках);
- ошибки и др.

В случае таких изменений, как тест новичка и изучение возможностей Википедии, возможность автоматического обнаружения ожидаема. Поэтому в данной работе примем такое определение вандализма:

Вандальные правки – такие правки, которые можно полностью отменить без потери информации, несущей ценность для статьи.

Таким образом, требуется синтезировать алгоритм A , который для двух версий статьи, представляемых набором характеристик I , а также исторических сведений H , на выходе выдавал бы свое решение относительно совершенных изменений, т.е.:

$$A: I, H \rightarrow \{Vandal, Nonvandal\} \quad (1)$$

Решение будет искажаться со следующими ограничениями:

- Правки в пространстве имен статей (номер 0).
- Вандальность правки определяется по определению выше.
- Игнорирование правок пользователей с расширенными полномочиями – особенности русскоязычного раздела Википедии, такие, как механизм патрулирования, позволяют однозначно доверять действиям этих пользователей (они совершают 60% всех правок).

3. Структура и реализация классификатора

Для решения задачи был выбран подход машинного обучения. Использовалась комбинированная схема классификаторов:

– Классификатор по методу логистической регрессии (далее – классификатор 1), использующий модель данных «bag of words». Предназначен для оценки влияния набора добавленных в статью слов на вандальность правки.

– Основной классификатор, принимающий на вход численную оценку от предыдущего классификатора наравне с остальными выделяемыми из статьи признаками. Конкретный алгоритм классификатора необходимо подобрать экспериментально.

Обучение данных классификаторов осуществлялось на не пересекающихся между собой обучающих выборках.

При реализации использовался язык Python, а также библиотеки:

- Scikit-learn (версия 0.18.1);
- NLTK (версия 3.2.3);
- Matplotlib (версия 2.0.2).

4. Обучение классификатора 1

Процесс обучения классификатора 1 состоит из следующих этапов:

- разбиение предыдущей и текущей ревизии на слова;
- стемминг слов;
- определение добавленных слов;
- формирование данных по модели «bag of words» с использованием «hashing trick»;
- передача на вход алгоритму.

5. Обучение основного классификатора

Для основного классификатора были выделены 45 признаков, которые можно разделить на следующие категории.

Признаки, полученные из текста:

- выходное значение классификатора 1;
- отношение латинских / заглавных букв / цифр к добавленным символам;
- абсолютное и относительное изменение длины текста статьи;
- изменение количества открывающих/закрывающих двойных квадратных скобок (элемент вики-синтаксиса);
- длина наибольшего добавленного слова и др.

Признаки, полученные из комментариев к правке:

- длина комментария;
- отношение цифр/заглавных/латинских букв к общему числу и др.

Признаки, полученные из метаданных:

- поставлен ли флажок «небольшое изменение».

Признаки, полученные для автора статьи:

- время участия в Википедии на момент правки (0 для гостей);
- общее количество совершенных правок и др.

Признаки, полученные из истории изменений:

- доля правок данного пользователя в истории статьи;
- был ли пользователь откочен / отпатрулирован и др.

6. Формирование выборки данных

Обучающие выборки формировались автоматически. Использовались правки за 2016 год, полученные из публикуемых архивов с содержимым баз данных Википедии (URL: <https://dumps.wikimedia.org/>). Были приняты две гипотезы:

- Откоченные правки – вандальные.
- Отпатрулированные правки – не вандальные.

В соответствии с этими гипотезами было выбрано 15 000 случайных откоченных правок, 15 000 случайных патрулированных правок. Это

составляет основную обучающую выборку. Дополнительно, были выбраны 6000 вандальных и 6000 хороших, не пересекающихся с предыдущими правок для обучающей выборки классификатора 1.

В целях исключения влияния последствий принятия этих гипотез и повышения объективности результатов эксперимента, тестирующая выборка была размечена вручную. Были использованы случайно выбранные правки за февраль-март 2017 года. Размечено 2268 хороших правок и 769 вандальных. Разметка осуществлялась по принципу, описанному в разделе 2.

7. Результаты эксперимента

В силу высокой несбалансированности проблемы в качестве метрики, используемой при подборе алгоритма классификации и его параметров, было решено использовать значение площади под ROC-кривой – AUC (Area Under the Curve). ROC-кривая показывает соотношение чувствительности алгоритма и ошибок первого рода. На данный параметр не оказывает влияние несбалансированность задачи [6].

В ходе эксперимента было исследовано поведение алгоритмов при различных значениях их параметров. Лучшие результаты рассмотренных алгоритмов отражены в таблице.

Таблица

Результаты эксперимента

Алгоритм	AUC на тестовой выборке	AUC на фрагменте обучающей выборки
SVM (линейное ядро)	0.862	0.949
SVM (ядро rbf)	0.684	0.773
Random forest	0.913	0.975
Gradient boosting	0.921	0.978
Naive bayes	0.811	0.922
Ada boosting	0.905	0.972

Рассмотрены два варианта:

- оценка на тестовой выборке;
- оценка на фрагменте обучающей (в данном случае перед обучением от нее были отделены случайные 2000 вандальных и 2000 невандалных правок, которые потом рассматривались в качестве тестирующей выборки).

Первый результат отражает поведение алгоритма на реальных данных, второй позволяет оценить влияние обучающей выборки на результат классификации.

Как видно из приведенной таблицы, наилучшее качество классификации было достигнуто при использовании алгоритма градиентного бустинга. Полученный результат был достигнут при следующих параметрах алгоритма:

- 1000 деревьев;
- глубина расщепления 4;
- скорость обучения 0.05.

На рис. 1 изображен график ROC-кривой. По оси абсцисс отмерена доля хороших правок, классифицированных как вандальных (ошибка 1 рода). По оси ординат изображена доля вандальных правок, распознанных как вандальные (точность). Кривая построена при варьировании порогового значения. Диагональная линия отражает классификатор с нулевыми способностями распознавания.

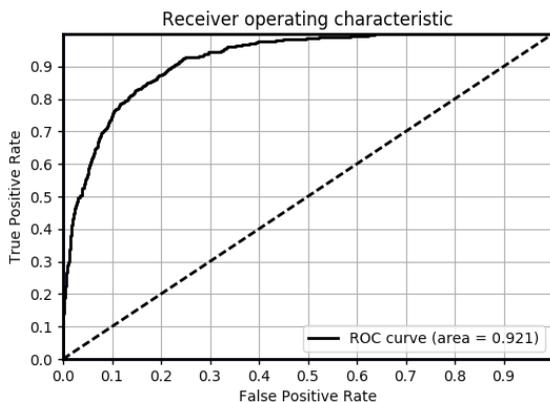


Рис. 1. График ROC-кривой

ROC-кривая позволяет сравнивать точность различных классификаторов. Однако она не дает наглядного представления о том, как полученный алгоритм будет работать в реальных условиях с учетом несбалансированности задачи.

На рис. 2 изображен график кривой Точности-Полноты, который позволяет оценить работу классификатора на реальном потоке правок.

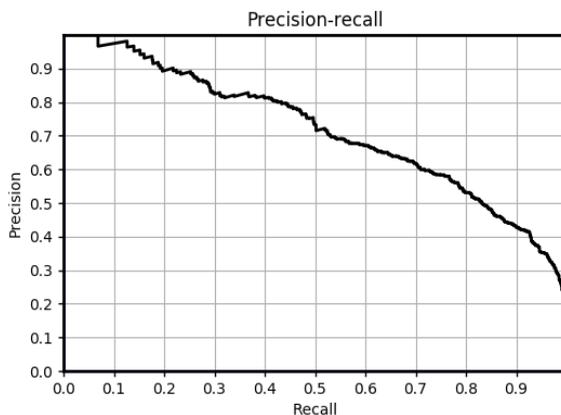


Рис. 2. График PR-кривой

Здесь по оси абсцисс – полнота обнаружения вандализма, по оси ординат – точность. В данном графике учтена доля вандализма в общем потоке правок (7%), а также условие про доверие пользователям с расширенными полномочиями, взятое из постановки задачи. Как видно из рис. 2, полученный классификатор позволяет находить 50% вандализма с точностью 75%, а 80% вандализма – с точностью 53%.

Следует также отметить существенное различие между результатами классификации при оценке на размеченной вручную выборке и на отделенном фрагменте обучающей выборки [7].

Заключение

Таким образом, была рассмотрена возможность автоматизации обнаружения вандализма в русскоязычном разделе Википедии. Полученные результаты позволяют предложить принципы организации работы такой системы, которые позволят ей функционировать с приемлемым уровнем качества. Дальнейшим направлением работ можно считать как непосредственно реализацию такой системы, так и повышение качества классификации, в первую очередь, посредством ручного формирования обучающей выборки.

Список литературы

1. Alexa Top 500 Global Sites [Электронный ресурс]. – Режим доступа : <http://www.alexa.com/topsites>
2. Wikipedia vandalism detection : Combining natural language, metadata, and reputation features / T. Adler, L. Alfaro, M. Santiago [et al.] // Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics. – Berlin : Springer. – 2011. – 277-288.

3. Overview of the 2nd International Competition on Wikipedia Vandalism Detection [Электронный ресурс] : Notebook Papers of CLEF 11 Labs and Workshops. – Режим доступа : <http://ceur-ws.org/Vol-1177/CLEF2011wn-PAN-PotthastEt2011b.pdf>
4. Сайт ПО для вики-проектов MediaWiki [Электронный ресурс]. – Режим доступа : <https://www.mediawiki.org>
5. Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org>
6. Fawcett Tom. An introduction to ROC analysis / Tom Fawcett // Pattern Recognition Letters. – 2006. – № 27. – P. 861-874.
7. Сирота, А. А. Методы и алгоритмы анализа данных и их моделирование в MATLAB : учеб. пособие / А. А. Сирота. – СПб. : БХВ-Петербург, 2016. – 384 с.

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН НА ПРИМЕРЕ ПРОТОТИПА ПРИЛОЖЕНИЯ ДЛЯ ПРОВЕДЕНИЯ ЛОТЕРЕЙ

А.В. Обручников

Студент

Д.И. Соломатин

Старший преподаватель

Введение

Цепочка блоков транзакции или «Блокчейн» (в настоящий момент используется как устоявшийся термин) – технология, которая до сих пор прочно ассоциируется с криптовалютой «Bitcoin». Однако сфера возможного применения данной технологии намного шире оборота криптовалют и охватывает любые задачи, где требуется обеспечить достоверность и неизменность информации. По сути своей, блокчейн – это децентрализованная база данных. Внесение информации в эту базу жестко фиксируется, за счет применения криптологических средств и децентрализованности, эту информацию невозможно удалить или подделать, а значит, такой информации можно доверять.

На сегодняшний день блокчейн находит свое применение во множестве областей, таких как: финансы, регистрация прав владения, интернет вещей, электронное голосование и другие [1]. Одним из самых известных примеров использования блокчейна является проведение банком «Barclays» и стартапом «Wave» первой в мире сделки-аккредитива.

Ее проведение заняло всего 4 часа, тогда как обычно такие операции выполняются в течение 7-10 дней [2]. По словам Байхаса Багдади, главы отделения торговли и оборотного капитала в «Barclays», использование блокчейна помогло решить им четыре задачи: ускорить проведение транзакций, уменьшить затраты, оптимизировать банковские процессы и уменьшить риск подделки документов [3]. Подобные примеры наглядно демонстрируют потенциал этой технологии.

В данной работе ставилась задача изучения возможности применения технологии блокчейн в сфере проведения лотерей. В данной области применение блокчейн позволит гарантировать честность организаторов лотереи по отношению к участникам.

1. Блокчейн

Блокчейн – это построенная по определенным алгоритмам цепочка из блоков транзакций. В системах, строящихся на основе блокчейна, участник имеет полную копию всей цепочки, то есть всех данных. Это гарантирует то, что единолично никто не сможет подменить информацию.

Блок транзакций – это структура данных, которая хранит в себе транзакции. Транзакции являются основным носителем информации в блокчейне.

У каждого блока есть указатель на предыдущий блок. Все они вместе выстраиваются в цепочку, которая будет содержать данные обо всех совершённых когда-либо транзакциях в блокчейне. Самым первым блоком в цепочке называется первичный блок – у него отсутствует предыдущий блок. Обычно в него закладывается информация о параметрах сети.

Блок состоит из заголовка и тела. В заголовке блока указывается хеш текущего блока, хеш предыдущего блока, хеши транзакций и дополнительная информация.

По перечисленным выше правилам добывается блок. Добыть блок – значит собрать все или часть новых транзакций пользователей внести их в блок и добавить блок в блокчейн. В реальных крупных сетях добычей занимаются специальные участники – майнеры. Когда майнер добывает блок, он добавляется в цепочку, как это показано на рис. 1 [4].

Поддерживание целостности такой системы непростая задача, так как необходимо отсылать цепочку новым пользователям системы, следить за тем, чтобы майнеры и пользователи не могли подменить блоки, регулировать процесс добычи и все это при отсутствии централизованного контроля. Для решения этих задач существуют алгоритмы консенсуса. Наиболее известные из них это Proof-of-work (доказательство выполнения работы) и Proof-of-stake (подтверждение доли).

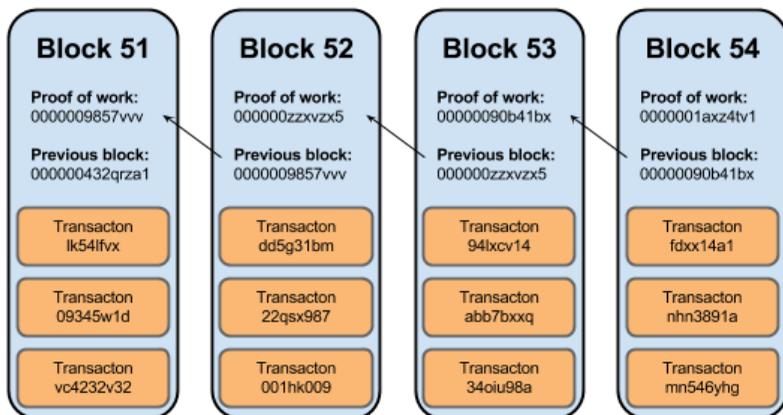


Рис. 1. Пример цепочки блоков

В качестве примера рассмотрим алгоритм добычи блока в системе Proof-of-work. Суть состоит в том, что майнеру необходимо подобрать значение нового заголовка, которое будет равно или ниже определённого числа. Число периодически корректируется в зависимости от скорости добычи блоков. Подбор осуществляется путем вычисления хэш-суммы от разных комбинаций символов. Задача майнера подставлять значения до тех пор, пока не будет вычислена нужная хэш-сумма. За то, что майнер использовал свои вычислительные мощности, система награждает его криптовалютой – это цифровой актив, уникальный для каждой сети. Когда майнер решает задачу, он рассылает сформированный блок другим пользователям, которые проверяют блок.

Из вышесказанного можно сделать вывод, что данные вносимые в блокчейн невозможно подделать или изменить, следовательно, его можно использовать для хранения ценной информации.

2. Практическое применение блокчейн к задаче организации лотерей

В ходе исследования необходимо было реализовать приложение, которое бы могло обеспечить доверие между организатором лотереи и участником. В лотерейных системах хранится информация о ставках игроков для определения победителя. Однако использование обычной базы данных позволяет организатору лотереи с легкостью подменить эту информацию в своих интересах. Проблему доверия между организатором и участником лотереи может снять дополнительная запись информации о ставке игрока в базу данных блокчейн таким образом, чтобы участник лотереи мог данный факт независимо проверить.

В качестве реализации технологии блокчейн в данной работе используется Ethereum – платформа для создания децентрализованных приложений на базе блокчейна, работающих на базе умных контрактов.

Ethereum существует как глобальная развитая сеть, и за внесение информации в его блокчейн необходимо платить. Однако в целях отладки приложения существует возможность развернуть локальную сеть блокчейн, полностью реализующую функциональность Ethereum, что и было сделано.

Как уже было сказано, отличительной особенностью Ethereum является возможность создавать умные контракты – классы, которые функционируют внутри Ethereum. Переменные, описанные в таком классе, будут храниться в блокчейне, их можно изменять или просматривать с помощью функций, тоже описанных в этом классе. Эти функции являются более высокоуровневыми реализациями транзакций.

Так как контракт – это класс, то существует язык программирования, на котором он может быть описан. В Ethereum для написания контрактов используется язык Solidity – статически типизированный JavaScript подобный язык.

Для реализации необходимой для лотереи функциональности был написан контракт, обеспечивающий работу со ставками в виде следующего класса (см. листинг).

Листинг

Умный контракт для лотерей

```
Pragma solidity 0.4.11;
contract lottery
{
    mapping (bytes32 => bool) data;

    function insert(
        string lotID, string fName,
        string passN, string combination
    ) {
        data[sha256(lotID, fName, passN, combination)] =
            true;
    }

    function check(
        string lotID, string fName,
        string passN, string combination
    ) constant returns (bool b) {
        return
            data[sha256(lotID, fName, passN, combination)];
    }
}
```

В контракте есть переменная типа `mapping (bytes32 => bool)` – это структура данных (словарь), которая сопоставляет число с логическим значением. Этим числом будет являться хэш-сумма от идентификатора лотереи, имени и номера паспорта участника и его комбинации. Соответственно, переменная будет хранить в себе информацию, была ли сделана соответствующая ставка.

Так же в контракте есть две функции `insert` и `check`, которые необходимы для внесения и проверке информации о ставке. Функция `insert` объединяет входные данные и хэширует их алгоритмом `sha256`, а затем соотносит этой хэш-сумме значение `true`. Функция `check` выполняет аналогичные действия, но не вносит, а просматривает переменную на предмет наличия определенной хэш-суммы. Таким образом, используя этот контракт, можно внести информацию о ставке или проверить действительно ли человек играет в определенную лотерею с определенной комбинацией.

Для того чтобы контракт стал функционировать в блокчейне участнику сети необходимо сформировать транзакцию, содержащую этот контракт, затем эта транзакция должна быть обработана майнером. Тогда контракт будет считаться внесенным в блокчейн и ему будет присвоен адрес, по которому к нему можно будет обращаться, при помощи JSON-RPC протокола. В приложении это реализовано с помощью фреймворка `web3.js`.

3. Реализация приложения

Результатом исследования стали: веб приложение, база данных и блокчейн инфраструктура. На рис. 2 приведена схема развертывания и взаимодействия этих элементов между собой и пользователем.

Функционально приложение разделено на несколько разделов:

1. Организация лотереи – для этого организатору необходимо перейти во вкладку «Для организаторов» внести данные и параметры лотереи и начать лотерею, это приведет к созданию в базе данных лотереи и генерации билетов для участия. Интерфейс для организатора приведен на рис. 3.

2. Участие в лотерее – для этого участнику необходимо перейти во вкладку «Для участников», выбрать лотерею, в которой он хочет участвовать, приобрести билет и активировать его. Эти действия приведут к тому, что будет создана запись об участнике и о его ставке, а так же произойдет посылка транзакции на добавление ставки в сеть блокчейн. На рис. 4 приведена html страница “Для участников”.

3. Проверка участия – для этого, после выбора лотереи, необходимо активировать кнопку «проверить участие» и внести свои данные. После

этого приложение обратится к блокчейну, проверит значение ставки и отобразит результат в браузере.

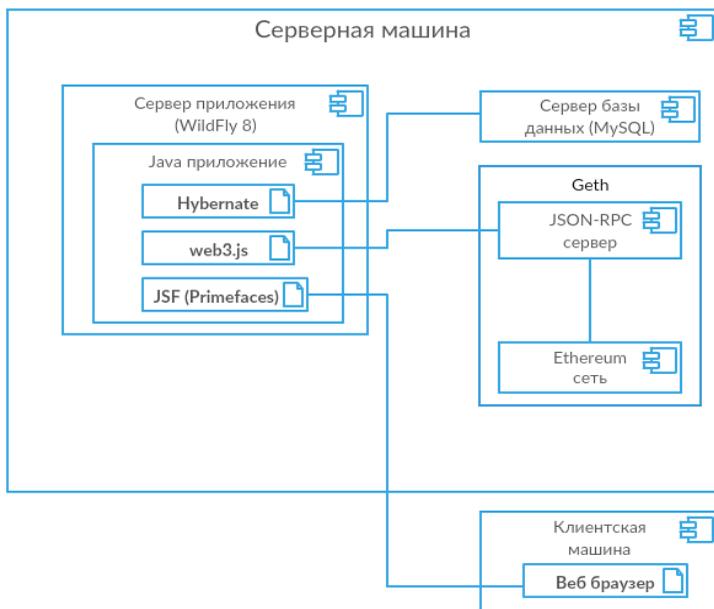


Рис. 2. Диаграмма развертывания приложения

The screenshot shows a web interface for creating a lottery. On the left is a navigation menu with the following items:

- Главная
- Для участников
- Для организаторов

The main content area is titled "Блочная лотерея" and contains the following form fields:

- Название:
- Описание:
- Параметры лотереи: Количество билетов:
- Начать лотерею:

Рис. 3. Интерфейс для создания лотереи

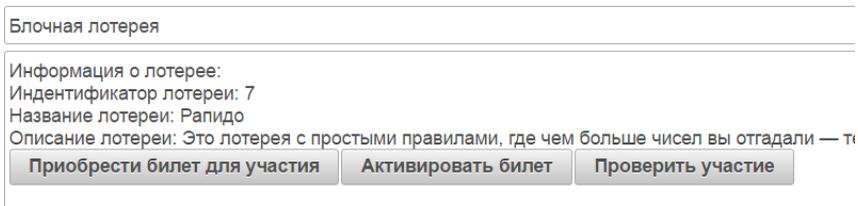


Рис. 4. Интерфейс для участия в лотерее

Заключение

В результате проделанной работы было реализовано приложение, которое для подтверждения ставок дополнительно сохраняет информацию в блокчейн (Ethereum), что позволяет гарантировать честность при проведении лотереи. Безусловно, данное приложение является только прототипом и непригодно для реального использования, однако оно показывает, что средства работы с блокчейн уже сейчас достаточно развиты и могут быть использованы в различных проектах.

Список литературы

1. Wirex, 20 областей применения Блокчейн вне финансовых сервисов, ч. 1 [Электронный ресурс]. – Режим доступа : <https://geektimes.ru/company/wirex/blog/281140/>
2. Reuters, проведение сделки-аккредитива [Электронный ресурс]. – Режим доступа : <http://www.reuters.com/article/us-banks-barclays-blockchain-idUSKCN11D23B>
3. GTR, интервью с Байхасом Багдади [Электронный ресурс]. – Режим доступа : <http://www.gtreview.com/news/global/first-live-blockchain-trade-transaction-conducted/>
4. Брикман Е. Bitcoin by analogy [Электронный ресурс]. – Режим доступа : <http://www.ybrikman.com/writing/2014/04/24/bitcoin-by-analogy/>

РЕКОМЕНДАЦИИ ПРИ ВЫБОРЕ СИСТЕМЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

А.С. Паневина

Студент

И.В. Абрамов

Доцент

Введение

Современный мир постоянно развивается, на смену одним технологиям, приходят совершенно новые. Для того, чтобы компания могла конкурировать с остальными необходимы постоянное развитие, быстрая реакция на смену внешней ситуации. Одним из направлений автоматизации является внедрение Системы Электронного Документооборота (СЭД). Система предназначена для повышения эффективности работы предприятия, автоматизации делопроизводства, работы с канцелярией, внутренними документами и управления потоками задач. Переход на электронную работу с документами и распоряжениями позволяет руководителю ускорить процессы управления, планирования и анализа. Таким образом обеспечивается быстрый поиск, структурирование документов, определение объема задач и контроль исполнения, что благоприятно влияет на скорость работы и реакцию на изменение внешней среды, которые влияют на конкурентоспособность предприятия.

1. СЭД (Система электронного документооборота)

Цель данной статьи – описание алгоритма выбора СЭД с учетом их основных характеристик, функциональности и значимости решаемых задач.

Система автоматизации документооборота, система электронного документооборота (СЭДО) — автоматизированная многопользовательская система, сопровождающая процесс управления работой иерархической организации с целью обеспечения эффективного и быстрого выполнения этой организацией своих функций; это система компьютерного ПО, которая нужна для организации работы с электронными неструктурированными документами. При этом предполагается, что процесс управления опирается на человеко-читаемые документы, содержащие инструкции для сотрудников организации, необходимые к исполнению [1].

Виды СЭД по количеству реализуемых функций:

- Комплексные или ЕСМ-системы (Enterprise content management) – ПО (программное обеспечение) для Управления корпоративным контентом
- Электронные архивы
- Системы делопроизводства
- Workflow-системы [2].

2. Задачи и область применения СЭД

СЭД помогает упростить деятельность в фирме:

– Обеспечение прозрачности работы. Руководитель может проконтролировать каждый этап прохождения документооборота от одного сотрудника к другому и степень его завершенности.

– Повышение качества исполнительской дисциплины. Происходит увеличение числа выполненных задач, сокращается процент забытых и просроченных заданий.

– Сокращение временных затрат на операции с документами. Осуществление поиска, согласования, создания документов намного ускоряется, а заодно ускорение происходит и в остальных процессах (взаимодействии с клиентами, передача документов между подразделениями и т.д.)

– Обеспечение конфиденциальности информации. Каждый сотрудник имеет определенный уровень доступа, набор функций, при которых ему открыты действия, соответствующие его обязанностям и квалификации. Любое действие (чтение, изменение, подписание) протоколируется.

– Обеспечение выполнения международных стандартов ISO 9000. Прозрачность документооборота и информационного воздействия.

– Безболезненное и легкое внедрение обновлений и расширений. Обучение работе по новым правилам происходит заметно быстрее, а так же новые сотрудники быстро вливаются в среду.

– Развитие корпоративной культуры. Отлажено взаимодействие между сотрудниками, организованы горизонтальные связи, четкая работа всех подразделений, каждый работник имеет свой уровень доступа к определенным документам и функциям. Только при таких условиях создается эффективная работоспособная команда.

Помощь человеку – главная цель автоматизации. Задача автоматизации – распределение ряда обязанностей и ответственности между сотрудниками и контроль этапов работы непосредственно онлайн. Компания, взявшая себе в помощники СЭД, имеет конкурентные преимущества в скорости и качестве обслуживания клиента.

Качества, которыми должны обладать современные системы:

- Управляемость документов – единые правила для всех сотрудников, обеспечение безопасности и контроля версий, беспрепятственный экспорт и импорт файлов.

- Ведение записей – сопровождение задания/поручения/документа от создания до завершения и отправки в архив.

- Возможность коллективного пользования в проектных командах.

- Взаимодействие с образами документов – сканирование, хранение и управление бумажными оригиналами.

- Контроль потоков работ – БП автоматизируются, выполняется постановка задач с четким регламентом по выполнению.

- Поддержка мультимедиа – хранение в структурированном виде изображений, видеоматериалов и аудиофайлов.

- Единство знаний – соединение в одной системе всей информации, которая необходима для бизнеса

- Поддержка web-клиента и мобильная версия для доступа к системе в любой момент времени без привязки к рабочему месту.

Для того, чтобы выбрать СЭД, покрывающую большее число БП фирмы, нужно учесть ряд критериев, позволяющих оценить функциональные возможности системы. Для этого необходимо определить:

- цели и задачи, которые должна решать система;

- подразделения, требующие автоматизации;

- наличие территориально распределенных подразделений;

- объем и сроки хранения документов;

- необходимость осуществления контроля за исполнительской дисциплиной;

- необходимость в интеграции с другими системами, используемыми в компании;

- требования по модификации системы под нужды компании;

- требования, выдвигаемые к безопасности хранящихся документов;

- необходимость использования ЭП(электронная подпись);

- необходимость мобильной версии [3].

3. Алгоритм выбора

После того, как выбраны основные направления, которым должна соответствовать система, стоит начать поиск СЭД по алгоритму:

1. Изучение СЭД по функциональности. Важно найти список конкретных задач, которые реализует СЭД. Общие фразы не демонстрируют функциональность системы, а значит это говорит о недостаточно хорошо проработанной системе.

Многие системы схожи по своей функциональности, но тем не менее имеют ряд различий: либо определенный функционал необходимо докупать, либо не реализовано в полной мере, либо отсутствует вообще.

К примеру, на сайте системы ТЕЗИС представлен раздел «Возможности» с конкретным перечнем задач поделенным на блоки по функциям [4].

Другим примером служит сайт DocsVision в разделе «Каталог решений» предоставляются решения в общем случае без конкретных решаемых задач [5].

Определившись с конкретными задачами, которые должна решить СЭД, формируется список функционала, а следовательно на 1 этапе уже можно отсеять большинство неподходящих заданным требованиям систем.

2. Изучение примеров использования. Важно знать, как реализован тот или иной функционал. Необходимо найти разделы с реализованными конкретными модулями, а именно тематические презентации или видео, которые доступно продемонстрируют интересующие вас операции.

Хорошим примером служит систем Verдох, в разделе «Виртуальный тур» показаны готовые решения, сопровождающиеся скриншотами по каждому этапу и подробным описанием [6].

Напротив, система Логика бизнеса в разделе «Решения» не рассказывает о предоставляемых решениях, нет конкретно реализованных модулей, а увидеть можно только общий набор фраз про введение СЭД на предприятии [7].

Опять же большинство систем схожи по функционалу, но это определенно по-разному будет реализовано, примеры помогут увидеть различия и отметить для себя наиболее подходящую и удобную реализацию.

3. Изучение скриншотов. Снимков экрана должно быть много и хорошего качества, это говорит о том, что все, что вы видели в описании, соответствует действительности, и разработчики стремятся продемонстрировать хорошее качество своего товара.

Optima-workflow в разделе «Типовые решения» по каждому из модулей предоставляет большое количество скриншотов с подробным описанием решаемых задач [8].

В то время как на сайте системы Логика бизнеса нет скриншотов.

Скриншоты позволяют увидеть, как система выглядит в рабочем состоянии, насколько понятен общий вид рабочей платформы и ее отдельные модули.

4. Изучение видеоматериалов. Обзорные видеоролики дают понять, как работает СЭД, какие функции в ней реализованы и насколько они

удобны в пользовании. Обучающие видео и записи вебинаров так же говорят о качестве продукта.

Каждый модуль возможностей системы ТЕЗИС в разделе «Демонстрация» сопровождается видеорядом на конкретное реализованное решение с подробным рассказом менеджера схемы работы в СЭД [9].

Doclogic имеет общую презентацию про свою систему и два видеоряда плохого качества без сопровождения и объяснения производимых действий [10].

Видеозаписи или вебинары проводятся менеджерами по продажам, они рассказывают подробно и показывают, как именно выполняется заявленный функционал, прямо отвечают на вопросы, возникающие по ходу изучения. Это позволяет понять какая из систем подходит и все ли опции реализованы в полной мере, а значит на данном этапе так же отсекутся некоторые представители продуктов СЭД.

5. Проверка наличия демо-стенда или бесплатного пробного периода. Попробовав, как работает система изнутри, вы сможете ответить на вопросы: «Реализует ли система все необходимые функции?», «Удобна ли она в использовании?», «Нужна ли нашей компании автоматизация?» и др. Вы сможете определиться с выбором нужного вам варианта. Разработчики, предоставляющие доступ, уверены в полноте и работоспособности своей системы, что говорит о ее надежности и качестве.

Система ТЕЗИС предоставляет демо-стенд на неограниченный период времени, но с ограниченным набором функций, и прилагает к нему запись вебинара с уже конкретно реализованной под заказчика системой, где подробно рассказывает каждое выполненное им действие [9].

E1 ЕФРАТ предоставляет бесплатную версию на период 30 дней и презентацию, но зато с полным функционалом [11].

Чтобы сократить количество рассматриваемых вариантов, необходимо на основе общих характеристик посмотреть различия в реализации и степень удобства. Видеоматериалы помогут разобраться с логикой программы, но в руках специалиста всё выглядит логично и понятно, тогда как на деле вам такое не подойдет. Поэтому обязательно нужно изучить СЭД изнутри и попробовать проделать определенный цикл задач.

6. Важная часть – стоимость. Прайс-листы должны быть информативны и понятны, каждый из подключаемых модулей должен иметь цену, должны быть понятны принципы ценообразования.

Система ТЕЗИС имеет понятное ценообразование, есть различия в стоимости разных редакций и при этом понятно, какой набор функций в каждой из них и сколько это стоит [12].

В то время как в Directum разобраться в комплектациях и стоимости без менеджера или подробного изучения системы нельзя [13].

Выясняя ценовые характеристики, нужно учитывать количество подключаемых пользователей или мест работы, стоимость базовой комплектации и дополнительных подключаемых модулей и расширений, ежегодных обновлений, а так же стоимость технической поддержки и обслуживания.

Заключение

Данная статья посвящена советам по выбору СЭД (Система Электронного Документооборота). Выделены основные задачи и качества современных систем автоматизации делопроизводства, так же определены основные области применения СЭД. Чтобы грамотно выбрать систему, необходимо прежде всего определить ряд критериев, которым она должна соответствовать.

Универсального решения не существует, плюсы и минусы есть у всех, поэтому стоит проанализировать, протестировать рынок предложенных вариантов. Это позволит найти систему, которая будет решать задачи, связанные со спецификой вашей компании. Изучайте сайты разработчиков, связывайтесь с менеджерами и службами техподдержки, потратив некоторое количество времени вы найдете, то что вам подходит.

Список литературы

1. Википедия [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/Система_автоматизации_документооборота
2. ЕСМ // «Бизнес Консалт» — Профессиональные бизнес решения. 2017 [Электронный ресурс]. – Режим доступа : <https://bconsult.pro/proektiruem-i-razrabatyvaem/esm>
3. Как выбрать систему электронного документооборота? [Электронный ресурс] // ТЕЗИС. – Режим доступа : <https://www.tezis-doc.ru/>
4. Возможности [Электронный ресурс] // ТЕЗИС. – Режим доступа : <https://www.tezis-doc.ru/features>
5. Каталог решений [Электронный ресурс] // DocsVision. – Режим доступа : <http://www.docsvision.com/kupit/dv-catalog/>
6. Виртуальный тур [Электронный ресурс] // Verdox. – Режим доступа : <http://verdox.ru/tour/>
7. Решения [Электронный ресурс] // Логика бизнеса. – Режим доступа : <http://ecm.blogic20.ru/logikaecm/docflow-solutions>
8. Типовые решения. Делопроизводство [Электронный ресурс] // Optima-workflow – Режим доступа : <http://optima-workflow.ru/clerical-work.html>

9. Демонстрация [Электронный ресурс] // ТЕЗИС. – Режим доступа : <https://www.tezis-doc.ru/demo>
10. Смотреть презентацию [Электронный ресурс] // Doclogix. – Режим доступа : <http://www.doclogix.ru/ru/produkt/poleznaja-informatsija/smotret-video/>
11. Демо-версия [Электронный ресурс] // Е1 ЕФРАТ. – Режим доступа : <http://www.evfrat.ru/download/>
12. Купить [Электронный ресурс] // ТЕЗИС. – Режим доступа : <https://www.tezis-doc.ru/buy>
13. Цены [Электронный ресурс] // Directum – Режим доступа : <http://www.directum.ru/price>

СТАТИСТИЧЕСКИЙ АНАЛИЗ ПАРАМЕТРОВ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА

А.Г. Плотникова

Студент

М.Г. Матвеев

Профессор

Введение

Статистический контроль технологического процесса является одним из наиболее эффективных и перспективных направлений повышения качества продукции серийного производства. Традиционный подход к контролю качества подразумевает под собой проверку готовой продукции и отбраковку не соответствующих требованиям единиц. Но такая постфактум проверка, когда бракованная продукция уже создана, приводит к экономическим потерям. Эффективнее использовать стратегию предупреждающих потерь, которая позволяет исследовать сам ход процесса производства с целью его контроля и регулирования.

Любой технологический процесс подвержен изменчивости, даже если он функционирует в полном соответствии с установленными правилами и нормами [1]. Различается два вида изменчивости:

1. Вызванная воздействием простых или неустранимых причин;
2. Вызванная воздействием неслучайных (специальных) причин.

Первый вид обусловлен набором разнообразных причин, существующих всегда и фактически неустранимых до тех пор, пока сам процесс не будет изменен. Их появление связано с особенностями функционирования процесса. Исключение или уменьшение влияния

обычных причин требует как управленческих решений, так и выделения ресурсов на улучшение процесса.

Второй вид изменчивости обусловлены реальными переменными в процессе, вызванные причинами, не присущими процессу внутренне и являющимися устраняемыми. Они появляются ввиду возникновения особых изменений в самом технологическом процессе или окружающей среде, которые можно выявить.

Статистический контроль за технологическим процессом решает задачу обеспечения и поддержания процессов на приемлемом и стабильном уровне, гарантируя соответствие продукции требованиям. Основным статистическим инструментом такого подхода являются контрольные карты – графический способ представления и сопоставления информации, основанной на последовательности выборок, отражающих текущее состояние процесса, с границами, установленными на основе внутренней присущей процессу изменчивости [2]. Использование контрольных карт и их анализ позволяет определить находится ли процесс в статистически управляемом состоянии, а затем поддерживать управление им.

1. Контрольные карты

Контрольная карта – это линейчатый график, построенный на основании данных измерений показателей процесса в различные периоды времени [3]. Он позволяет отразить динамику изменений показателя и за счет этого контролировать процесс. Наносимый на карту статистический показатель должен быть коррелирован с качеством производимой продукции. При этом порядок нанесения на карту значений показателя привязан к времени или порядку отбора выборок.

Помимо значений выборочного статистического показателя качества на карту наносят статистические определяемые контрольные границы относительно центральной линии (CL), которые называются верхней контрольной границей (UCL) и нижней контрольной границей (LCL) (рис. 1).

Целью использования контрольных карт заключается в обнаружении изменчивости, вызываемой особыми причинами, на фоне собственной изменчивости. Контрольные границы устанавливаю допустимые пределы собственной изменчивости, и их нарушение является показателем действия особых причин и выхода процесса из управляемого состояния. В основе расчетов контрольных границ лежат допущения, связанные с применением различных вероятностных моделей распределения для аналитического описания собственной изменчивости процесса.

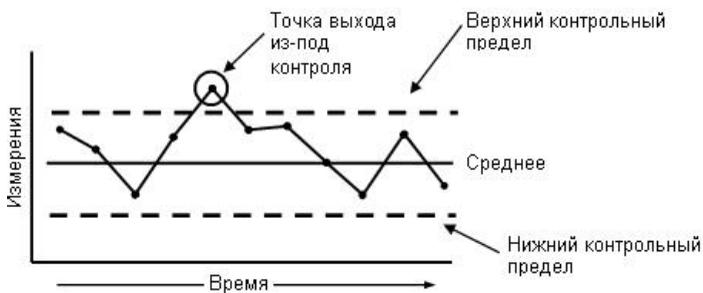


Рис. 1. Вид контрольной карты

Контрольные карты используются для определения состояния контролируемого процесса. Подобную оценку можно получить двумя способами [4]:

1. Осуществляя проверку замеров важнейших параметров изделия;
2. Осуществляя проверку отдельных качественных характеристик изделия.

В первом случае используются контрольные карты количественного, а во втором – альтернативного признака.

2. Контрольные карты количественных признаков

Если генеральная совокупность имеет нормальное (близкое к нормальному) распределение с μ – среднее значение, σ – стандартное отклонение, тогда выборочное распределение выборочного среднего также является нормальным и имеет μ – среднее значение, σ/\sqrt{n} – стандартную ошибку, где n – объем выборки. Графическая иллюстрация граничных значений для выборочного распределения выборочного среднего приведена на рис. 2.

Значения μ и σ получают по результатам расчетов среднего значения и стандартного отклонения соответствующих параметров технологического процесса на протяжении длительного периода времени. Изменчивость внутри подгрупп является мерой случайных вариаций. Для получения оценки σ вычисляют выборочное стандартное отклонения или умножают выборочных размах на соответствующий коэффициент. Это мера не включает межгрупповых вариаций, а оценивает только изменчивость внутри подгрупп.

Границы $\pm 3\sigma$ указывают, что около 99,7% значений характеристики подгрупп попадут в эти пределы, если процесс находится в статистически управляемом состоянии. А любое выборочное значение, попадающее за

границы $\pm 2\sigma$, может служить предостережением о грозящей ситуации выхода процесса из состояния статистической управляемости.

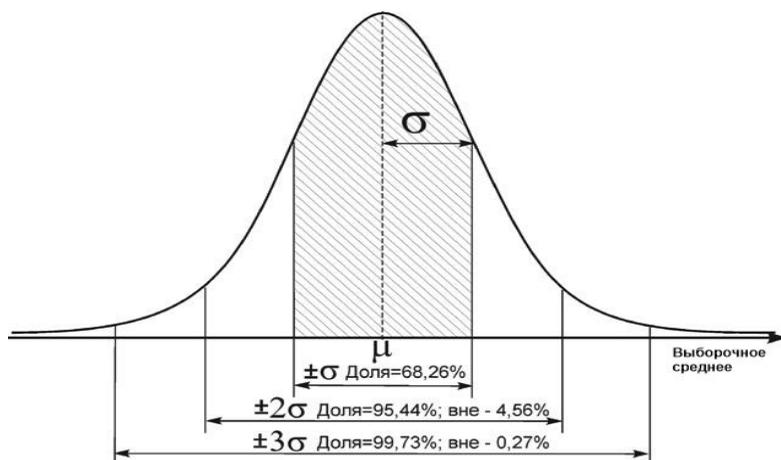


Рис. 2. Выборочное распределение выборочного среднего

3. Контрольные карты альтернативных признаков.

Альтернативные данные – это наблюдения, фиксирующие наличие или отсутствие характеристик (признаков) у каждой единицы продукции. На основе этих данных производится подсчет числа единиц, которые обладают или не обладают установленным признаком.

При использовании контрольных карт альтернативных признаков достаточно рассмотреть одну карту, так как предполагаемое распределение имеет только один независимый параметр – средний уровень. При этом карта долей несоответствующих единиц продукции (p) и карта числа несоответствующих единиц (np) основаны на биномиальном распределении, а карта числа несоответствий (c) и карта числа несоответствий, приходящихся на единицу продукции (u) – на распределении Пуассона. Расчеты для карт производятся одинаково, за исключением случаев непостоянства объема подгрупп. Параметры, необходимые для построения карт по альтернативному признаку не отличаются от параметров для карт по количественному признаку: центральная линия, верхняя контрольная граница и нижняя контрольная границы.

4. Автоматизация построения и анализа контрольных карт.

По значениям статистических характеристик в последовательных выборках и по их расположению относительно контрольных границ можно своевременно обнаружить нарушения в технологическом процессе и принять меры по их устранению. Предложенная автоматизация процесса построения и анализа контрольных карт упрощает решение широкого круга задач контроля и обеспечения качества, а также позволяет выполнять данные работы операторам без приобретения специальных знаний в области контроля качества.

Довольно обширный объем критериев затрудняет и увеличивает во времени ручной анализ. Для упрощения данных процедур и было принято решение разработать программное обеспечение, позволяющее проводить анализ хода технологического процесса, рассчитывать и строить контрольные карты Шухарта, управлять процессами контроля качества.

Рассмотрим критерии выявления по контрольным картам отклонений в технологическом процессе:

1. Одна или более точек оказались за UCL или LCL ;
2. Серия из семи или более точек оказалась с одной стороны от CL . Под серией понимается проявление такого состояния, когда ряд точек неизменно оказывается по одну сторону от центральной линии;
3. Шесть или более точек подряд образуют ряд непрерывно возрастающих (убывающих) значений. Тогда говорят, что имеет место тренд (дрейф), что сигнализирует о ненормальности технологического процесса;
4. Четырнадцать точек скачут вверх-вниз;
5. Две из трех точек подряд оказались в крайней трети диапазона контрольных пределов (приближение к UCL и LCL);
6. Четыре из последовательных пяти точек находятся с одной стороны от CL и не попадают в центральную треть диапазона контрольных пределов;
7. Восемь точек расположены с двух сторон от CL , ни одна из которых не попадает в центральную треть диапазона контрольных пределов;
8. Периодичность изменения контролируемой характеристики. Если точки укладываются на кривую, имеющую повторяющуюся форму «подъем-спад» с примерно одинаковыми временными интервалами, то это тоже характеризует ненормальность процесса;

Пятнадцать точек подряд оказались в пределах центральной трети диапазона контрольных пределов.

При применении контрольных карт возможно появление двух видов ошибок: первого и второго рода:

1. Ошибка первого рода возникает при отнесении процесса к статистически неуправляемому (случайное появление точки за пределами контрольных границ), когда на самом деле в процессе нет нарушений.

2. Ошибка второго рода возникает, когда процесс находится в статистически неуправляемом состоянии, а точки случайно оказываются внутри контрольных границ. При этом упускается возможность предупредить рост выхода бракованной продукции.

Система карт Шухарта учитывает только ошибки первого рода, которые определяются $0,3\%$ в $\pm 3\sigma$. Так как в общем случае непрактично делать полную оценку потерь от ошибки второго рода в конкретно рассматриваемой ситуации, а удобнее рассматривать малый объем подгруппы, то используют границы $\pm 3\sigma$ и акцентируют внимание на управлении и улучшении качества процесса. Стоит отметить, что Шухарт подчеркивал именно эмпирическую полезность контрольных карт для установления отклонения от состояния статистической управляемости, а не их вероятностную интерпретацию.

Заключение

Таким образом, рассмотренный метод контрольных карт является статистическим методом обработки информации для принятия обоснованных решений. Подобный взгляд на вещи является наиболее эффективным и объективным средством диагностики процесса. Многие фирмы тратят большие средства для обучения персонала статистическим методам и способам их применения. В связи с этим возрастает важность автоматизации данного процесса. Разработанное в ходе научной работы программное обеспечение может способствовать применению статистических методов в более широком диапазоне областей деятельности.

Список литературы

1. Эддоус М. Методы принятия решений: пер. с англ. / М. Эддоус, Р. Стэнфилд. – М. : Аудит, 1997. – 590 с.
2. Солонин С.И. Метод контрольных карт [Электронный ресурс] : Электронное текстовое издание / С. И. Солонин. – Екатеринбург: УрФУ, 2014. – 213 с.
3. ГОСТ Р ИСО 7870-2-2015. Статистические методы. Контрольные карты. Часть 2. Контрольные карты Шухарта. – Введ. 2016-12-01. – М. : Стандартиформ, 2016. – 41 с.
4. Уилер Д. Статистическое управление процессами. Оптимизация бизнеса с использованием контрольных карт Шухарта: пер. с англ. / Д. Уилер, Д. Чамберс. – М. : Альпина Бизнес Букс, 2009. – 409 с.

СВЯЗЬ ПЛАТЫ STK600 С ПК

Х.А. Полещук

Студент

Н.А. Тюкачев

Доцент

Введение

Связь платы с компьютером является важной проблемой в современном мире. Платы имеют самое разнообразное применение, и в некоторых случаях во время своей работы имеют необходимость в контроле со стороны оператора, но не имеют для этого средств.

Однако каждая плата имеет разъем для USB соединения с компьютером для программирования платы. Так как USB соединение предоставляет возможность для двунаправленного общения с компьютером, то именно оно может послужить средством для контроля работы платы.

Постановка задачи

Со стороны пользователя необходимо отправлять команды на чтение/запись в регистры LMН0303 и LMН0384 с использованием COM порта.

Со стороны платы STK600 (рис. 1) необходимо реализовать создание COM порта с помощью USB CDC и исходя из команды, полученной от пользователя прочитать/записать значение в регистр, а затем отправить пользователю ответ.



Рис. 1. Плата STK600

В данной статье рассмотрено построение пользовательской части, а также создание COM порта с помощью USB communications device class (CDC), чтение команд, полученных от пользователя, и отправка ответа со стороны платы.

Для начала необходимо научить плату слушать свое соединение и обрабатывать сигналы.

Вообще настройка платы с нуля это очень трудоемкий процесс, требующий больших затрат времени и хорошего знания не только программирования, но и принципов работы с платами вообще и данной конкретной платой в частности. Поэтому в общем случае изготовитель платы обычно оставляет в свободном доступе все необходимые для работы с его изделием библиотеки, которыми конечный пользователь и пользуется.

Например, для работы с Arduino существует Arduino Studio. В нашем случае (для STK600) лучше всего использовать Atmel Studio. Для ПО на стороне платы в качестве платы можно использовать для USB части USB communication device class (USB CDC). Он является составным классом устройства универсальной последовательной шины. USB CDC может включать один или несколько виртуальных COM портов.

Следующим действием является создание проекта на стороне пользователя. Пусть это будет C++ приложение. Разработка будет вестись в Qt Creator.

1. USB CDC на STK600

Для более полного понимания поставленной задачи необходимо рассмотреть архитектуру USB CDC (рис. 2) и последовательность выполняемых команд (рис. 3).

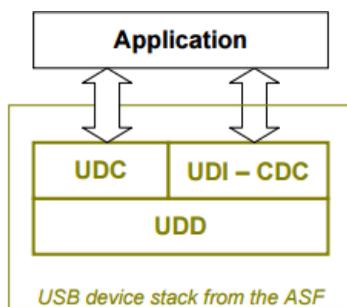


Рис. 2. Архитектура USB CDC

На рис. 2 изображены главные модули организации CDC приложения [1]:

- Пользовательское приложение;
- Интерфейс устройства USB CDC (UDI-CDC);
- Контроллер устройства USB (UDC);
- Драйвер устройства USB (UDD);

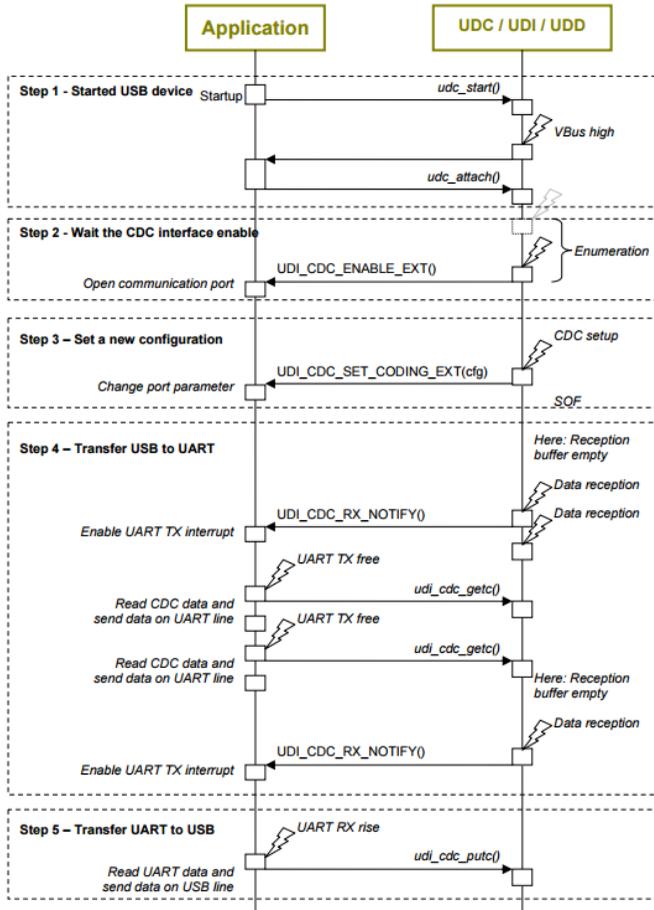


Рис. 3. Последовательность действий

На рис. 3 изображена последовательность действий работы пользовательского приложения с USB CDC.

Но перед этим необходимо совершить ряд подготовительных действий:

– Сначала надо вызвать метод `irq_initialize_vectors()` для того, чтобы инициализировать векторы прерывания (включая все уровни прерываний, с векторами, расположенными в разделе приложения и с фиксированным приоритетом) [2].

– После вызывается метод `cpu_irq_enable()` для глобального разрешения прерываний.

– Следующим вызывается метод `sleepmgr_init()`, который инициализирует службу диспетчера сна.

– Потом вызывается метод `sysclk_init()`, который инициализирует службу часов.

– Затем вызывается метод `board_init()` для того, чтобы гарантировать правильную инициализацию целевого оборудования платы.

– Затем можно включить светодиод, чтобы показать что все идет как запланировано (`LED_On(LED0_GPIO)`).

Теперь, когда подготовительные действия завершены, можно приступить к работе непосредственно с USB CDC. Можно отойти от варианта, указанного на рис. 3 и поступить следующим образом:

1. Запустить стек USB устройства (вызвать метод `udc_start()`)
2. Вызвать метод `udc_attach()` (если в драйвер включен элемент управления VBus, он будет подключать устройство, когда обнаружен приемлемый уровень Vbus от хоста)
3. Создать буфер для хранения команд
4. Ждать и получить значение по линии CDC (`udi_cdc_getc()`)
5. Считать буфер RAM в линии CDC (`udi_cdc_read_buf(void*, iram_size_t)`)
6. Среагировать на команду определенным образом
7. Отправить ответ (`udi_cdc_write_buf(void*, iram_size_t)`)

Таким образом, метод `main()` выглядит как на листинге 1:

Листинг 1

Основная часть программы

```
int main(void)
{
    irq_initialize_vectors();
    cpu_irq_enable();
    sleepmgr_init();
    sysclk_init();
    board_init();
    LED_On(LED0_GPIO);
    udc_start();
    udc_attach();
    char* buf = (char*)malloc(sizeof(char) * MAX_LEN);
    char* response = (char*)malloc(sizeof(char) * MAX_LEN);
```

```

while(1)
{
    udi_cdc_getc();
    udi_cdc_read_buf(buf, MAX_LEN);
    .....
    udi_cdc_write_buf(response, MAX_LEN);
}
delete buf;
delete response;
}

```

2. Приложение на стороне ПК

Как ранее было сказано, USB CDC создает COM порт.

Для работы с COM портами в Qt имеется библиотека QSerialPort [3].

Для начала необходимо посмотреть и выбрать COM порт к которому мы хоти подключиться.

Код, необходимый для просмотра подключенных к компьютеру COM портов (листинг 2).

Листинг 2

Код для просмотра COM портов

```

foreach (const QSerialPortInfo &info,
QSerialPortInfo::availablePorts()) {
    QString name = info.portName();
    QString description = info.description();
    QString manufacturer = info.manufacturer();
}

```

После выбора необходимого COM порта необходимо к нему подключиться (листинг 3).

Листинг 3

Подключение к COM порту

```

QSerialPort* serial = new QSerialPort(this);
serial->setPortName(Com::name);
serial->open(QIODevice::ReadWrite);
serial->setBaudRate(QSerialPort::Baud9600);
serial->setStopBits(QSerialPort::OneStop);
serial->setParity(QSerialPort::NoParity);
serial->setDataBits(QSerialPort::Data8);
serial->setFlowControl(QSerialPort::NoFlowControl);

```

Далее необходим класс для формирования команды (листинг 4).

Листинг 4

Класс формирования команды

```

class Command
{
public:
    bool read;
}

```

```

QString chip;
int chipID;
QString reg;
uint8_t value;
QString reg_field;
Command();
QByteArray ToString();
private:
    QString inttohex(uint8_t dec);
};

```

Формирование команды происходит, как показано на листинге 5.

Листинг 5

Код формирования команды

```

Command cmd;
QByteArray text;
cmd.read = true;
cmd.chip = "LMH0303";
cmd.chipID = 5;
cmd.reg = "Status";
cmd.value = 0;
text = cmd.ToString();

```

Составные части команды формируются исходя из действий пользователя с окном работы с регистрами (рис. 4).

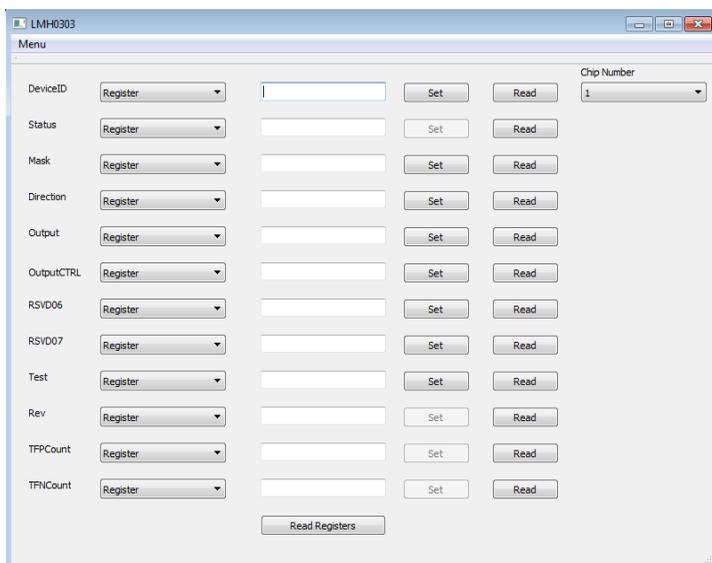


Рис. 4. Окно чтения и записи в регистры

Отправлять сообщение можно следующей командой:

```
serial->write(text);
```

Привязываем метод чтения сообщений из COM порта к событию его получения:

```
connect(serial, &QSerialPort::readyRead, this, readData);
```

Метод выглядит, как показано на листинге 6.

Листинг 6

Метод чтения сообщений из COM порта

```
void readData()  
{  
    QByteArray data = serial->readAll();  
}
```

Заключение

В статье было рассмотрено построение пользовательской части, позволяющей отправлять команды на чтение/запись в регистры LMN0303 и LMN0384 с использованием COM порта, а также создание COM порта с помощью USB communications device class (CDC), чтение команд, полученных от пользователя, и отправка ответа со стороны платы.

USB соединение предоставило возможность для двунаправленного общения с компьютером и послужило средством для контроля работы платы.

Список литературы

1. Atmel AVR4907: ASF – USB Device CDC Application [Электронный ресурс] : тех. документ / Atmel Corporation. – Режим доступа : <http://www.atmel.com/Images/doc8447.pdf>
2. STK600 User Guide [Электронный ресурс] : тех. документ / Atmel Corporation. – Режим доступа : http://www.atmel.com/ru/ru/Images/Atmel-32221-STK600_UserGuide.pdf
3. Qt Documentation [Электронный ресурс] : база данных. – Режим доступа : <http://doc.qt.io>

КЛИЕНТ-СЕРВЕРНОЕ ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ЛИЧНЫМ ВРЕМЕНЕМ

Д.В. Разуваев

Студент

А.И. Другалев

Ассистент

Введение

Какой самый ценный ресурс для человека? Время. С одной стороны это потому, что данный ресурс можно обменять на любой другой ресурс. Можно потратить время на любимое занятие, будь то футбол или компьютерные игры – получаем удовольствие и хорошее настроение. Потратив на чтение учебников и конспектов, получим хорошую оценку на экзамене. Можем потратить время на получение высшего образования и учебу – получаем хорошо оплачиваемую работу. С другой стороны, время никак нельзя получить обратно. Т.е. мы можем конвертировать этот ресурс на практически любой другой, но обратно поменять что-либо на время невозможно. И не важно, сколько у вас знаний, влияния и денег – повернуть время вспять мы не сможем. Вот почему невероятно важно правильно распределять свое время и управлять им.

Многие люди оправдывают свои неудачи или отсутствие попыток к действию нехваткой времени. Однако большинство из них даже не пробовало контролировать этот ресурс. Чтобы успевать выполнять все необходимые задачи и добиваться своих целей, нужно грамотно управлять своим временем и организовывать свою жизнь.

1. Постановка задачи

Для управления временем нужно правильно его планировать. План лучше никогда не держать в голове, а записывать на бумаге или в компьютере.

Целью данной работы и является разработка приложения, позволяющего пользователю создавать цели и задачи, планировать их выполнение в календаре, а так же анализировать выполненные или невыполненные задачи, для получения наглядной картинки своего развития в различных областях.

Почему это приложение будет полезно? Ведь многие думают, что вполне справляются, запоминая свои цели на день. Как показывает практика это заблуждения. Так как, полагаясь только на свою память, люди перестают замечать, на что тратят время. Цель, которая не

сформулирована четко и конкретно – это только мечта, которая вряд ли сбудется. К тому же хорошо известно правило 10/90, суть которого в том, что если вы затратите 10% времени на планирование задачи, до ее выполнения, то это может сэкономить до 90% при решении этой задачи.

Таким образом, опираясь на данные суждения, целью приложения является стремление помочь пользователю в формировании четких целей и задач, на различное направление деятельности (например, бизнес, отношения), а так же их дальнейший анализ.

2. Архитектура приложения

Проект было решено разрабатывать в виде клиент-серверного веб-приложения. Клиентом в данном случае выступает браузер пользователя, а сервером – сервер приложений. Браузер посылает запросы на сервер, который в свою очередь имеет возможность работать с базой данных. На рис. 1 представлена обобщенная схема архитектуры приложения.

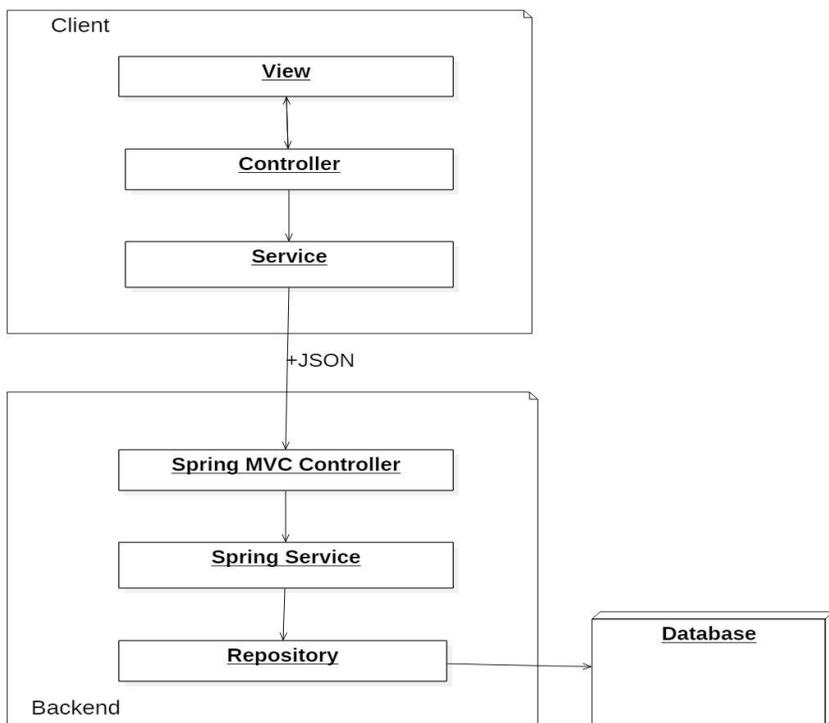


Рис. 1. Архитектура приложения

Для создания клиентской части был выбран AngularJS – JavaScript-фреймворк, направленный на создание SPA (Single Page Application – Одностраничное приложение) [1]. Серверная часть основана на Spring – фреймворке для Java-платформы, который предназначен для создания сложных корпоративных бизнес-приложений [2]. В качестве системы управления базой данных была выбрана MySQL.

Spring и AngularJS вместе образуют продуктивный и удобный стек технологий для разработки веб-приложений [3].

Уровень представления (View) в AngularJS состоит из шаблонов HTML, CSS таблиц, а так же т.н. директив Angular. С помощью директив можно добавлять новое поведение существующим HTML элементам, можно создавать новые компоненты.

Уровень представления запрашивает данные для отображения на странице у контроллеров (Controller). Контроллеры управляют данными для представления. AngularJS позволяет модели на уровне представления автоматически обновлять содержание страницы, без перезагрузки этой страницы, и наоборот, реагировать на изменения модели [1].

Контроллеры получают данные с уровня сервисов (Service). Сервисы взаимодействуют с приложением на сервере при помощи HTTP-запросов, в которых передается JSON-файл, содержащий необходимую информацию.

Эта информация на стороне backend запрашивается у Spring контроллеров (Spring MVC Controller). Эти контроллеры обрабатывают входящие по технологии RESTful HTTP запросы, после чего отдают ответ.

Spring контроллеры в свою очередь получают информацию из Spring сервисов (Spring Service). Spring сервисы предназначены для выполнения всей бизнес-логики приложения.

Следующим уровнем являются т.н. репозитории (Repository). Эти Java-классы необходимы для работы с базой данных, а именно для извлечения данных из таблиц, добавления новой информации или удаления ненужной [2].

Так же для корректной работы приложения были использованы такие библиотеки, как Material design, Bootstrap, Quartz, Highcharts, D3 [4].

3. Функционал приложения

Итак, основная цель приложения обеспечить удобный интерфейс для планирования времени и жизни пользователя. Таким образом, можно выделить основные функции приложения:

- Возможность создания цели;
- Возможность создания задачи;

- Возможность каждую цель декомпозировать на несколько других целей;
- Возможность каждую задачу декомпозировать на несколько других задач.

Представленный функционал позволяет грамотно создавать цели и задачи, а так же декомпозировать их.

Так же другая основная цель проекта – помочь пользователю определять, в каких направлениях он лучше развивается, какая у него продуктивность и т.п. Таким образом, можно сформулировать следующие функциональные требования:

- Предоставление графиков активности по дням, неделям, месяцам (рис. 2);
- Предоставление графиков по типу выполненных целей и задач;
- Предоставление графиков по различным направлениям деятельности.

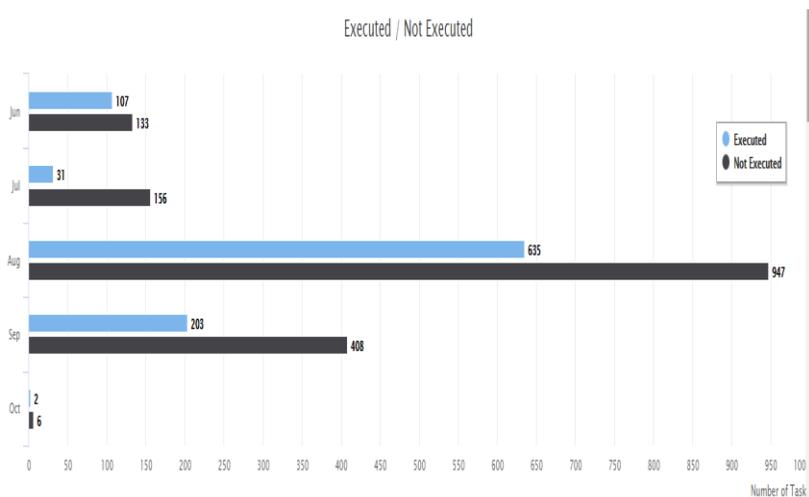


Рис. 2. Пример графика

Следующий функционал необходим для обеспечения удобства работы с целями и задачами:

- Представление целей и задач в виде графа;
- Возможность получать уведомления перед началом предстоящей задачи;
- Возможность просмотреть в календаре запланированные задачи;
- Возможность оставлять комментарии к задаче или цели.

4. Интерфейс пользователя

Одним из условий того, что клиенты будут использовать данное приложение, это предоставление удобного, современного, адаптивного дизайна.

Адаптивным называется дизайн, который обеспечивает корректное отображение веб-страниц на устройствах, имеющих различные разрешения экрана. Целью адаптивного веб-дизайна является универсальность отображения содержимого сайта. Таким образом, не нужно создавать отдельные версии веб-сайта для различных типов устройств. Один сайт будет корректно работать и отображаться на компьютере, планшете или смартфоне.

Адаптивность достигается, с помощью таких средств как медиа-запросы в CSS, Material design и Bootstrap.

На рис. 3 представлен пример одной из страниц приложения.

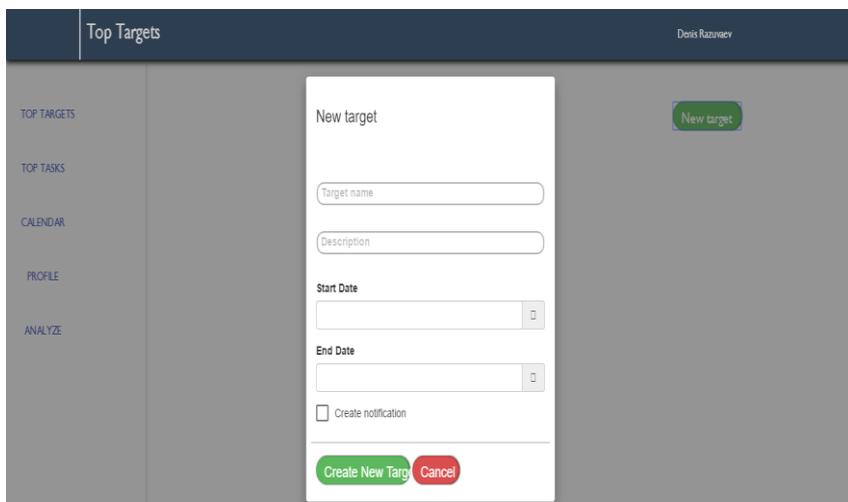


Рис. 3. Пример страницы приложения

Заключение

Данная статья посвящена описанию проблемы планирования управления временем, а так же путям их решения. В итоге было получено клиент-серверное веб-приложение, на базе таких технологий как AngularJS и Spring. Описываемый проект обладает обширным функционалом для планирования времени, путем создания целей и задач, а так же анализом эффективности выполнения задач в различных дисциплинах.

Список литературы

1. Angular [Электронный ресурс] : сайт фреймворка. – Режим доступа : <https://angularjs.org>
2. Spring [Электронный ресурс] : сайт фреймворка. – Режим доступа : <http://projects.spring.io/spring-framework>
3. Хабрахабр [Электронный ресурс] : новостной журнал. – Режим доступа : <https://habrahabr.ru>
4. D3 JS [Электронный ресурс] : сайт библиотеки. – Режим доступа : <https://d3js.org>

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ ПРЕДСКАЗАНИЯ ОТТОКА КЛИЕНТОВ С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Е.В. Резниченко

Студент

М.Г. Матвеев

Профессор

Введение

Анализ данных – это исследования, связанные с обчетом многомерной системы данных, имеющей множество параметров [1]. В общем смысле анализ данных – это процесс исследования, фильтрации, преобразования и моделирования данных с целью извлечения полезной информации и принятия решений. В процессе анализа исследователь производит совокупность действий с целью формирования определенных представлений и характере явления, описываемого этими данными. Как правило, для анализа данных используются различные математические методы.

Анализ данных нельзя рассматривать только как обработку информации после её сбора. Прежде всего, анализ данных – это средство проверки гипотез и решения задач исследования. Особый интерес к анализу данных и его методов, получивших широкое применение во многих сферах, возник в связи с развитием средств сбора и хранения данных, позволившим накапливать большие объемы информации.

Проблемы бизнес анализа формулируются по-другому, но решение большинства из них сводится к той или иной задаче интеллектуального анализа данных или к их комбинации.

Одной из важнейших задач, имеющих значение для компаний, является задача прогнозирования оттока клиентов. Показатель оттока клиентов – это бизнес-термин, описывающий интенсивность покидания компании клиентами [2]. Это ключевой показатель компании, так как приобретение новых клиентов обходится намного дороже, чем удержание действующих. Следовательно, понимание того, как поддерживать заинтересованность клиентов представляет собой логический фундамент для разработки стратегий их удержания. В итоге, предприятия стремятся получить более совершенные технологии выявления возможного ухода клиентов.

1. Цель эксперимента

Цель эксперимента в данной статье – создать программное средство, позволяющее визуализировать поведение клиента и прогнозировать его вероятный уход из компании. Данная разработка позволит компании вовремя реагировать и поддерживать заинтересованность клиента, в результате чего произойдет укрепление отношений и противодействие миграции.

Основными задачами эксперимента являются:

- Подготовка набора данных для обучения;
- Исследование и преобразование данных для обучения;
- Выбор алгоритма машинного обучения с лучшим результатом;
- Проектирование и разработка программного средства.

2. Описание используемых данных

Для проведения эксперимента используются две выборки данных по клиентам: обучающая за период с января 2015 по июнь 2016 (18 месяцев) и тестовая за период с июля 2016 по декабрь 2016 (6 месяцев).

Общее количество уникальных клиентов, данные которых используются в исследовании: 500 с 16 признаками. В выборки не входят данные по новым клиентам за рассматриваемый период времени.

Общая статистика по составу обучающей и тестовой выборки представлена в табл. 1.

Таблица 1

Статистика по составу обучающей и тестовой выборки

Выборка	Период	Общее количество объектов	Количество уникальных клиентов	Количество ушедших клиентов	Количество клиентов на начало периода	Количество клиентов на конец периода
Обучающая	01.2015 – 06.2016	8712	500	48	500	452
Тестовая	07.2016 – 12.2016	2661	452	17	452	435

3. Выбор методов машинного обучения

Исходя из анализа работ по теме, были выделены несколько перспективных методов машинного обучения: нейронные сети, логистическая регрессия, random forest, градиентный бустинг над решающими деревьями.

Использовалась обучающая выборка для обучения моделей. На тестовой выборке модели делают предсказание и оцениваются по заданным метрикам. Лучшая модель выбиралась по результатам предсказания на тестовой выборке.

Результаты экспериментов представлены в табл. 2.

Таблица 2

Сравнение моделей на тестовой выборке

Модель	precision	recall	F1	F0.5	AUC
Нейронные сети	0.69	0.59	0.63	0.66	0.825
Логистическая регрессия	0.60	0.37	0.46	0.52	0.839
Random forest	0.72	0.64	0.66	0.68	0.842
Градиентный бустинг	0.70	0.60	0.65	0.69	0.832

Итоговые сравнения классификаторов производились с помощью AUC метрики. Лучшие результаты показала модель random forest.

Random forest (случайный лес) – это один из самых универсальных методов машинного обучения, предназначенный для решения задач классификаций и регрессии. Алгоритм является модификацией метода бэггинг (bootstrap aggregating) [3]. Одним из основных преимуществ метода является встроенная оценка качества.

Особенности алгоритма Random forest заключаются в использовании комитета деревьев и наличием встроенной оценки качества предсказания.

Для предсказания ухода клиента в будущем использовался алгоритм прогнозирования «линейная регрессия».

4. Пример использования приложения

После получения необходимых данных и выбора лучшей модели регрессора и алгоритма прогнозирования было разработано приложение.

При запуске приложения появляется начальная форма приложения со вкладкой «Загрузка данных» (рис. 1).

На вкладке «Загрузка данных» пользователь может совершать такие действия как: выбор необходимого месяца для обновления данных, выбор одного или несколько клиентов для обновления данных в базе, загрузка файла с новыми данными по клиентам.

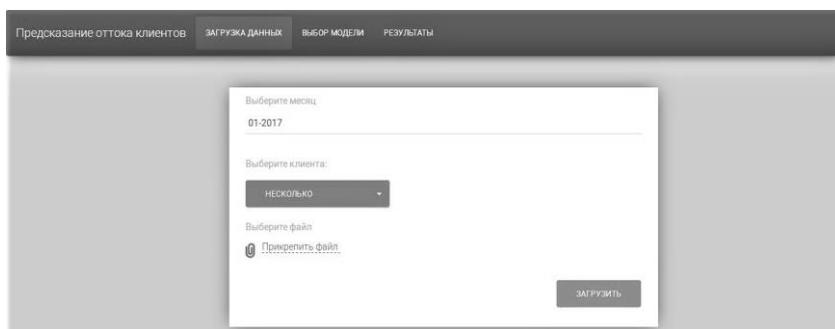


Рис. 1. Форма использования приложения

После заполнения данных нужно нажать кнопку «Запустить».

Результатом данного действия будет являться переход на вкладку «Выбор модели» (рис. 2), где пользователю будет доступен выбор модели классификатора и выбор модели прогнозирования.

После заполнения данных нужно нажать кнопку «Запустить».

Результатом данного действия будет являться переход на вкладку «Результаты» (рис. 3), где пользователь получает график, по которому можно отследить тенденцию поведения клиента и предсказанное значение степени отнесения клиента к ушедшим.



Рис. 2. Форма использования приложения



Рис. 3. Результаты использования приложения

Заключение

В рамках данной работы были проанализированы и обработаны исходные данные о клиентах, сравнены результаты моделей машинного обучения на данных, спроектировано и разработано программное средство для прогнозирования оттока клиентов.

Таким образом, разработанный продукт позволит выявлять возможность ухода клиента, что даст возможность компании вовремя реагировать и поддерживать заинтересованность клиента. В результате таких действий произойдет укрепление отношений и противодействие оттоку, а, следовательно, повысится прибыль компании.

Список литературы

1. Паклин Н.. Визуализация данных // Бизнес-аналитика. От данных к знаниям / Н. Паклин, В. Орешков. – СПб. : Питер, 2013. – 693 с.

2. Барсегиан А. А. Методы и модели анализа данных: OLAP и Data Mining / А. А. Барсегиан, М. В. Куприянов. – СПб. : БВХ-Петербург, 2004. – 336 с.
3. Random forest [Электронный ресурс] : Статья из википедии. – Режим доступа : https://ru.wikipedia.org/wiki/Random_forest

ЭКСПЕРТНАЯ СИСТЕМА «АГРОКОНСУЛЬТАНТ» НА ОСНОВЕ АППАРАТА НЕЧЕТКОЙ ЛОГИКИ

И.Ю. Савченко
Студент
М.Г. Матвеев
Профессор

Введение

Как известно – сельскохозяйственная (с\х) промышленность является одной из основных в России. Предприятия и организации, работающие в данной отрасли, размещены более чем в 30 регионах нашей страны.

В с\х промышленности существует огромное количество производственных процессов, для которых сложно обеспечить точное управление. Они обычно являются многомерными, нелинейными и зависят от многих факторов.

Данная работа направлена на совершенствование процесса принятия управленческих решений в области сельского хозяйства.

Аналитическая часть

Все процессы в сельском хозяйстве обусловлены той или иной технологией, вне зависимости от их типа, будь то посадка культуры, обработка земельных угодий или др.

Технологии включают в себя наборы критериев и условий, соблюдение которых влечет за собой возможность или невозможность применения.

Для принятия управленческих решений, касающихся выбора технологии, необходим эксперт в этой области, который сможет объективно оценить все условия и указать каким типом технологий необходимо воспользоваться в той или иной ситуации.

К сожалению, большинство предприятий с\х не имеют в своем штате подобных сотрудников, в следствии чего возникает потребность в привлечении сторонних консалтинговых организаций.

Основными недостатками такого подхода являются:

- Высокая вероятность ошибочного принятия решения консультантом при выполнении обработки данных клиента;
- Длительное время получения результата;
- Высокая стоимость подобной услуги.

Реализация

Описанные выше проблемы могут быть решены путем разработки экспертной системы, которая основывается на аппарате нечеткой логики [1, 2], в частности, использует алгоритм Мамдани [3, 4]. Общая схема системы представлена на рис. 1.

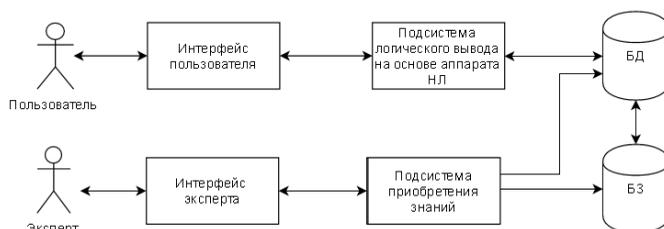


Рис. 1. Общая схема экспертной системы

Основными компонентами системы являются:

- База данных (БД);
- База знаний (БЗ);
- Подсистема приобретения знаний;
- Подсистема логического вывода;
- Интерфейс эксперта;
- Интерфейс пользователя.

Формирование БЗ заключается в ее наполнении правилами вида (листинг):

Листинг

Формирование базы знаний

П1: ЕСЛИ x ЕСТЬ A_1 И y ЕСТЬ B_1 , ТО z ЕСТЬ C_1 ,

П2: ЕСЛИ x ЕСТЬ A_2 И y ЕСТЬ B_2 , ТО z ЕСТЬ C_2 ,

П3: ЕСЛИ x ЕСТЬ A_3 И y ЕСТЬ B_3 , ТО z ЕСТЬ C_3 ,

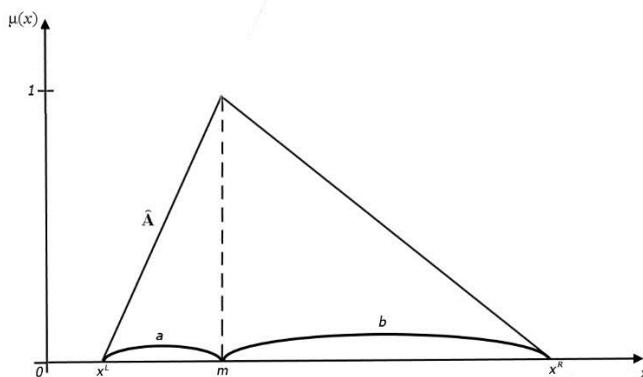
П4: ЕСЛИ x ЕСТЬ A_4 И y ЕСТЬ B_4 , ТО z ЕСТЬ C_4 ,

...

ПN: ЕСЛИ x ЕСТЬ A_N И y ЕСТЬ B_N , ТО z ЕСТЬ C_N ,

где x и y – имена входных переменных, z – имя переменной вывода, $A_1, A_2, B_1, B_2, C_1, C_2$ – некоторые заданные триангулярные функции

принадлежности (рис. 2) которым в соответствие ставятся лингвистические переменные.



m – мода, a x^L , x^R – левая и правая границы носителя X соответственно

Рис. 2. Триангулярная функция принадлежности нечеткого высказывания и ее параметры

На основе описанного подхода по формированию БЗ была разработана подсистема приобретения знаний, которая включает в себя возможность внесения новых знаний(правил), внесение данных в БД (наименование правила, диапазоны значений функций принадлежности и т.д.) и формирование шаблонов запросов для пользователей.

Пример продукционного правила из системы:

«ЕСЛИ влажность почвы ЕСТЬ хорошая И глубина посева ЕСТЬ неглубокая И климат ЕСТЬ теплый И плотность посева ЕСТЬ плотная, ТО с\х культура пшеница»;

Подсистема логического вывода является инструментом пользователя, позволяющая на основе введенных им значений выдавать результат, который может стать основой для принятия управленческого решения. Структурная модель данной подсистемы изображена на рис. 3.

Принцип действия указанной подсистемы следующий, после того, как пользователь ввел данные, запускается механизм их обработки.

Первым этапом в данной цепочке является этап фаззификации.

На этом этапе происходит определение (введение) нечёткости. Каждому конкретному значению отдельной входной переменной системы нечёткого управления ставится в соответствие значение функции принадлежности соответствующего ей терма входной лингвистической переменной [5].

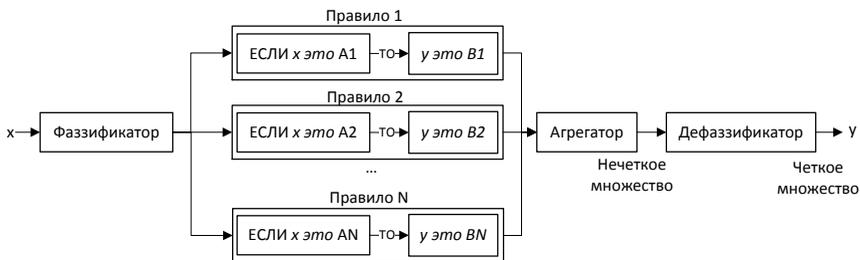


Рис. 3. Структурная модель нечеткого логического вывода

Далее значение функции принадлежности по каждому правилу (1):

$$\mu_{\bar{A}}(x) = \begin{cases} \mu_{\bar{A}}^L = \frac{x - x^L}{m - x^L}; x \in [x^L; m], \\ \mu_{\bar{A}}^R = \frac{x - x^R}{m - x^R}; x \in [m; x^R], \\ 0, \text{ в остальных случаях} \end{cases} \quad (1)$$

где m – мода, x – введённое значение пользователя, а x^L , x^R – соответственно, левая и правая границы носителя x .

После этого полученные значения функции по каждому из продукционных правил агрегируются:

$$\mu_i(x) = \max\{\mu_1(x), \dots, \mu_n(x)\} \quad (2)$$

где $\mu_1(x)$, $\mu_2(x)$ — функции принадлежности объединяемых множеств.

И последний этап, это этап дефаззификации, иными словами процесс перехода от функции принадлежности выходной лингвистической переменной к её четкому (числовому) значению [5].

$$y = \frac{\int_{\min}^{\max} x \mu(x) dx}{\int_{\min}^{\max} \mu(x) dx}, \quad (3)$$

где переменная y – результат дефаззификации; x – переменная, соответствующая выходной лингвистической переменной и принимающая значения от $x = \min$ до $x = \max$; \min и \max – левая и правая точки интервала носителя нечёткого множества; $\mu(x)$ – функция принадлежности нечёткого множества.

Численное значение, полученное на выходе интерпретируется в ответ пользователю.

Заключение

Итогом данной работы является экспертная система принятия управленческих решений, позволяющая, на основе предоставленных пользователем данных определять, какую с\х технологию необходимо применять, сократить время получения результата обработки данных и снизить затраты на услуги консалтинговых организаций.

Список литературы

1. Матвеев М.Г. Модели и методы искусственного интеллекта. Применение в экономике. / М.Г. Матвеев, А.С. Свиридов, Н.А. Алейникова. – М. : Финансы и статистика, Инфра-М, 2008. – 448 с.
2. Михайлов В.В. Формализация задач повышения качества управления в условиях стохастичности / В.В. Михайлов, М.Г. Матвеев. – Системы управления и информационные технологии. – 2005. – Т.22. – №5. – С.49-52.
3. Авернин А.Н. Нечеткие множества в моделях управления и искусственного интеллекта / А.Н. Авернин, И.З. Батыршин. – М. : Мир, 1986. – 312 с.
4. Кофман А. Введение в теорию нечетких множеств / А. Кофман. – М. : Радио и связь, 1982.– 432 с.
5. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику [Электронный ресурс]. – Режим доступа : <http://matlab.exponenta.ru/fuzzylogic/book1/>

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ МЕТОДА АНАЛИЗА ИЕРАРХИЙ ДЛЯ ОПТИМИЗАЦИИ ВЫБОРА ПАРТНЕРОВ КОМПАНИИ

А.Н. Семенова

Студент

И.В. Абрамов

Доцент

Введение

Развитие информационных технологий позволяет подойти более осознанно к принятию важных управленческих решений. Выбор компетентных партнеров компании является одним из критериев её успешности. В реальности очень тяжело сделать этот выбор эффективно, учитывая все плюсы и минусы партнеров. В этом может помочь система поддержки принятия решений – СППР. При помощи СППР можно

© Семенова А.Н., Абрамов И.В., 2017

выполнить сбор и анализ информации, непосредственно влияющей на процесс принятия решений.

В СППР с помощью метода анализа иерархии (МАИ) можно выявить варианты, являющиеся наиболее оптимальными, отвечающими требованиям к решению проблемы и самой её сути. Данный метод используется в ситуациях, когда основная информация основана на мнении экспертов. МАИ помогает структурировать проблему в виде иерархии и выполнить сравнение вариантов принятия решения на основе приоритетов.

В данной работе рассматривается проблема выбора партнеров компании, которая может быть эффективно решена с помощью метода анализа иерархий.

1. Описание деятельности компании

Деятельность нашей компании основана на продаже услуг в игре World of Warcraft. Продажа происходит через интернет-магазин, в котором представлен каталог предоставляемых услуг. Компания ведет сотрудничество со сторонними людьми, являющимися исполнителями заказов. При поступлении заказа менеджер, являющийся сотрудником, выполняет поиск исполнителей, способных выполнить заказ. Для компании данные лица являются партнерами. После поступления заказа, менеджеру организации требуется его обработать. Обработка обычно включает в себя общение с клиентом по поводу деталей, предварительный поиск исполнителей, контроль проведения оплаты, окончательный выбор исполнителей, контроль выполнения заказа и его завершение. Блок-схема обработки заказа представлена на рисунке.

Общения с клиентом по поводу деталей заказа. Обычно, попав на сайт, у потенциальных клиентов появляются вопросы по поводу услуги, которую они хотели бы приобрести или, например, по поводу конкретного времени и условий выполнения заказа. Изначально диалог ведется через онлайн-чат, размещенный на сайте. Именно на данном этапе клиент узнает всю интересующую его информацию. Иногда данный этап пропускается и происходит переход к контролю проведения оплаты. Однако обычно клиенту требуется некоторая дополнительная информация, поэтому очень важно иметь менеджера, обладающего хорошими коммуникабельными навыками и обладающего достаточной компетенцией.

Этап предварительного поиска исполнителей. Большинство клиентов интересуются конкретным временем выполнения заказа и только потом производят его оформление. Исполнители имеют расписание, согласно которому могут сообщить о своей возможности взять заказ. Общение с исполнителями происходит через skype-конференцию, где менеджер описывает общую информацию о заказе. Исполнители в свою очередь сообщают ему в личной переписке о возможности выполнения. Также

предварительно обговаривается стоимость и основные детали заказа. После общения с исполнителем, менеджер доводит информацию о возможном времени исполнения до клиента.

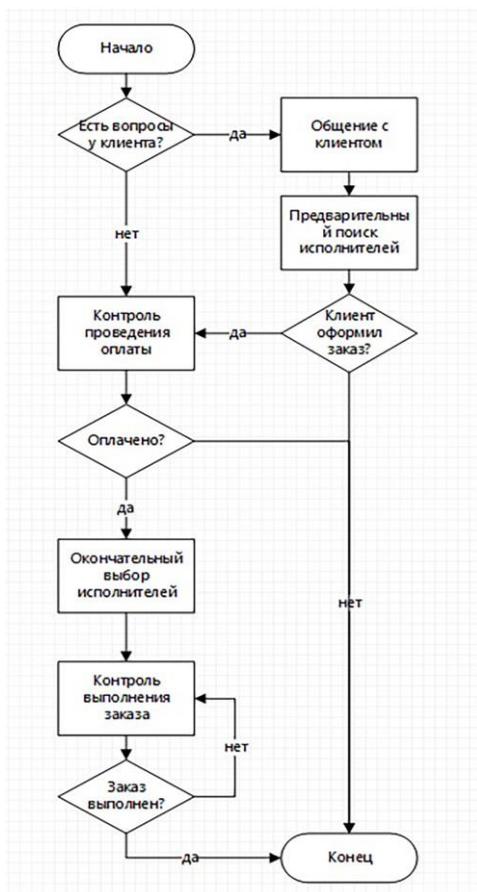


Рисунок. Блок-схема обработки заказа

Если клиенту подходит время, то он производит оформление заказа. На данном этапе требуется выполнить контроль проведения оплаты. Менеджер при необходимости помогает клиенту с оформлением заказа и проверяет поступление средств от данного клиента. Дальнейшее общение с клиентом происходит через программы Skype, WhatsApp или онлайн-чат при отсутствии предыдущих.

После оплаты необходимо договориться с исполнителями о брони конкретного времени для клиента и окончательно прописать все требования к заказу. Данный этап является очень важным для предотвращения возникновения путаницы с деталями заказа.

Во время контроля выполнения заказа требуется проследить за исполнением всех оговоренных критериев. В случае успешного окончания можно перейти к стадии завершения заказа.

На данном этапе происходит обратная связь с клиентом. Таким образом, заказ можно считать завершенным.

Главной целью компании помимо получения прибыли является помощь игрокам в эффективном времяпрепровождении внутри игры.

Основными клиентами данной компании являются иностранцы, которым требуется помощь в совершенствовании своих навыков в игре Word of Warcraft.

2. Проблемы при взаимодействии с клиентами

Основные проблемы компании при взаимодействии с клиентами:

– Отсутствие информации о длительности выполнения заказа или долгое выполнение некоторых услуг клиентам:

- Недоверие со стороны клиентов;
- Желание клиентов получить больше за меньшие деньги;
- Опоздания клиентов;
- Наличие мошенников;
- Трудности нахождения и выбора хороших исполнителей.

Содержание основных проблем следующее.

Отсутствия информации о длительности выполнения заказа или его долгое выполнение. Данная проблема связана с тем, что разные клиенты имеют различный прогресс игрового персонажа и с тем, что в игре присутствует элемент случайности. Некоторые сложные заказы могут выполняться не одну неделю, поэтому многие клиенты могут отказываться от выбора подобных услуг. Также усложняется процесс исполнения заказа для менеджера. Это происходит из-за того, что не все исполнители готовы взяться за выполнение длительного заказа. Также менеджеру приходится контролировать и оповещать клиента об этапах выполнения.

Недоверие со стороны клиентов может возникать из-за недобросовестной деятельности сторонних компаний. Многие компании, которые также предоставляют услуги в игре Word of Warcraft, делают это некачественно. В результате страдает не только их деятельность, но и возникает недоверие к другим компаниям. Для борьбы с данным недоверием используются отзывы реальных клиентов, разрабатывается политика возвратов и многое другое.

Естественным является желание клиентов получить большее за меньшие деньги. Однако это достаточно часто вызывает проблемы. Большинство клиентов не могут в полной мере оценить, сколько времени и сил тратят менеджеры и исполнители на обработку заказа. В связи с этим клиенты хотят получить несоизмеримо большую скидку по сравнению с объемом заказа. Поэтому менеджер должен обладать достаточной компетенцией для того, чтобы объяснить из чего складывается данная стоимость и договориться о ней.

В любой работе с людьми присутствует так называемый человеческий фактор. Тем самым может возникнуть проблема опоздания клиентов, которая влечет за собой задержки в исполнении заказов. Данные задержки несут в себе потери средств исполнителей, так как они могли бы взять на место опоздавшего другого человека. Также они отрицательно влияют на другие заказы, которые могут выполняться этими исполнителями параллельно: опоздавшего приходится ждать не только исполнителям, но и другим клиентам. Немаловажным является то, что сроки заказа сдвигаются, в связи с чем теряется возможность оформления данным клиентам ещё одного заказа, так как чаще всего пока исходный не исполнен, клиент не создает новый.

Почти все компании тем или иным образом сталкиваются с мошенниками. В данном случае люди производят покупку с краденных интернет-кошельков и уже после исполнения заказа истинный владелец производит возврат денежных средств в связи с кражей. Однако это происходит уже после оплаты исполнителям, в результате чего теряются немалые денежные средства и время. Описанная проблема требует особой внимательности менеджеров для её предотвращения.

Чаще всего заказ выполняется не одним человеком, а командой людей, имеющих представителя. Именно с ним менеджер договаривается о всех деталях заказа. Проблема, связанная с трудностями нахождения и выбора хороших исполнителей, включает в себя не только плохое исполнение заказа, но и некомпетентных представителей команд-исполнителей.

Существуют представители, с которыми невозможно договориться о чем-либо. В связи с этим приходится либо отказываться от работы с ними, либо отдавать им заказы исключительно в крайних случаях.

Что касается проблем с плохим исполнением заказа, то они могут быть выражены в:

- желании исполнителей взять одного клиента в ущерб другому, тем самым получить больше прибыли;

- непродуманном составе команды. В результате чего некоторые заказы могут иметь более длительный срок выполнения или могут быть в принципе не выполнены;
- задержках времени начала исполнения. В данной ситуации, в лучшем случае приходится договариваться о новом времени, а в худшем случае клиент может запросить возврат денежных средств;
- недопонимании между менеджером и представителем команды при обсуждении деталей заказа;
- переманивании клиентов командами.

Данная статья посвящена проблеме, связанной с трудностями нахождения и выбора хороших исполнителей. Её можно решить при помощи метода анализа иерархий. Данный метод основан на декомпозиции проблемы на составные части и обработку точек зрения лицом, отвечающим за принятие решений. Таким образом, каждая альтернатива приобретает свою значимость и определяется наилучшее решение, для которого рассчитанный критерий будет максимальным.

3. Использование метода анализа иерархий для решения проблемы выбора партнеров компании

Для использования метода анализа иерархий как метода решения проблемы выбора партнеров требуется учесть:

- Стоимость исполнения заказа;
- Время исполнения заказа;
- Качество исполнения заказа;
- Надежность исполнителей;

Рассмотрим каждый пункт немного подробнее.

При выборе партнера стоимость исполнения заказа играет большую роль. Менеджер должен уметь договариваться с исполнителями на наиболее выгодную для компании стоимость. В условиях сложившейся конкуренции между партнерами, это вполне реально сделать. Поэтому естественным является предпочтение команды, предоставляющей более выгодные условия перед остальными аналогичными.

Под временем исполнения заказа в данном случае понимается время начала выполнения. Оно в основном зависит от клиента, который выбирает наиболее удобный ему промежуток. Таким образом, приходится отказываться от некоторых исполнителей для создания комфорта клиенту.

Под качеством [1] исполнения заказа понимается соответствие завершенного заказа исходным требованиям и ожиданиям клиента. В нашем случае, исполнитель должен предоставить услугу в соответствии со всеми купленными клиентом опциями. Также если не все, то хотя бы большинство ожиданий клиента должны оправдаться и полученный опыт игры

должен превзойти ожидания клиента. Таким образом, и клиент будет доволен и менеджер предпочтет данного исполнителя другим.

Надежность [2] предполагает успешное выполнение заказа в течении заданного периода времени при наличии определенных требований касательно длительности выполнения работ. Стоит отметить, что надежность является одним из понятий качества, но в более узком смысле. Поэтому следует четко понимать различия между надежностью и качеством.

Если качество представляет собой некие свойства получаемой услуги, которые напрямую связаны с ожиданиями клиента, тем самым происходит удовлетворение его потребностей, то надежность должна соотноситься с требованиями заказа, она не связана с ожиданиями клиента, то есть не связана с его субъективным мнением.

Полученный список можно использовать в качестве критериев для выбора партнеров компании, тем самым можно применить метод анализа иерархий [3].

Заключение

Данная статья посвящена методу анализа иерархий для помощи компании при выборе партнеров. Представлено описание компании для лучшего понимания её деятельности. Рассмотрены основные проблемы при взаимодействии с клиентами и была выделена основная проблема: трудности при нахождении и выборе хороших исполнителей (партнеров). Представлен список пунктов, которые необходимо учесть для решения проблемы при помощи метода анализа иерархий. Получен результат: данный список будет являться критерием для выбора партнеров.

Использование МАИ позволяет системно подойти к решению поставленной проблемы, что может значительно сократить как время принятия решения, так и силы, затраченные на его поиск.

Список литературы

1. Свободная энциклопедия Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Качество>
2. Свободная энциклопедия Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Надежность>
3. Саати Т. Принятие решений. Метод анализа иерархий / Т. Саати; пер. с англ. – М. : Радио и связь, 1989. – 316 с.

ОБРАБОТКА СИГНАЛОВ BLE-УСТРОЙСТВ НА БАЗЕ ПРОФИЛЯ FIND ME В ANDROID-ПРИЛОЖЕНИЯХ

В.А. Соколова

Студент

А.Л. Гавшин

Ассистент

Введение

Каждый из нас когда-нибудь терял ключи, кошелек или другие ценные вещи. И если на телефон еще можно позвонить, то поиск других предметов значительно осложняется. Данная работа описывает ключевые аспекты работы android-приложения, взаимодействующего с поисковыми маяками iTrack. Функционирование этих маяков основано на технологии Bluetooth Low Energy (BLE). Они имеют малые габаритные размеры и легко прикрепляются к любым предметам. Это позволяет, принимая сигнал от маяков, выяснять местоположение этих предметов, инициировать подачу звукового сигнала на них.

1. Bluetooth Low Energy

Bluetooth Low Energy (BLE) – это беспроводная технология передачи информации с малым потреблением энергии и значительно увеличенным сроком службы от аккумулятора. BLE впервые вошел в протокол Bluetooth 4.0. На данный момент уже вышла спецификация 5.0. Устройства BLE связываются друг с другом лишь при необходимости отправки или получения информации, а после засыпают, этим и обуславливается низкое потребление энергии.

Такие устройства могут быть как двухрежимные (Bluetooth Smart Ready), совместимые с классическими Bluetooth-устройствами, так и однорежимные (Bluetooth Smart), совместимые только с BLE. Протокол BLE строго структурирован по принципу своей коммуникации с другими устройствами. Вначале поисковый маяк изучает доступные сервисы для отправки/принятия данных; неотъемлемая часть этих сервисов – их характеристики, определяющие тип данных для будущей передачи. Характеристики, из соображений наглядности, могут иметь в своём составе описания-дескрипторы, которые помогают определить тип данных.

2. BLE и Android

Рассмотрим, что платформа Android предлагает для работы с BLE устройствами. В операционной системе Android поддержка BLE устройств появилась с версии Android 4.3 Jelly Bean (API 18), когда в состав системы

вошёл мощный API [1]. Он позволяет производить сканирование окружающего пространства на предмет наличия готовых к соединению устройств, передавать данные между устройствами и многое другое. Bluetooth API пережил серьезные изменения с версии Android 5.0 Lollipop (API 21). Были введены новые методы, а прежние стали считаться устаревшими, но новая версия не содержит обратной совместимости, поэтому, если вы хотите поддерживать устройства на более ранних системах, приходится писать две версии кода.

Работа с Bluetooth состоит из четырех этапов:

- установка настроек bluetooth адаптера;
- поиск доступных для соединения устройств;
- установка соединения;
- передача данных.

3. Роли и обязанности при взаимодействии Android с BLE-устройством

Роли и обязанности, которые возможны при общении Android-устройства с BLE-устройством

- Центральная/периферическая роль.

Относится к самому BLE-соединению. Устройство в центральной (ведущей) роли сканирует, ищет объявления, а устройства в периферийной (ведомой) роли создаёт объявления.

- GATT-сервер/GATT-клиент.

Определяет, каким образом два устройства общаются друг с другом, когда они установили связь. Представьте, что у вас есть Android-смартфон и поисковый маяк. Смартфон поддерживает центральную роль; поисковый маяк поддерживает периферийную роль. Чтобы установить BLE-соединение, вам нужно по одному устройству, поддерживающему каждую из ролей; два периферийных устройства не смогут общаться друг с другом, также, как и два центральных. Как только телефон и поисковый маяк наладили связь, они начинают передачу метаданных GATT (Generic Attribute Profile) друг другу. Например, если маяк хочет сообщить данные о нажатии кнопки или заряде батареи, маяку имеет смысл работать в качестве сервера. Представим другую ситуацию: у нас есть музыкальная колонка, которая умеет только играть музыку, предоставляемую телефоном. При таком соединении в качестве сервера имеет смысл использовать телефон как устройство, генерирующее данные.

4. Find Me профиль

Профиль Bluetooth-устройства определяет набор сервисов, доступных для этого устройства. Для совместной работы Bluetooth-устройств необходимо, чтобы все они поддерживали общий профиль. Каждое

устройство может реализовывать несколько профилей, их можно рассматривать как контракты между устройствами поддерживать определенные функции. Профиль Find Me определяет поведение при нажатии кнопки на одном устройстве и подачу сигнала оповещения на сопряженном (рис. 1). Он может использоваться для нахождения устройства по звуковому сигналу. Find Me определяет две роли: Find Me Locator (должен быть клиентом GATT) и Find Me Target (должен быть сервером GATT) [2].

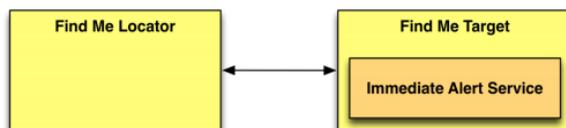


Рис. 1. Взаимодействие между службами и двумя ролями профиля

5. Сервисы и характеристики iTrack

Для работы с BLE-устройствами необходимо читать или записывать значения в определенные характеристики сервисов. Подробное описание возможных значений характеристик для устройств iTrack, которые мы использовали, представлены в табл. 1 и 2 [3, 4].

Таблица 1

Список характеристик iTrack

Название характеристики	Read OrWrite	Services ID	CharacteristicID	Длина	Значения
Paring mode (Режим сопряжения)	ReadOnly	0xFFFF0	0xFFFF5	1	0x1: paring mode 0x0: other mode
Key press (Нажатие кнопки)	Notification	0xFFE0	0xFFE1	1	0x01: Enable Key notification 0x02: long Key press notification 0x08: short Key press notification

Окончание таблицы 1

Название характеристики	Read OrWrite	Services ID	CharacteristicID	Длина	Значения
Close iTrack/disable alert (Закрыть соединение / Отключить оповещения)	Write	0x1803	0x2a06	1	0x03: close iTrack
Call iTrack (Вызвать iTrack)	Write	0x1802	0x2a06	1	0x02: start alert on iTrack 0x00 disable alert on iTrack

Таблица 2

Описание характеристик iTrack

Название характеристики	Описание
Paring mode (Режим сопряжения)	Используется для определения находится ли iTrack в режиме сопряжения. Значение по умолчанию – 0x0. ITrack установит это значение в 0x1 после длительного нажатия на кнопку, и установит это значение обратно в 0x0 через 30 секунд
Key press (Нажатие кнопки)	<ol style="list-style-type: none"> 1. Приложение устанавливает значение в 0x01 для включения уведомления о нажатии 2. После этого iTrack будет посылать код 0x02, при долгом нажатии на кнопку (> 3 секунды) на нем 3. Или 0x08, при коротком нажатии

Название характеристики	Описание
Close iTrack/disable alert (Закрывать соединение / Отключить оповещения)	1. Если приложение установило значение в 0x03, iTrack выполнит следующие действия: – отключит функцию предупреждения о разъединении; – отключится от приложения; – прекратит вещание. 2. Если приложение установило 0x02, iTrack будет уведомлять об отключении от приложения. 3. Если приложение установило 0x01, iTrack не будет предупреждать, об отключении от приложения
Call iTrack (Вызвать iTrack)	Если приложение установило 0x03, iTrack запустит оповещение в течение 20 секунд. Если приложение установило это значение в 0x00, iTrack остановит оповещение сразу при запуске

6. Сопряжение

Рассмотрим один из ключевых моментов взаимодействия с устройством на базе профиль Find Me, а именно процесс обнаружения и сопряжения (рис. 2).

Шаг 1 – Шаг 4. Значение по умолчанию для режима Paring равно 0x0. После длительного нажатия (> 3 секунд) кнопки на устройстве, оно устанавливает режим Paring в 0x1 и начинает трансляцию, но через 30 секунд он вернёт это значение обратно в 0x0. Это используется для того, чтобы приложение соединялось с нужным маяком, кнопка на котором недавно была длительно нажата. Если приложение попытается подключиться к маяку, на котором не была нажата кнопка, и, соответственно, Paring Mode имеет значение 0x0, оно не сможет связаться с ним.

Шаг 5. Приложение подключается к устройству.

Шаг 6. В случае нашего устройства, после того, как приложение подключится к iTrack, оно должно записать 0xA1A2A3A4 в Password code (пароль) в течение 30 секунд, в противном случае iTrack отключится. Не все маяки данного профиля работают так.

Шаг 7. Эта характеристика – MAC-адрес, используется для iOS приложений, поэтому в данной работе не имеет особого назначения.

Шаг 8 – Шаг 9. Сопряжение между приложением и поисковым маяком преуспевает после прочтения Paring Mode равного 0x01.

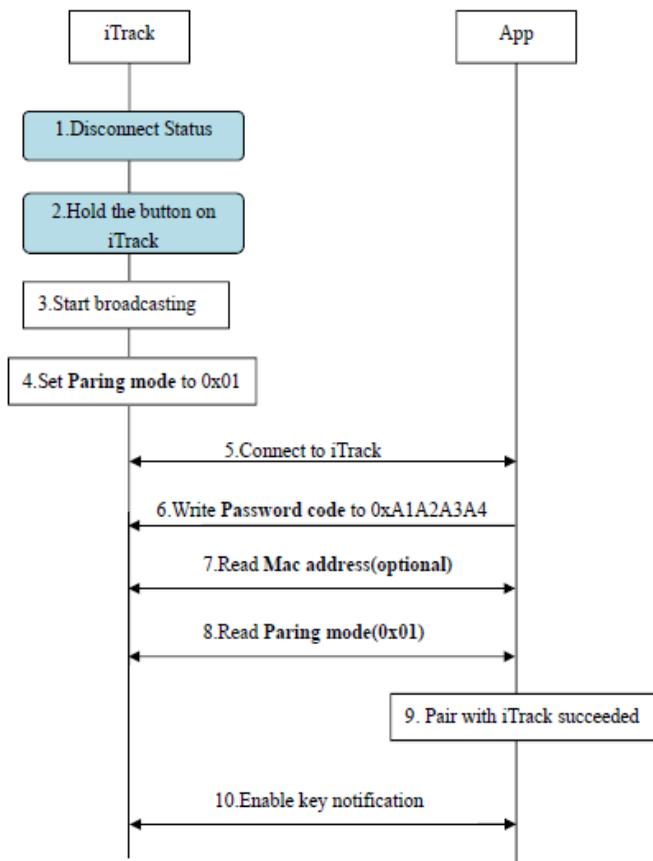


Рис. 1. Сопряжение с iTrack

7. Включение/выключение оповещения о разъединении

Как становится понятно из предыдущего пункта, ключевым моментом в работе с поисковыми маяками является установка значений определённых характеристик. Рассмотрим включение и выключение оповещения о разъединении (рис. 3).

Шаг 2 – Шаг 4: эти шаги описывают, как включить функцию предупреждения о разъединении с поисковым маяком.

Шаг 6 – Шаг 8: эти шаги описывают, как отключить функцию предупреждения об разъединении. Важно: значение по умолчанию для параметра Close / DisconnAlert равно 0x0. Это означает, что маяк установит значение 0x0 после завершения соединения.

8. Выключение

Для выключения следует записать значение 0x03 в параметр Close iTrack.

После того, как приложение отключит устройство, оно прекратит трансляцию и войдет в энергосберегающий режим. В этом режиме приложение не может повторно подключиться к маяку, пока вы длительно не нажмёте кнопку, которая позволит ему начать вещание снова.

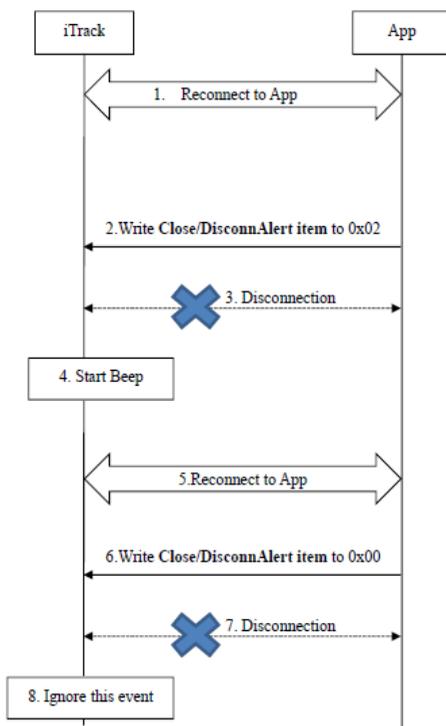


Рис. 2. Включение/выключение оповещения о разъединении

Заключение

В данной работе рассмотрены основные аспекты взаимодействия Android – приложений с BLE устройствами на базе профиля Find Me, в частности с поисковыми маяками iTrack. Для написания приложения, работающего с такими устройствами, следует хорошо представлять саму концепцию протокола BLE и владеть методами Android SDK. Большую часть данной работы занимает обзор характеристик и методов работы с поисковыми маяками iTrack, так как данный продукт не был предназначен

для использования сторонними разработчиками, такая работа была проведена впервые. Это даёт возможность ознакомившимся с ней специалистам, легко использовать маяки iTrack или любые другие BLE устройства этого профиля в собственных разработках.

Список литературы

1. Android API Guides [Электронный ресурс]. – Режим доступа : <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>
2. Bluetooth Adopted Specifications [Электронный ресурс]. – Режим доступа : <https://www.bluetooth.com/specifications/adopted-specifications>
3. iTrack API Design Introduction [Электронный ресурс]. – Режим доступа : http://www.kkmcn.com/Smart_tracker.
4. iTrack services and characteristic [Электронный ресурс]. – Режим доступа : http://www.kkmcn.com/Smart_tracker

ПРИМЕНЕНИЕ SPI-ИНТЕРФЕЙСА ДЛЯ УПРАВЛЕНИЯ ЭКВАЛАЙЗЕРОМ ВХОДНОГО ВИДЕОСИГНАЛА

Р.И. Струков

Студент

Н.А. Тюкачев

Доцент

Введение

На сегодняшний день область применения видеосигналов чрезвычайно велика: спортивные видеотрансляции, репортажи в прямом эфире, домашнее телевидение, онлайн-вебинары и многое другое. Также, в связи с широким распространением IP-сетей по всему миру, многие системы используют технологию Video Over IP, позволяющую передавать поток видео кадров в виде битового потока через IP-сети. Для организации потока бит в сети служит протокол Real-time Transport Protocol (RTP). Важным этапом трансформации видеопотока в битовый (RTP-пакет) является выравнивание входного сигнала с помощью эквалайзера.

1. Постановка задачи

Конвертация видеосигнала из формата SDI в формат RTP-пакета – сложный многоступенчатый процесс. В связи с этим даже небольшие фазовые или частотные отклонения передаваемого сигнала могут негативно повлиять на результирующий, преобразованный в новый формат видеопоток. Таким образом, перед трансформацией сигнала в биты

необходимо подготовить сигнал, подкорректировать его частотные характеристики с учетом затуханий и помех.

Работа с эквалайзером суть ключевой момент в рамках данного вида обработки сигнала. В настоящее время существует ряд способов коммуникации управляющего микроконтроллера (процессора) и эквалайзера. Основной задачей является выбор наиболее эффективного и, в то же время, оптимального для данного аспекта способа коммуникации между ними. Выбранный интерфейс должен отвечать следующим критериям:

- Низкая сложность программной реализации.
- Низкая сложность аппаратной реализации.
- Высокая скорость передачи данных.

2. Существующие пути коммуникации

Сегодня связь между эквалайзером и управляющим микроконтроллером чаще всего осуществляется через следующие интерфейсы [1]:

- Serial Peripheral Interface (SPI).
- Inter-Integrated Circuit (I²C) или его аналог Two Wire Interface (TWI).
- Universal Serial Bus (USB).
- Control Area Network (CAN).
- UART.

Далее, в данной статье будут более подробно рассмотрены интерфейсы SPI и I²C как наиболее часто применяемые в рамках решения поставленной задачи.

3. Сравнительный анализ интерфейсов SPI и I²C

Рассмотрены существующие на сегодняшний день способы управления эквалайзером. Далее, для наглядного сравнения, в таблице 1 произведено сравнение двух наиболее используемых интерфейсов в рамках решения поставленной задачи.

Предельная простота протокола передачи на физическом уровне обуславливает высокую надежность и быстродействие передачи. Предельное быстродействие шины SPI измеряется десятками мегагерц, благодаря чему она широко используется в высокоскоростных ЦАП/АЦП (цифро-аналоговый преобразователь/аналого-цифровой преобразователь), драйверах светодиодных дисплеев и микросхемах памяти. Все линии шины SPI однонаправленные, что значительно упрощает решение задачи преобразования уровней и гальванической изоляции микросхем.

Кроме того, простота программной реализации последовательного периферийного интерфейса, зачастую гарантирует удобство применения

данного интерфейса с точки зрения разработчика программного обеспечения.

Таблица 1

Сравнение интерфейсов по критериям

Критерий	SPI	I²C
Программная реализация	Особенности аппаратного подключения и работы интерфейса обеспечивают простоту программной реализации.	Достаточно сложно программируемый интерфейс в силу топологии подключения ведомых(Slave) устройств к ведущему(Master) [2] и механизма распределения адресов.
Аппаратная реализация	Четыре однонаправленные линии связи: MOSI, MISO, SCLK, SS. При кольцевой структуре связи не требует дополнительных проводов.	Две двунаправленные линии связи: SDA и SCL в независимости от количества подключенных микросхем.
Скорость передачи данных	До 20 Мбит\сек.	До 400 Кбит\сек.

4. Интерфейс SPI

Путем сравнения по трем критериям, установлено, что последовательный периферийный интерфейс является наиболее оптимальным интерфейсом для передачи данных между управляющим микроконтроллером и эквалайзером. Далее выбранный интерфейс будет рассмотрен более детально.

Интерфейс SPI организован по принципу «ведущий-подчиненный». В качестве ведущего шины выступает микроконтроллер, но им также может быть программируемая логика, DSP-контроллер или специализированная ИС[2]. Подключенные к ведущему внешние устройства образуют подчиненные шины, которыми могут выступать запоминающие устройства (EEPROM, Flash-память, SRAM), часы реального времени (RTC), АЦП/ЦАП и др. [3]

Аппаратная реализация последовательного периферийного интерфейса представляет из себя достаточно простую структуру (рис. 1).

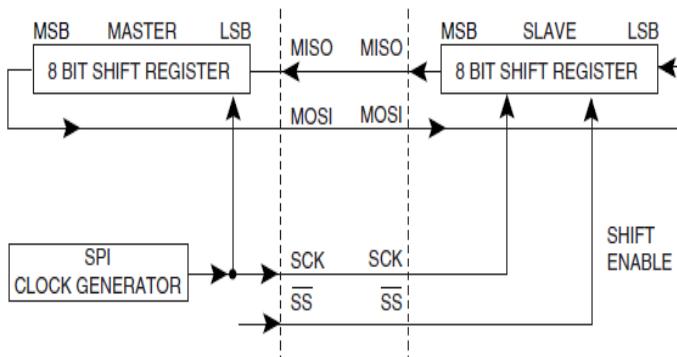


Рис. 1. Базовая структура SPI

И в ведущем, и в ведомом устройстве имеется 8-битный регистр сдвига, который передает и принимает биты одновременно, что позволяет шине работать в полнодуплексном режиме [4].

На рис. 2. представлена диаграмма классов для одного из вариантов программной реализации обмена данными через последовательный периферийный интерфейс.

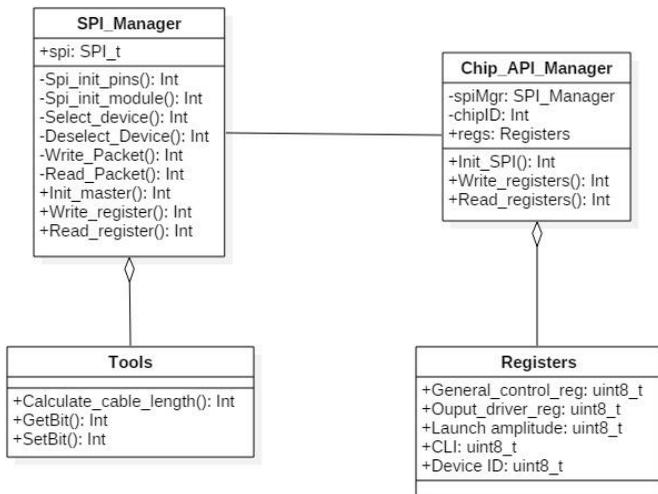


Рис. 2. Диаграмма классов программной реализации интерфейса SPI

Данная диаграмма классов является статической моделью системы, состоящей из управляющего микроконтроллера и линейного адаптивного эквалайзера. Эквалайзер имеет определенный набор регистров. Передача и чтение данных регистров с помощью последовательного периферийного интерфейса [4] позволяет эффективно управлять эквалайзером.

Ниже (рис. 3) приведена осциллограмма логического анализатора, которая иллюстрирует процесс передачи данных для рассматриваемой в данном разделе шины.

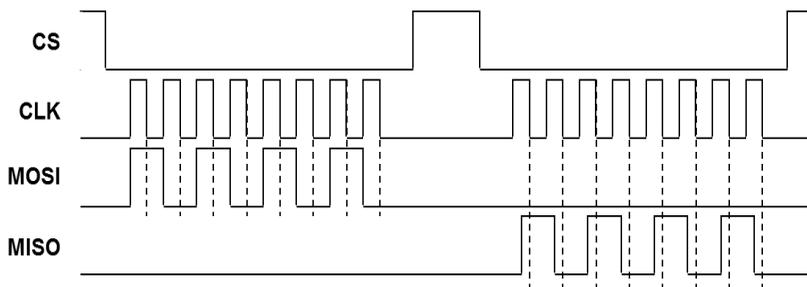


Рис. 3. Осциллограмма работы интерфейса SPI.

Заключение

Рассмотрено применение последовательного периферийного интерфейса (SPI) для осуществления коммуникации и управления линейным адаптивным эквалайзером. Опираясь на значимые критерии качества работы последовательных интерфейсов, последовательный периферийный интерфейс был определен как наиболее оптимальный способ коммуникации между управляющим микроконтроллером и адаптивным эквалайзером с точки зрения аппаратной реализации интерфейса, программной реализации интерфейса, а также скорости передачи данных. Также был рассмотрен возможный вариант программной реализации работы SPI.

Таким образом, применение последовательного периферийного интерфейса в целях управления адаптивным эквалайзером позволит произвести необходимое выравнивание видеосигнала перед его трансформацией в новый формат наиболее эффективно.

Список литературы

1. Лапин А.А. Интерфейсы. Выбор и реализация / Лапин А.А. – М. : Техносфера, 2005. – 168 с.

2. Логан, Стив. Управление несколькими периферийными устройствами по линиям шин SPI/I²C / Стив Логан // Компоненты и Технологии. – 2008. – №84. – С. 66-70.

3. Цифровая обработка сигналов. Конспект лекций : учебно-методическое пособие для вузов / Воронеж. гос. ун-т ; сост. : Д.Н. Борисов, Н.Н. Винокурова. – Воронеж : ИПЦ ВГУ, 2010 – 98 с.

4. Болл Стюарт Р. Аналоговые интерфейсы микроконтроллеров. / Болл Стюарт Р. – М. : Додэка-XXI, 2007. – 360 с.

ОЦЕНКА ВОЗМОЖНОСТИ РЕАЛИЗАЦИИ УГРОЗ БЕЗОПАСНОСТИ ПЕРСОНАЛЬНЫХ ДАННЫХ

М.Ю. Танвель

Студент

С.А. Вялых

Доцент

Введение

Определение угроз безопасности информации, реализация которых может привести к нарушению безопасности информации в информационной системе, и разработка на их основе модели угроз безопасности информации, является одним из важнейших этапов построения современных защищенных информационных систем [1, 2].

Основным методическим документом, определяющим порядок выявления актуальных угроз безопасности информации, является Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных, утвержденная заместителем директора ФСТЭК России 14.02.2008 [3] (далее – Методика). Актуальность угрозы определяется по результатам оценки двух основных показателей – возможности реализации угроз безопасности информации и их опасности. Источниками данных об угрозах безопасности информации, на основе которых определяются актуальные угрозы, являются:

– банк данных угроз безопасности информации ФСТЭК России [4] (<http://bdu.fstec.ru>, далее – Банк данных угроз);

– базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных, утвержденная заместителем директора ФСТЭК России 15.02.2008 [5].

1. Алгоритм оценки возможности реализации угроз безопасности информации в соответствии с Методикой

Согласно п. 2 Методики [1], возможность реализации угрозы безопасности информации определяется на основе оценок двух параметров – «уровень исходной защищенности системы» и «частота реализации угрозы».

Параметр «уровень исходной защищенности системы» может иметь три градации: высокий, средний, низкий. Оценка значения этого параметра подробно описана в Методике и зависит от технических и эксплуатационных характеристик информационной системы. Далее в Методике каждой степени защищённости ставится в соответствие числовой коэффициент Y_1 , значение которого зависит от уровня:

- 0 – для высокого;
- 5 – для среднего;
- 10 – для низкого.

Параметр «вероятность реализации угрозы» имеет 4 градации – маловероятно, низкая вероятность, средняя вероятность, высокая вероятность. Его оценка определяется экспертным путем в зависимости от того, насколько вероятным является реализация конкретной угрозы безопасности для оцениваемой информационной системы в складывающихся условиях. Далее каждой градации вероятности возникновения угрозы ставится в соответствие числовой коэффициент Y_2 :

- 0 – для маловероятной угрозы;
- 2 – для низкой вероятности угрозы;
- 5 – для средней вероятности угрозы;
- 10 – для высокой вероятности угрозы.

На основе данных числовых коэффициентов Y_1 и Y_2 вычисляется коэффициент реализуемости угрозы Y , который будет определяться соотношением:

$$Y = (Y_1 + Y_2) / 20, \quad (1)$$

где Y – коэффициент, определяющий реализуемость угрозы.

По значению коэффициента реализуемости угрозы Y формируется вербальная интерпретация реализуемости угрозы следующим образом:

- если $0 \leq Y \leq 0,3$, то возможность реализации угрозы признается низкой;
- если $0,3 \leq Y \leq 0,6$ – средняя возможность реализации угрозы;
- если $0,6 \leq Y \leq 0,8$ – высокая возможность реализации угрозы;
- если $Y > 0,8$ – возможность реализации угрозы очень высокая.

2. Оптимизация алгоритма оценки возможности реализации угроз безопасности информации

Алгоритм оценки возможности реализации угроз безопасности информации, изложенный в Методике, может быть оптимизирован путем математических преобразований. Для этого предлагается представленный выше алгоритм пошагово представить в табличном виде. Тогда в дальнейшем при применении оптимизированного алгоритма Методики промежуточные этапы вычислений можно не проводить. Если последовательно проделать предлагаемые шаги по присвоению числовых коэффициентов, вычислению по формуле (1) и формированию вербальной интерпретации реализуемости угрозы, то получим следующие результаты.

1. Результат суммирования числовых коэффициентов Y_1 и Y_2 представлен в табл. 1.

Таблица 1

Сумма коэффициентов Y_1 и Y_2

Вероятность реализации угрозы, коэффициент	Исходная степень защищенности системы, коэффициент		
	Высокий уровень, 0	Средний уровень, 5	Низкий уровень, 10
Маловероятно, 0	0	5	10
Низкая вероятность, 2	2	7	12
Средняя вероятность, 5	5	10	15
Высокая вероятность, 10	10	15	20

2. Результат оценки коэффициента, определяющего реализуемость угрозы Y по (1), представлен в табл. 2:

Таблица 2

Коэффициент, определяющий реализуемость угрозы Y

Вероятность реализации угрозы	Исходная степень защищенности системы		
	Высокий уровень	Средний уровень	Низкий уровень
Маловероятно	0	0,25	0,5
Низкая вероятность	0,1	0,35	0,6
Средняя вероятность	0,25	0,5	0,75
Высокая вероятность	0,5	0,75	1

3. По значению коэффициента Y формируется результат, табл. 3:

Таблица 3

Возможность реализации угрозы

Вероятность реализации угрозы	Исходная степень защищенности системы		
	Высокий уровень	Средний уровень	Низкий уровень
Маловероятно	низкая	низкая	средняя
Низкая вероятность	низкая	средняя	средняя
Средняя вероятность	низкая	средняя	высокая
Высокая вероятность	средняя	высокая	очень высокая

Таким образом, при оценке возможности реализации угрозы безопасности информации с применением Методики можно сократить вычисления и использовать оптимизированный алгоритм, задаваемый табл. 3.

Заключение

Эксперты при разработке модели угроз безопасности информации рассмотренный выше алгоритм должны применять для каждой из оцениваемых угроз. В настоящее время таких угроз может насчитываться более 200 [4, 5]. Использование оптимизированного алгоритма оценки возможности реализации угрозы безопасности информации позволит сократить время разработки таких моделей и более наглядно представить зависимость возможности реализации угрозы безопасности информации от исходных параметров.

Список литературы

1. Федеральный закон от 27 июля 2006 года № 152-ФЗ «О персональных данных» // Собрание законодательства Российской Федерации, 31 июля 2006 года. – № 31 (1 ч.). – Ст. 3451.
2. Приказ ФСТЭК России от 11 февраля 2013 года № 17 «Об утверждении Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах» // Российская газета. – 26.06.2013. – № 136.
3. Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных. – ФСТЭК России. Утверждена Заместителем директора ФСТЭК России 14 февраля 2008 г.

4. Банк данных угроз безопасности информации ФСТЭК России [Электронный ресурс] : база данных. – Режим доступа : <http://bdu.fstec.ru/>

5. Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных (выписка). Утверждена заместителем директора ФСТЭК России 15 февраля 2008 г. [Электронный ресурс] : база данных. – Режим доступа : <http://fstec.ru/component/attachments/download/289>

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ОДНОГО КАНАЛА ПРИЕМА АНТЕННОЙ РЕШЕТКИ QPSK-СИГНАЛА

С.И. Тюленев

Студент

Н.А. Семенихин

Студент

Ю.Б. Нечаев

Профессор

Введение

Разработка и внедрение цифровых антенных решеток (ЦАР) позволит понизить массогабаритные размеры радиотехнических комплексов, повысить чувствительность приемного тракта, без дополнительных аппаратных затрат формировать одновременно множество лучей в заданных направлениях, а также в реальном масштабе времени реализовать управление амплитудно-фазовым распределением для пространственной селекции сигналов [1]. Использование технологии независимой многоканальной обработки сигналов позволяет значительно увеличить помехозащищенность и информационную емкость радиоканалов.

В статье представлена схема одного канала приема QPSK-сигнал, которая может быть использована в цифровой антенной решетке для отделения сигнала от шумов.

1. Генерация QPSK-сигнала

На рис. 1 представлена схема генерация квадратурного сигнала. Двоичная последовательность бит разделяется на четные и нечетные биты. По четным битам будет строиться синфазная часть $I(t)$ (2), а по нечетным квадратурная часть $Q(t)$ (3). Синфазная часть перемножается с синусоидой, а квадратурная перемножается с косинусоидой. Перед суммированием и

получением QPSK сигнала необходимо повысить частоту сигнала. Для этого каждая часть перемножается с синусоидой более высокой частоты и суммируется. После блока суммирования сигнал попадает на блок усиления, после чего он передается.

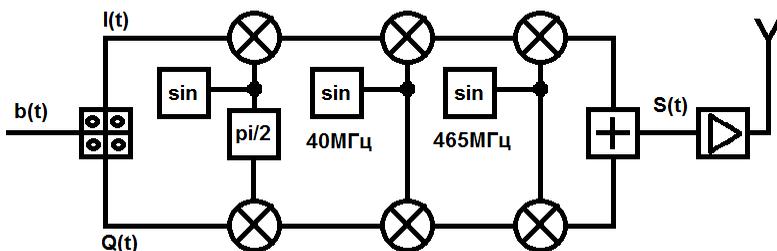


Рис. 1. Схема генерация сигнала

Произвольный полосовой сигнал приближенно может быть описан следующим выражением:

$$s(t) = b(t) \cos(\omega_0 t + \psi(t)) = I(t) \sin(\omega_0 t) \pm Q(t) \cos(\omega_0 t), \quad (1)$$

где ψ – фаза колебания, ω_0 – несущая частота.

Компоненты I и Q могут быть выражены следующим образом:

$$I(t) = b(t) \sin(\psi(t)) \quad (2)$$

$$Q(t) = b(t) \cos(\psi(t)) \quad (3)$$

2. Схема приема

На рис. 2 представлена структурная схема одного элемента антенной решетки.

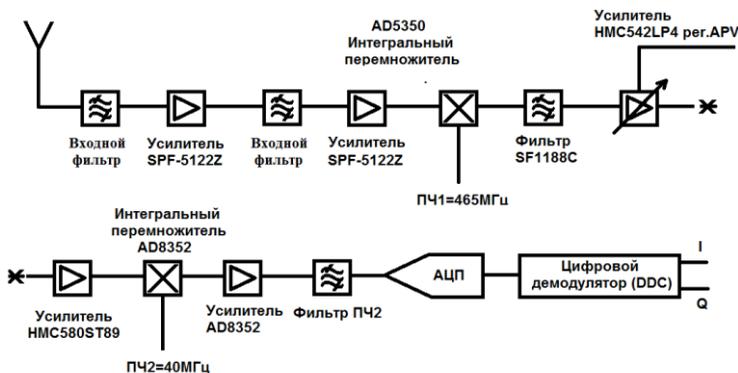
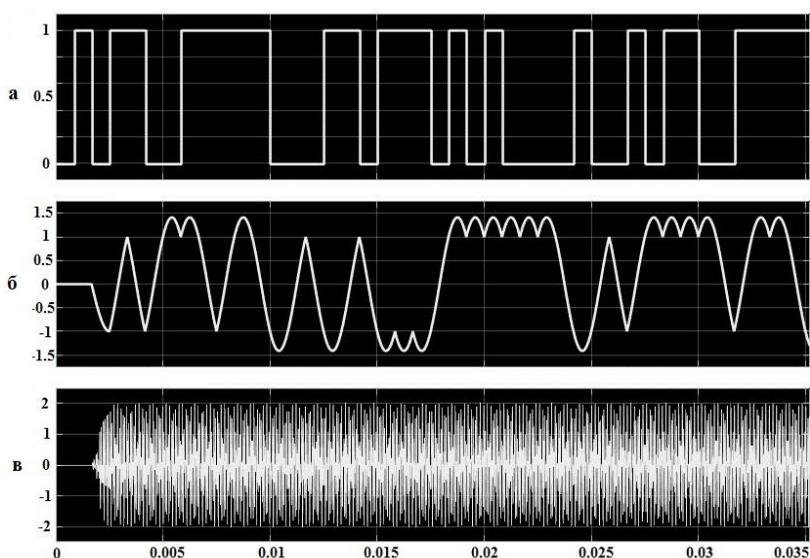


Рис. 2. Структурная схема одного канала антенной решетки

После того, как сигнал поступил на элементы антенной решетки, он проходит через систему фильтров и усилителей. Перед оцифровкой, сигнал необходимо снять с первой и второй промежуточной частоты, а затем он попадает на цифровой демодулятор, на выходе которого получается синфазная и квадратурная части сигнала. Антенная решетка может быть различной формы. За антенные элементы были взяты U-образные патч-антенны на 1.5-1.7 ГГц [2].

3. Результат моделирования

В математическом пакете Matlab/Simulink была промоделирована схема генерации, передачи и приема QPSK-сигнала (рис. 3). При передаче на сигнал действовал аддитивный белый Гауссовский шум.



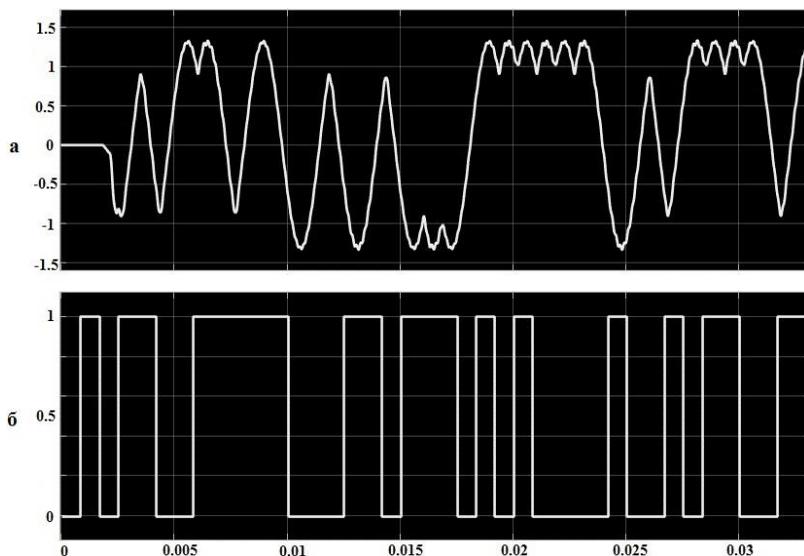
*а – передаваемая последовательность, б – QPSK-сигнал,
в – передаваемый сигнал*

Рис. 3. Генерация сигнала

При моделировании использовались различные фильтры. На рис. 3 и рис. 4 представлены результаты работы программы при использовании фильтра Баттлворта. Перед снятием первой промежуточной частоты, необходимо использовать фильтр третьего и четвертого порядка, а после снятия второй промежуточной частоты шестого или седьмого порядка, иначе фазы сигнала будут сильно искажены.

На рис. 4 видно, что фазы сигнала были немного искажены под воздействием шума, но цифровая последовательность после демодуляции полностью совпадает с переданной последовательностью бит.

Очень хороший результат показал фильтр Бесселя, но после него требуется усиливать сигнал более чем в два раза, по сравнению с фильтром Баттерворта. Фильтры Чебышева, первого и второго порядка, а также Эллиптический фильтр, не убрали искажения полностью. На выходе фильтра Чебышева второго порядка не были получены многие фазы сигнала. Рекомендуется на практике использовать фильтры Бесселя и Баттерворта.



а – полученный QPSK-сигнал, б – последовательность бит после цифровой демодуляции

Рис. 4. Полученный сигнал

Большое количество фильтров и усилителей необходимо для того, чтобы полностью избавиться от шумов. В итоге на приемной стороне был получен QPSK-сигнал без искажений [3, 4].

Заключение

В ходе имитационного моделирования одного канала приема QPSK-сигнала для антенной решетки выбраны оптимальные компоненты приемного тракта. Моделирование различных фильтров, позволило выбрать наиболее подходящие, для QPSK-сигналов в данной схеме. Также

было установлено что, данная схема может быть использована для нахождения координат источников радиоизлучения. Для этого рекомендуется использовать такие алгоритмы как MUSIC или Кейпона [2].

Список литературы

1. Sarkar. Т.К. Smart Antennas / Т.К. Sarkar, М.С. Wicks, М. Salazar-Palma, R.J. Bonneau. – New Jersey: John Wiley & Sons, Inc., 2003, 453 p.
2. Нечаев Ю.Б. Оценка влияния параметров модуля первичной обработки на работу цифровой антенной решетки / Ю.Б. Нечаев, Д.Н. Борисов, И.В. Пешков // Нелинейный мир. – 2010. – Т. 8. – № 3. – С. 151-159.
3. Скляр Б. Цифровая связь. Теоретические основы и практическое применение / Б. Скляр. – М. : Вильямс, 2003. – 1104 с.
4. Wirth W.-D. Radar techniques using array antenna / W.-D. Wirth. – London : The Institution of Electrical Engineers, 2001. – 420 p.

АЛГОРИТМ ПОСТРОЕНИЯ ДИСКРЕТНОЙ ДИАГРАММЫ ВОРОНОГО К-ГО ПОРЯДКА ПРОИЗВОЛЬНОЙ РАЗМЕРНОСТИ С ЗАДАННОЙ ТОЧНОСТЬЮ

С. А. Хальзов

Студент

В. В. Фертиков

Доцент

Введение

Диаграмма вороного широко применяется при решении множества задач, в том числе для получения скелета изображения [1]. Классическая диаграмма Вороного является непрерывной и строится для множества точек в n -мерном пространстве. Со временем появилось множество обобщений классической диаграммы Вороного, например, сайтами могут быть не только точки, как в классическом варианте, но и отрезки прямых. Такой вариант используется при построении непрерывного скелета изображения по его контуру, аппроксимированному отрезками прямых [1]. Также существуют обобщения, в которых, например, используют не евклидово расстояние, а манхэттенское или расстояние Чебышева.

Существующие на данный момент алгоритмы построения непрерывной обобщенной диаграммы Вороного могут построить диаграмму для точек, отрезков и дуг в 2 -х мерном пространстве, и для

© Хальзов С. А., Фертиков В. В., 2017

точек, отрезков и многоугольников в 3-х мерном пространстве. Алгоритмы в данном случае уже достаточно сложны. Дальнейшее обобщение форм сайтов требует разработки на порядок более сложных алгоритмов. В общем случае построение непрерывной обобщенной диаграммы Вороного невозможно [1]. Альтернативным подходом является построение дискретной диаграммы и последующая ее аппроксимация непрерывными фигурами. Такой подход позволяет обойти алгоритмически сложный этап, но для обеспечения приемлемой точности необходимо произвести значительно больше расчетов, чем для построения непрерывной диаграммы для такого же количества сайтов.

В данной работе представлен алгоритм, который позволяет рекурсивно строить обобщенную дискретную диаграмму Вороного k -го порядка в пространстве произвольной размерности с заданной точностью в областях прохождения бисекторов. Данный алгоритм является развитием алгоритма, представленного в [2].

1. Основная идея алгоритма

Самый простой способ построить дискретную диаграмму Вороного, это дискретизировать область построения диаграммы, для каждой полученной дискретной точки найти ближайший сайт. То есть, например, для 2-мерного случая на область построения диаграммы накладывается регулярная сетка $N \times N$ точек, для каждой точки ищется ближайший сайт, после чего каждой точке назначается индекс ближайшего сайта. Точки границы (бисектора) между ячейками Вороного находятся как места, в которых хотя бы по одному из направлений изменяется индекс ближайшего сайта.

В большинстве случаев интерес представляет форма ячейки Вороного, то есть в дискретном случае рассматриваются только граничные точки (точки бисектора), и желательно достигнуть большей точности при построении дискретной диаграммы Вороного именно в областях прохождения бисекторов. Для достижения данной цели можно использовать алгоритм, который основан на рекурсивном разбиении пространства. Например, для 2-мерного случая строится дерево квадратов, для 3-мерного – дерево октантов, и так далее. Глубина построения дерева больше в области прохождения бисекторов, за счет чего и достигается большая точность при меньшем объеме хранимых данных и объеме требуемых вычислений по сравнению с регулярной дискретной сеткой.

Дерево строится по слоям, при построении следующего слоя находят узлы дерева, которые предположительно расположены рядом с бисектором, то есть имеют соседа из другой (отличающейся от рассматриваемой) ячейки Вороного. Каждый такой узел разбивается на

соответствующее число меньших (для 2-мерного случая на 4 части). Данная процедура повторяется до тех пор, пока не будет достигнута заданная глубина построения дерева, а, следовательно, и требуемая точность. Формально алгоритм можно описать следующей последовательностью действий:

1. построить некоторое первое приближение;
2. найти области, представляющие интерес;
3. построить диаграмму с большей точностью в областях, представляющих интерес;
4. если не достигнута требуемая точность, то перейти к пункту 2.

При таком подходе в области прохождения бисекторов будет достигнута требуемая точность без необходимости производить расчеты для полной дискретной сетки, что приводит к значительной экономии ресурсов ЭВМ.

В классической диаграмме Вороного сайты являются точками, и расстояние до них вычисляется в евклидовой метрике. В предлагаемом алгоритме сайты будут представлены функциями, которые будут вычислять расстояние от заданного сайта до произвольной точки области построения диаграммы. Такое обобщение снимает ограничение на форму сайта и тип используемой метрики.

Также для обеспечения большей гибкости при определении сайтов, алгоритмом уже предусмотрено некоторое обобщение. Сайты могут быть представлены как совокупность подсайтов, и расстояние до сайта считается равным минимальному среди расстояний до его подсайтов.

Также использование сайтов-функций позволяет обобщить систему координат, масштаб осей координат, и даже до некоторой степени тип пространства (например, вместо евклидова пространства можно использовать некоторое параметрическое пространство).

Возможно также использовать абстрактные сайты, в данном контексте под абстрактным подразумевается то, что сайт не является геометрической фигурой до которой вычисляется расстояние, а вместо этого является некоторой функцией от вектора координат. Для абстрактных сайтов термин «ближайший» следует заменить на «доминирующий», то есть в общем случае для тестовой точки будет определяться не ближайший, а доминирующий сайт.

2. Улучшение алгоритма

Помимо уменьшения количества рассчитываемых точек, также можно уменьшить количество перебираемых сайтов для точки при поиске ближайшего (доминирующего) сайта. Данное упрощение возможно, например, потому, что влияние сайта, при использовании евклидовой или манхэттенской метрик, убывает с увеличением расстояния до него, и

поэтому имеет смысл искать ближайший сайт не среди всех сайтов, а только среди тех, которые попали в некоторую окрестность точки. Для других метрик также может быть свойственна зависимость влияния сайта от расстояния до него. Размер окрестности естественно будет уменьшаться (не увеличиваться) при уменьшении размера дискретной ячейки.

Окрестность должна вычисляться на основе свойств доминирующего сайта (ближайшего к точке сайта), и при формировании нового списка сайтов-кандидатов сайты должны выбираться на основе принадлежности данной окрестности.

Такой подход позволит значительно ускорить построение дерева, но будет требоваться дополнительная память для хранения списков сайтов-кандидатов для каждого узла дерева. Модифицированный алгоритм можно описать следующей последовательностью действий:

1. построить некоторое первое приближение используя для поиска ближайшего сайта все множество сайтов;

2. найти области, представляющие интерес, и для каждой построить множество сайтов-кандидатов;

3. построить диаграмму с большей точностью в областях, представляющих интерес, используя для каждой области свой набор сайтов-кандидатов;

4. если не достигнута требуемая точность, то перейти к пункту 2.

Обычно диаграмма Вороного k -го порядка строится последовательно, начиная с диаграммы 1-го порядка, но описанный в данной статье алгоритм получения дискретной диаграммы Вороного после небольшой модификации позволяет находить диаграмму k -го порядка за один проход. Модификация заключается в следующем:

1. при поиске ближайшего сайта необходимо искать не один, а k ближайших сайтов;

2. в качестве метки-сайта узла дерева использовать вместо индекса ближайшего сайта множество (или сортированный список) из индексов k ближайших сайтов.

3. Сложность алгоритма

Асимптотическая сложность представленного алгоритма составляет $O(kN^{d-1})$, где k – порядок диаграммы, s – число сайтов, N – число разбиений по измерению, d – число измерений. Для сравнения сложность наивного алгоритма, строящего полное разбиение пространства, составляет $O(kN^d)$. Сложность алгоритма, основанного на расчете карты расстояний, составляет $O(N^d)$, но существующие на данный момент алгоритмы расчета карты расстояний рассчитаны на использование одной метрики для всех сайтов, и сайт должен быть точкой, либо

аппроксимирован множеством точек, а также строится только диаграмма 1-го порядка.

4. Эксперименты

На рис. 1-3 в качестве осей графиков показаны евклидовы координаты, сайты-точки обозначены маркерами в форме звезды, точками показаны позиции, выбранные алгоритмом для производства вычислений. На рис. 1 и 2 показаны все рассчитанные точки. На рис. 3 показаны только целевые точки (точки бисектора).

На рис. 1 представлен результат работы алгоритма для 2-х точек на плоскости. На рисунке видно, что точность выше в области прохождения бисектора.

На рис. 2 показан результат работы алгоритма для случая обобщенной диаграммы Вороного, сайтами могут быть точки и отрезки прямых. Для отрезка использовалась евклидова метрика, для левой точки – манхэттенская, для средней – евклидова, а для правой – расстояние Чебышева. Алгоритм позволяет одинаково просто находить дискретную диаграмму Вороного для различных комбинаций сайтов и метрик без необходимости производить расчеты для всей дискретной сетки.

Построено 4 диаграммы Вороного разного порядка (от 1-го до 4-го) для 10 точек (рисунок 3).

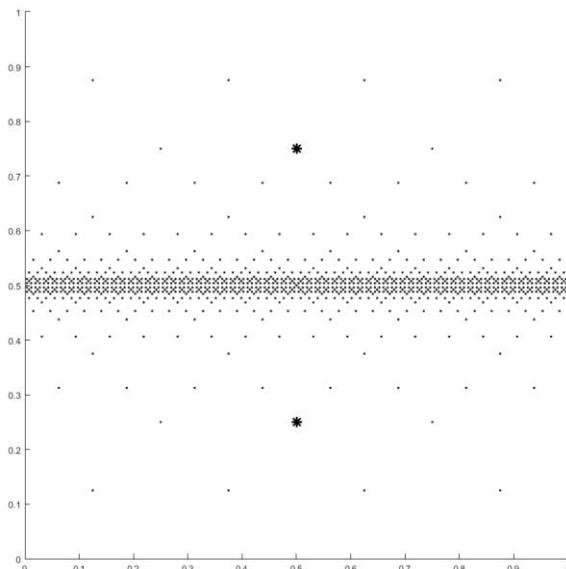


Рис. 1. Итеративное уточнение бисектора

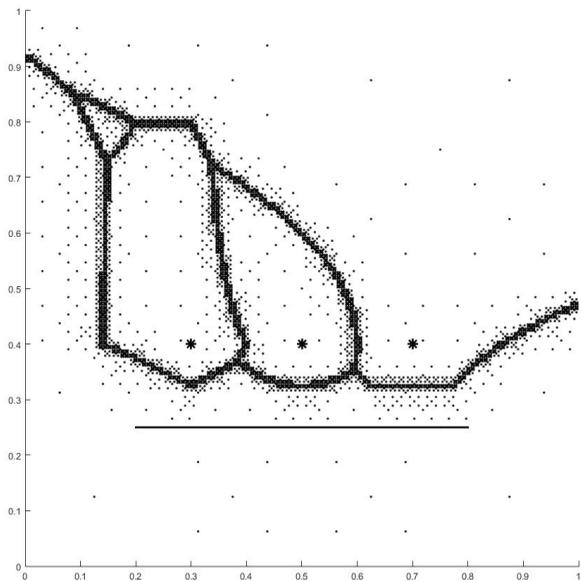


Рис. 2. Обобщение сайтов и метрик

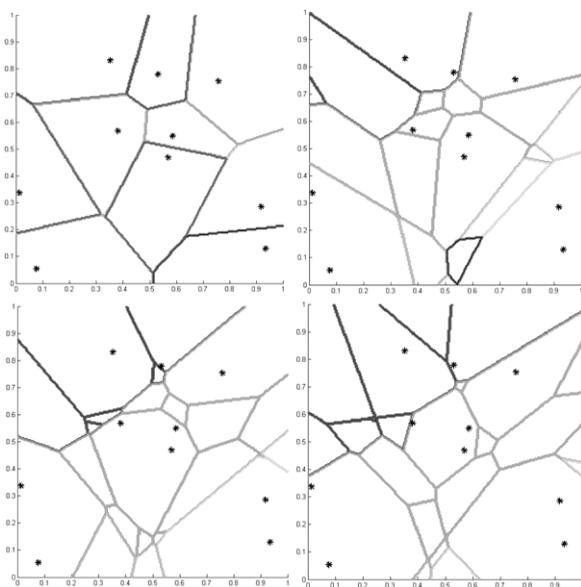


Рис. 3. Диаграммы Вороного с 1-го по 4-ый порядки

Заключение

В рамках данной работы был разработан алгоритм, позволяющий строить обобщенную дискретную диаграмму Вороного k -го порядка произвольной размерности с заданной точностью. Алгоритм обладает большой гибкостью по отношению к применяемой метрике или форме сайта. Асимптотическая сложность представленного в данной работе алгоритма меньше, чем у существующих альтернатив. Вследствие чего при построении диаграммы с достаточно большой точностью представленный алгоритм выиграет у альтернативных, которые рассчитывают полную дискретную сетку.

Были проведены эксперименты, подтверждающие работоспособность алгоритма, а также его эффективность по сравнению с алгоритмами, рассчитывающими полную дискретную сетку.

В качестве дальнейшего направления работы можно рассмотреть задачу оптимизации работы алгоритма и решение проблемы экспоненциального роста сложности алгоритма при увеличении размерности.

Список использованной литературы

1. Местецкий, Л.М. Непрерывная морфология бинарных изображений. Фигуры, скелеты, циркуляры / Л. М. Местецкий — М. : ФИЗМАТЛИТ, 2009. – 288 с.
2. Хальзов, С.А. Алгоритм построения дискретной диаграммы Вороного произвольной размерности с заданной точностью / С.А. Хальзов, В.В. Фертиков // Информатика: проблемы, методология, технологии: сборник материалов XVII международной научно-методической конференции: в 5 т. — Воронеж: Издательство «Научно-исследовательские публикации» (ООО «Вэлборн»), 2017. – Т. 3. – С. 183-188.

СИСТЕМА АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ДОМАШНИМ ОБОРУДОВАНИЕМ С ОБРАТНОЙ СВЯЗЬЮ «УМНЫЙ» ДОМ

С.В. Черепанова

Магистрант

Н.А. Тюкачев

Доцент

Введение

В современном мире дома и квартиры наполнены разнообразной техникой, электрическими приборами, системами отопления и вентиляции. За работой всего этого следят сами люди, при помощи пультов, выключателей, вентилей. Но техника может следить и сама за своей работой, используя домашнюю автоматизацию по заданным сценариям, выполнять все, что люди делают вручную. Именно автоматизация позволяет чувствовать домашний комфорт, а «умный» дом совмещает в себе домашнюю автоматизацию и интеллектуальные современные технологии [1].

1. Варианты и возможности домашней автоматизации

Возможности домашней автоматики на сегодняшний момент обширны и ограничиваются лишь потребностями людей. «Умные» дома позволяют достигать максимально слаженной и эффективной работы домашней техники, приборов и оборудования с минимальными затратами ресурсов. «Умная» автоматизация действительно позволяет значительно экономить электропотребление.

Из вариантов сценариев домашней автоматизации, пожалуй, можно выделить несколько основных:

- автоматизация внутреннего и наружного освещения;
- управление системами отопления, кондиционирования и вентиляции;
- обеспечение безопасности: защита от взломов и проникновений;
- работа домашней техники по заданным сценариям.

Примерами таких вариантов может служить использование датчиков движения, позволяющие осветительным приборам включаться, когда человек входит в комнату и выключаться, когда он из нее выходит. Модуль приема сообщений по определенному протоколу электрической сети позволяют включать и выключать приборы при помощи сообщений с мобильного телефона. Датчики температуры дают информацию по нагреву

воздуха, что в свою очередь обеспечивает необходимую автоматическую работу обогревательных приборов и кондиционеров. Биометрические считыватели – приборы, сканирующие отпечатки пальцев, позволяют запрограммировать систему так, что доступ в помещение, перед входом в которое они находятся, получают только, внесенные в базу считывателей, персоны. И именно биометрический считыватель позволит системе определить, кто вошел в дом и какой сценарий дальнейших действий нужно запустить. Сценарии записываются в зависимости от предпочтений человека [1, 2, 3].

Модули приема сообщений позволяют передавать команды бытовой технике: нагрев воды в чайнике или запуск работы кофе-машины, включение/выключение телевизоров, музыкальных систем, электрических плит и т.д. Также, уже весьма неновый, сценарий автоматического раскрытия штор в утренние часы пробуждения и закрытие в вечерние [3].

2. Система управления температурой с обратной связью

Для управления системой отопления возможно применение принципа автоматического управления температурой в помещении с обратной связью. Управляемой величиной является средняя температура в помещении $\theta_{ср}$, а управляющей величиной служит расход греющей жидкости D через нагревательный элемент.

Пусть возмущающим воздействием будет изменение температуры окружающей среды θ_{oc} . Измеряя величину, управляющее устройство устанавливает требуемое значение расхода греющей жидкости согласно заданной температуре $\theta_{ср}^{зад}$, путем изменения степени открытия клапана греющей воды μ , тем самым, компенсируя влияние возмущающего воздействия (рис. 1).

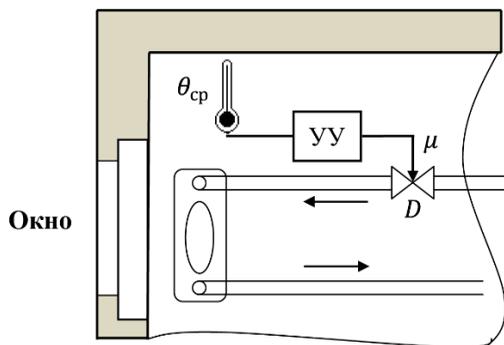


Рис. 1. Технологическая схема замкнутой системы управления температурой в помещении

На объект управления, как правило, действует множество случайных возмущений \mathfrak{D} , которые учесть невозможно. Например, число человек в комнате в определенный момент времени, момент открытия и закрытия дверей, форточек, окон и т. д.

Для того чтобы учесть влияние случайных возмущений, действующих на объект, применяют принцип управления по отклонению.

В подобной схеме (рис. 2) управляющее устройство для формирования управляющего воздействия использует разность между заданным значением температуры в помещении и текущим:

$$\varepsilon(t) = \theta_{cp}^{зад}(t) - \theta_{cp}(t), \quad (1)$$

где $\theta_{cp}^{зад}(t)$ – заданная температура, θ_{cp} – средняя температура в помещении.

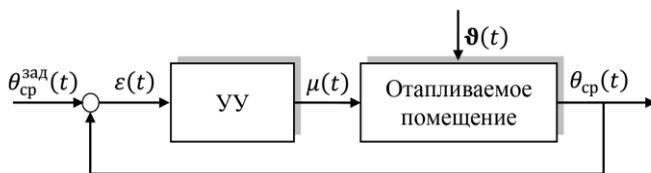


Рис. 2. Структурная схема замкнутой системы управления температурой в помещении

Величина $\varepsilon(t)$ в (1) называется отклонением или рассогласованием системы. В зависимости от того, прибавляется сигнал обратной связи к входному сигналу или вычитается из него, имеет место положительная или отрицательная обратная связь [4].

3. Объектно-ориентированный подход к программному коду домашней автоматизации

Работу всех датчиков, модулей и самого оборудования увязывает программный код, с помощью которого создаются сценарии. Грамотные сценарии позволяют в разы сократить потребление электроэнергии.

Для создания качественного программного кода отлично подойдут такие принципы объектно-ориентированного программирования (ООП) как полиморфизм и наследование. Для примера используем объектно-ориентированный язык программирования C# [5].

Базовым классом для всех устройств будет абстрактный класс Device (листинг 1).

Базовый класс для устройств

```

public abstract class Device {
private string name;
public string Name
{ get { return name; } set { name = value; } }
private bool state;
public virtual bool State
{ get { return state; } set { state = value; } }
public abstract void MessageBD();
public Device() { }
public Device(string name, bool state)
{ this.name = name;
  this.state = state; }
}

```

Он обладает общими свойствами, такими как: имя устройства Name и состояние State, конструкторами Device() и абстрактным методом MessageBD(), предназначенным для записи изменений состояния устройств в базу данных.

Класс Device порождает два класса: класс управляемых устройств OperatedDevice (листинг 2) и класс датчиков Guage (листинг 3).

Класс устройства

```

public class OperatedDevice : Device {
public OperatedDevice() { }
public OperatedDevice(string name, bool state) :
    base(name, state) { }
public virtual void MessagePhone() { }
public override void MessageBD()
{ Console.WriteLine("OperatedDevice-> name = {0}", Name); }
}

```

Класс датчиков

```

public class Guage : Device {
override public bool State
{ get { return State; } }
protected float voltage;
public override void MessageBD(){}
}

```

В классе управляемых устройств OperatedDevice включается виртуальный метод MessagePhone(), предназначенный для генерации сообщений от некоторых устройств на телефон.

От класса управляемых устройств OperatedDevice порождается класс – нагревательный котел HeatingBoiler (листинг 4).

Листинг 4

Класс нагревательный котел

```
public class HeatingBoiler : OperatedDevice, IMessageBD {
protected double temperature;
public double Temperature
{ get { return temperature; }
  set { if (value<22) {value=0; state = false; }
        temperature = value; } }
public override void MessagePhone() { }
public void MessageBD2() { }
public HeatingBoiler() { }
public HeatingBoiler(string name, bool state, float
                    temperature): base(name, state)
{ this.temperature = temperature; }
}
```

От класса датчиков Guage порождается класс датчиков температуры GuageHeat (листинг 5).

Листинг 5

Класс датчиков температуры

```
public class GuageHeat : Guage {
protected float temperature;
public float Temperature
{ get { return temperature; }
  set { temperature = value; } }
}
```

После описания всех классов, относящихся к оборудованию, понадобится описание класса, взаимодействующего с оборудованием, а именно, класса комнаты Room (листинг 6). В этом классе введено свойство температура Temperature и два конструктора.

Листинг 6

Класс комнаты

```
public class Room {
protected double temperature;
public double Temperature
{ get { return temperature; }
  set { temperature = value; } }
public Room() { }
public Room(double temperature)
{ this.temperature = temperature; }
}
```

И, наконец, опишем основной класс «Умный» дом IntellectualHouse (листинг 7). В этом классе введено свойство внешней температуры TemperatureOut, массив устройств myDevice[], конструктор IntellectualHouse() и метод WriteBD(), предназначенный для записи состояния устройств в базу данных [5].

Листинг 7

Основной класс умного дома

```
public class IntellectualHouse {
    protected static double temperatureOut; // температура вне
    public static double TemperatureOut
    { get { return temperatureOut; }
      set { temperatureOut = value; } }
    protected static Room room;

    public OperatedDevice[] myDevice = new OperatedDevice[3];

    public void WriteBD()
    { for (int i=0; i < myDevice.Length;i++)
      myDevice[i].MessageBD(); }
    public IntellectualHouse()
    { room = new Room(20);
      myDevice[0] = new HeatingBoiler("Котел", true,50); }
}
```

Общую схему всех классов можно наглядно представить в виде диаграммы (рис. 3).

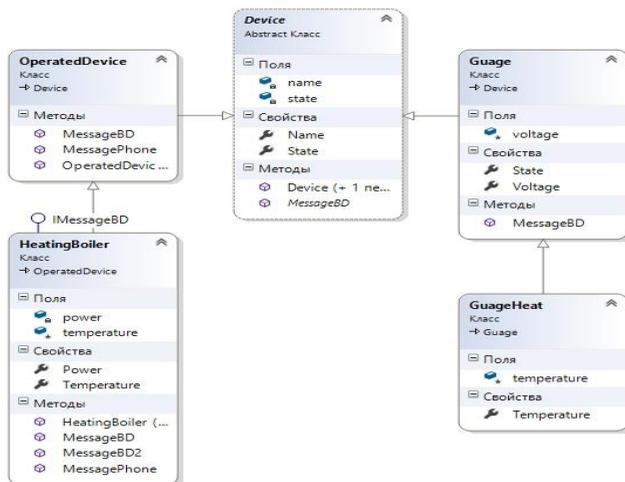


Рис. 3. Диаграмма классов для проекта «Умный» дом

Заключение

В данной статье представлено краткое описание основных систем домашней автоматизации, а также подробно рассмотрена система управления температурой с обратной связью. Показан один из вариантов создания программного кода при помощи принципов объектно-ориентированного программирования. В итоге данного эксперимента получилась рабочая система управления домашним нагревательным котлом. Использование решения автоматизированного управления домашним оборудованием позволяет решать задачи комфорта и экономии потребления энергоресурсов, что является несомненным преимуществом.

Список литературы

1. Goodwin, S. Smart Home Automation with Linux / S. Goodwin. – N.Y. : Springer-Verlag, 2010. – 289 p.
2. Головин Д. «Умный» дом: жизнь в ногу со временем [Электронный ресурс] : i-home.ru Laboratory. – Режим доступа : <http://www.i-home.ru/site.xp/049055048050124055057054052124.html>
3. Долин С. Сила света [Электронный ресурс]. – Хакер : журнал. – 2007. – № 99. – Режим доступа : <http://xaker-archive.ru/xa/099/060/1.htm>
4. Репин А. И. Теория автоматического управления / А. И. Репин. – Можайск : ИП Костюхина, 2011. – 150 с.
5. Тюкачев Н. А. С#. Программирование : учебное пособие : в 3 ч. / Н. А. Тюкачев, В. Г. Хлебостроев // Воронеж : Издательско-полиграфический центр Воронежского государственного университета, 2013. – 308 с.

ИНТЕГРИРОВАНИЕ МАТРИЧНЫХ АЛГЕБР ЛИ

А.В. Швырева

Студент

А.В. Лобода

Профессор

Введение

В данной работе представлено решение задачи интегрирования одного семейства вещественных матричных алгебр Ли, полученного А.В. Шиповской (см. [1]). Рассматриваемое семейство алгебр описывает аффинно-однородные поверхности так называемого $(\varepsilon_1, \varepsilon_2)$ -типа, который остается самым объемным и наименее изученным после

публикации серии статей об аффинной однородности в C^3 ([2] – первая из них).

Работа выполнена в рамках изучения курса «Математические методы в современных информационных технологиях» и продолжает исследования, в которых уже несколько лет принимают участие студенты ФКН ([3]).

1. Постановка задачи

Необходимо найти вещественнозначную функцию трех комплексных переменных $\Phi = -v + F(z_1, z_2, u)$, где $z_1 = x_1 + iy_1$, $z_2 = x_2 + iy_2$, $w = u + iv$, которая должна удовлетворять системе из пяти уравнений:

$$\operatorname{Re}\{E_k(\Phi)\}_{\Phi=0}, k = 1..5. \quad (1)$$

Каждое из аффинных векторных полей E_k имеет следующий вид

$$\begin{aligned} & (A_1 z_1 + A_2 z_2 + A_3 w + p) \frac{\partial}{\partial z_1} + (B_1 z_1 + B_2 z_2 + B_3 w + s) \frac{\partial}{\partial z_2} + \\ & + (a z_1 + b z_2 + c w + q) \frac{\partial}{\partial w}. \end{aligned} \quad (2)$$

В упрощенной форме алгебры аффинных векторных полей можно рассматривать как матричные алгебры, состоящие из матриц вида

$$\begin{bmatrix} A_1 & A_2 & A_3 & p \\ B_1 & B_2 & B_3 & s \\ a & b & c & q \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

с комплексными элементами.

Система (1) имеет решение, если базисные матрицы (3) из линейного пространства векторных полей вида (2) образуют матричную алгебру Ли (т.е. линейное пространство матриц, замкнутое относительно коммутатора АВ-ВА).

В нашем случае базисные матрицы пятимерной алгебры зависят от трех вещественных параметров ($T, \varepsilon_1, \varepsilon_2$) и имеют следующий конкретный вид:

$$\begin{bmatrix} 0 & 2T\mu_2 & -2T^2\mu_2 & 1 \\ 2I\mu_1 T & 0 & 0 & 0 \\ 2I\mu_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -2T\mu_2 & 2T^2\mu_2 & I \\ 2v_1 T & 0 & 0 & 0 \\ 2v_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 2I\mu_2 & 0 & 0 & 0 \\ 0 & 4I\mu_2 & -2I\mu_2^2 & 1 \\ 0 & 2I\mu_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -(2\mu_2 - 2)T & 0 & 0 & 0 \\ 0 & -(4\mu_2 - 4)T & 2I\mu_2 & I \\ 0 & 2v_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\begin{bmatrix} -2I\mu_2^2 & 0 & 0 & 0 \\ 0 & -2I\mu_2^2 & 0 & 0 \\ 0 & 0 & -2I\mu_2^2 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

Здесь введена замена:

$$\mu_1 = 1 + 2\varepsilon_1, \mu_2 = 1 + 2\varepsilon_2, \nu_1 = 1 - 2\varepsilon_1, \nu_2 = 1 - 2\varepsilon_2. \quad (5)$$

Все обсуждения проводятся в вещественном линейном пространстве, образованном данными пятью матрицами.

2. Упрощение базиса

За счет нескольких матричных подобию, т.е. преобразований типа $E_k = S^{-1} * E_k * S$, $k = 1..5$, удастся привести две из пяти матриц любого базиса вида (4) к диагональной форме, а остальные – существенно упростить.

Ниже рассматривается случай $T \neq 0$, т.к. ситуация нулевого T является более простой и сводится (при фиксированных $\varepsilon_1, \varepsilon_2$) к единственной однородной поверхности

$$\nu(x_1, x_2, y_1, y_2) = \mu_1 x_1^2 + \nu_1 y_1^2 + \mu_2 x_2^2 + \nu_2 y_2^2. \quad (6)$$

Отметим также, что в задаче об однородности оба параметра $\varepsilon_1, \varepsilon_2$ положительны.

Итоговая матрица подобию S (при $T \neq 0$) и преобразованный матричный базис выписаны ниже:

$$S = \begin{bmatrix} 0 & \frac{1}{\nu_1} & 0 & 0 \\ -\frac{1}{2Tv_1} & 0 & T & 0 \\ -\frac{1}{2T^2\nu_1} + \frac{1}{T^2\mu_2\nu_1} & 0 & 1 & -\frac{I}{2T^2\mu_2} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

$$E_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -I\nu_1 & 0 & 0 & 0 \\ 0 & I\mu_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, E_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, E_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$E_4 = \begin{bmatrix} \mu_2 & 0 & 0 & 0 \\ 0 & 2\varepsilon_2 & 0 & 0 \\ 0 & 0 & -\nu_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, E_5 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (8)$$

Система дифференциальных уравнений, отвечающая матрицам нового базиса (8), имеет в соответствии с (1, 2) следующий вид (в комплексной форме):

$$\operatorname{Re}(-I\nu_1 z_1 \frac{\partial \Phi}{\partial z_2} + I\mu_1 z_2 \frac{\partial \Phi}{\partial w}) = 0, \operatorname{Re}(z_1 \frac{\partial \Phi}{\partial z_2} + z_2 \frac{\partial \Phi}{\partial w}) = 0,$$

$$\operatorname{Re}(z_1 \frac{\partial \Phi}{\partial w}) = 0, \operatorname{Re}(\mu_2 z_1 \frac{\partial \Phi}{\partial z_1} + 2\varepsilon_2 z_2 \frac{\partial \Phi}{\partial z_2} - \nu_2 w \frac{\partial \Phi}{\partial w}) = 0,$$

$$\operatorname{Re}(Iz_1 \frac{\partial \Phi}{\partial z_1} + Iz_2 \frac{\partial \Phi}{\partial z_2} + Iw \frac{\partial \Phi}{\partial w}) = 0. \quad (9)$$

Используя свойства дифференцирования функций комплексной переменной

$$\frac{\partial F}{\partial z_1} = \frac{1}{2} \left(\frac{\partial F}{\partial x_1} - i \frac{\partial F}{\partial y_1} \right), \frac{\partial F}{\partial z_2} = \frac{1}{2} \left(\frac{\partial F}{\partial x_2} - i \frac{\partial F}{\partial y_2} \right), \frac{\partial F}{\partial w} = \frac{1}{2} \left(\frac{\partial F}{\partial u} - i \frac{\partial F}{\partial v} \right), \quad (10)$$

перейдём к вещественной форме дифференциальных уравнений:

$$U_1 : \nu_1 \left(\frac{\partial F}{\partial x_2} y_1 - \frac{\partial F}{\partial y_2} x_1 \right) + \mu_1 \left(-\frac{\partial F}{\partial u} y_2 - x_2 \right) = 0,$$

$$U_2 : \frac{\partial F}{\partial u} x_2 + \frac{\partial F}{\partial x_2} x_1 + \frac{\partial F}{\partial y_2} y_1 - y_2 = 0, U_3 : \frac{\partial F}{\partial u} y_1 + x_1 = 0, \quad (11)$$

$$U_4 : \mu_2 \left(\frac{\partial F}{\partial x_1} x_1 + \frac{\partial F}{\partial y_1} y_1 \right) + 2\varepsilon_2 \left(\frac{\partial F}{\partial x_2} x_2 + \frac{\partial F}{\partial y_2} y_2 \right) - \nu_2 \left(\frac{\partial F}{\partial u} u - F \right) = 0,$$

$$U_5 : -\frac{\partial F}{\partial u} F - \frac{\partial F}{\partial x_1} y_1 - \frac{\partial F}{\partial x_2} y_2 + \frac{\partial F}{\partial y_1} x_1 + \frac{\partial F}{\partial y_2} x_2 - u = 0.$$

3. Интегрирование

Исследуя систему (11), будем решать по известным схемам [4] каждое её отдельное уравнение. Начнем с самого простого из уравнений – U_3 . Решая его относительно функции F , получаем:

$$F(x_1, x_2, y_1, y_2, u) = -\frac{x_1 u}{y_1} + F1(x_1, x_2, y_1, y_2), \quad (12)$$

Подставляя данное решение в оставшиеся уравнения системы, получаем новый её вариант из 4-х уравнений относительно 4-х переменных. Далее пошагово уменьшаем количество независимых переменных в системе, доводя до конца её решение. Результаты промежуточных вычислений, полученные с помощью пакета Maple, приведены ниже:

$$F1 = \frac{\frac{1}{2} y_2^2 (\mu_1 x_1^2 + \nu_1 y_1^2) - x_1 x_2 y_1 y_2 (\mu_1 - \nu_2)}{\nu_1 (x_1^2 + y_1^2) y_1} + F2(x_1, x_2, y_1), \quad (13)$$

$$F2 = \frac{\frac{1}{2} x_2^2 (\mu_1 y_1^2 + \nu_1 x_1^2)}{2 \nu_1 (x_1^2 + y_1^2) y_1} + F3(x_1, y_1), \quad (14)$$

$$F3 = \frac{F4(x_1) (x_1^2 + y_1^2)^{\frac{2\varepsilon_2}{\mu_2}}}{y_1}, \quad (15)$$

$$F4 = C1. \quad (16)$$

Собрав результаты интегрирования, получаем следующий вид уравнений искомых поверхностей:

$$v(x_1, x_2, y_1, y_2) = \frac{(\frac{1}{2} y_2^2 x_1^2 - x_1 x_2 y_1 y_2 + \frac{1}{2} x_2^2 y_1^2) \mu_1}{\nu_1 (x_1^2 + y_1^2) y_1} - \frac{x_1 u}{y_1} + \frac{\frac{1}{2} y_2^2 y_1^2 + x_1 x_2 y_1 y_2 + \frac{1}{2} x_2^2 x_1^2}{(x_1^2 + y_1^2) y_1} + \frac{C1 (x_1^2 + y_1^2)^{\frac{2\varepsilon_2}{\mu_2}}}{y_1}, \quad (17)$$

где $C1$ – произвольная константа.

В итоге уравнения поверхностей, отвечающих изучаемому семейству алгебр после упрощения (17) зависят от двух (вместо трех начальных) вещественных параметров и имеют (с точностью до аффинных преобразований) следующий окончательный вид:

$$|z_1|^2 \operatorname{Re}(z_1 \bar{w}^*) = \mu_1 (\operatorname{Im}(z_1 \bar{z}_2))^2 + \nu_1 (\operatorname{Re}(\bar{z}_1 z_2))^2 + C |z_1|^{\frac{4\varepsilon_2+2}{\mu_2}}, \quad (18)$$

где $w^* = 2\nu_1 w, C = 2C_1 \nu_1$.

4. Проверка полученного решения

Построенные поверхности действительно являются интегральными поверхностями для изучаемых алгебр. Чтобы убедиться в этом, достаточно рассмотреть однопараметрические группы аффинных преобразований, порождаемые экспонентами базисных матриц (8).

Напомним, что экспонента матрицы A вычисляется по следующей формуле:

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k. \quad (19)$$

Данный ряд всегда сходится, так что экспонента матрицы всегда корректно определена.

Вычислим экспоненту на примере первой матрицы базиса (8):

$$\begin{aligned} e^{tE_1} &= E + tE_1 + \frac{1}{2}(tE_1)^2 + \frac{1}{3!}(tE_1)^3 + \dots = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -I\nu_1 t & 0 & 0 & 0 \\ 0 & I\mu_1 t & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mu_1 \nu_1 t^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \\ &+ \frac{1}{3!} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \dots = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -I\nu_1 t & 1 & 0 & 0 \\ \frac{1}{2}\mu_1 \nu_1 t^2 & I\mu_1 t & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (20) \end{aligned}$$

Рассчитав аналогичным образом экспоненты остальных матриц, получаем матричную форму пяти однопараметрических групп аффинных преобразований:

$$\begin{aligned}
g_1(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ -Iv_1t & 1 & 0 & 0 \\ \frac{1}{2}\mu_1v_1t^2 & I\mu_1t & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, g_2(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ t & 1 & 0 & 0 \\ \frac{1}{2}t^2 & t & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
g_3(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ It & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, g_4(t) = \begin{bmatrix} e^{\mu_2t} & 0 & 0 & 0 \\ 0 & e^{2t\varepsilon_2} & 0 & 0 \\ 0 & 0 & e^{-v_2t} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
g_5(t) &= \begin{bmatrix} e^{It} & 0 & 0 & 0 \\ 0 & e^{It} & 0 & 0 \\ 0 & 0 & e^{It} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{21}
\end{aligned}$$

В координатной форме эти группы имеют вид:

$$\exp(tE_1): z_2 \rightarrow z_2 - Iv_1tz_1, w \rightarrow w + I\mu_1tz_2 + \frac{1}{2}t^2\mu_1v_1z_1, t \in R,$$

$$\exp(tE_2): z_2 \rightarrow z_2 + tz_1, w \rightarrow w + tz_2 + \frac{1}{2}t^2z_1, t \in R,$$

$$\exp(tE_3): w \rightarrow w + Itz_1, t \in R, \tag{22}$$

$$\exp(tE_4): z_1 \rightarrow \exp(\mu_2t)z_1, z_2 \rightarrow \exp(2t\varepsilon_2)z_2, w \rightarrow \exp(-v_2t)w, t \in R,$$

$$\exp(tE_5): z_1 \rightarrow \exp(It)z_1, z_2 \rightarrow \exp(It)z_2, w \rightarrow \exp(It)w, t \in R.$$

Легко видеть, что при любых $\varepsilon_1, \varepsilon_2$ такие преобразования сохраняют уравнение (18), что доказывает аффинную однородность любой из поверхностей построенного семейства.

Заключение

В ходе данной работы было проинтегрировано семейство матричных алгебр Ли и получены уравнения соответствующих гиперповерхностей 3-мерного комплексного пространства. Также была проведена проверка полученного решения и доказана однородность полученной поверхности. Все вычисления выполнялись с использованием математического пакета Maple 17. Результаты данной работы опубликованы в [5].

Список использованной литературы

1. Лобода, А. В. Примеры аффинно-однородных поверхностей $(\varepsilon_1, \varepsilon_2)$ типов в пространстве C^3 / А. В. Лобода, А. В. Швырева, А. В. Шиповская // Компл. анализ и его прил.: матер. рег. конф. (Брянск, 23-24 ноября 2016 г.). – Брянск, 2016. – С. 22-24.
2. Лобода, А. В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства / А. В. Лобода, А. С. Ходарев // Известия ВУЗов. Математика. – 2003. – № 10. – С. 38-50.
3. Назарьева, В. А. Интегрирование матричных алгебр Ли / В. А. Назарьева, Т. В. Новицкая // Сборник студенческих научных работ факультета компьютерных наук. – 2016. – № 10. – С. 112-117.
4. Понтрягин, Л. С. Обыкновенные дифференциальные уравнения / Л.С. Понтрягин. – М. : Наука, 1974. – 331 с.
5. Швырева, А. В. О явных уравнениях аффинно-однородных поверхностей $(\varepsilon_1, \varepsilon_2)$ -типов в C^3 / А. В. Швырева // Некоторые вопросы анализа, алгебры, геометрии и математического образования: матер. молод. межд. науч. конф. «Методы современного математического анализа и геометрии и их приложения» (Воронеж, 23-25 декабря 2016 г.). – Воронеж, 2016. – С. 328-329.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИНТЕЗА ТЕСТОВЫХ СПОСОБОВ ВОЗДЕЙСТВИЯ ДЛЯ СИСТЕМЫ ЗАЩИТЫ СЕТЕЙ РАДИОСВЯЗИ ОТ КОМПЬЮТЕРНЫХ АТАК

А. В. Щеглов
Студент
Е.Ю. Обущенко
Студент
В. Ю. Храмов
Профессор

Введение

В современном мире информационные технологии являются одним из наиболее развивающихся направлений человеческой деятельности. Вместе с их развитием растет и значимость информационных ресурсов. Информационные ресурсы являются объектом права собственности и не

менее ценным ресурсом, чем материальные или энергетические. Поэтому их защита признана сегодня наиболее актуальной проблемой. Словосочетание «компьютерные атаки» знакомо любому пользователю компьютера. Пользователи сетей радиосвязи имеют риск стать жертвами злоумышленников. Между тем, за общим и расплывчатым обозначением "компьютерные атаки" скрываются вполне конкретные действия взломщиков, незнание которых пользователем, значительно упрощает задачу злоумышленников.

Регулярно организаторами и пользователями систем радиосвязи применяются различные меры, направленные на защиту сетей радиосвязи от компьютерных атак, однако известны далеко не все уязвимости.

В данной статье рассматриваются возможности разработанного программного обеспечения (ПО) синтеза полного множества тестовых способов специальных программных воздействий для системы защиты сетей радиосвязи от компьютерных атак, другими словами выявляется полное множество возможных уязвимостей систем радиосвязи.

1. Анализ предметной области компьютерных атак

Компьютерная атака (удалённая) — информационное разрушающее воздействие на распределённую систему информационно-технических средств (ИТС). К компьютерным атакам будем относить программные и электромагнитные воздействия. Их различие в том, что воздействующим фактором в первом случае является полученная после выделения из электромагнитных или электрических сигналов совокупность цифровых данных, являющихся исходными для алгоритмов функционирования ИТС, а во втором случае — электромагнитный или электрический сигнал, несущий исходные данные для алгоритмов функционирования ИТС только в своих амплитудных, частотных, фазовых, временных и пространственных характеристиках. Компьютерным атакам подвержены системы сетей радиосвязи, а также распределённые системы, поскольку их компоненты обычно используют открытые каналы передачи данных, и нарушитель может не только проводить пассивное прослушивание передаваемой информации, но и модифицировать передаваемый трафик.

По характеру воздействия атаки делятся на две группы: пассивные и активные. Первый тип воздействий на распределённую вычислительную систему (РВС) представляет собой некоторое воздействие, не оказывающее прямого влияния на работу системы, но в то же время способное нарушить её политику безопасности. Отсутствие прямого влияния на работу РВС приводит именно к тому, что пассивное удалённое воздействие (ПУВ) трудно обнаружить. Возможным примером типового ПУВ в РВС служит прослушивание канала связи в сети. Активное воздействие на РВС — воздействие, оказывающее прямое влияние на

работу самой системы (нарушение работоспособности, изменение конфигурации РВС и т. д.), которое нарушает политику безопасности, принятую в ней. Активными воздействиями являются почти все типы удалённых атак. Связано это с тем, что в саму природу наносящего ущерб воздействия включается активное начало. Явное отличие активного воздействия от пассивного — принципиальная возможность его обнаружения, так как в результате его осуществления в системе происходят некоторые изменения. При пассивном же воздействии, не остается совершенно никаких следов.

Классификация компьютерных атак по цели воздействия:

- нарушение функционирования системы (доступа к системе),
- нарушение целостности информационных ресурсов (ИР),
- нарушение конфиденциальности ИР.

Признак, по которому производится классификация, по сути есть прямая проекция трех базовых разновидностей угроз — отказа в обслуживании, раскрытия и нарушения целостности. Также при выполнении компьютерных атак систему связи возможно перевести в состояние потери работоспособности, сниженной эффективности функционирования, управляемости или доступности для углубленного анализа источником воздействия.

Уровни эталонной модели взаимодействий открытых систем, на котором осуществляется воздействие:

- физический,
- канальный,
- сетевой,
- транспортный,
- сеансовый,
- представительный,
- прикладной.

В ходе проделанной работы отметим, что компьютерные атаки производятся на вышестоящих шести уровнях, используя физический уровень в качестве среды доставки воздействующего фактора.

2. Программное обеспечение синтеза тестовых способов

Для обеспечения безопасности используемой системы связи необходимо решить задачу разработки полного множества тестовых удаленных компьютерных атак для потенциально доступной злоумышленнику информации об алгоритмах функционирования системы.

На основе способа разработки тестовых удаленных информационно-технических воздействий на пространственно-распределенные системы [1] информационно-технических средств разработано ПО, синтезирующее

полное множество тестовых способов компьютерных атак на системы радиосвязи. Формирования множества тестовых способов основывается на понятиях локального и распределенного алгоритмов. На сегодняшний день их определения приведены в работе [2].

Возможности ПО:

- добавлять/удалять локальные алгоритмы функционирования информационно-технических средств;
- добавлять/удалять/перемещать/соединять элементы блок-схем локальных алгоритмов функционирования информационно-технических средств;
- задавать/изменять номенклатуру сообщений в элементах блок-схем локальных алгоритмов функционирования информационно-технических средств;
- задавать/изменять информационные элементы в сообщениях и указывать информационные элементы, используемые при получении полного множества тестовых способов компьютерных атак;
- формировать отчет о результатах работы ПО, содержащего информацию об анализируемой системе сетей радиосвязи, представление алгоритмических моделей процесса функционирования всех информационно-технических средств сетей радиосвязи в виде блок-схем, номенклатуру, содержание и установленные оператором параметры сообщений, используемых в системе, а также полученный список тестовых способов компьютерных атак на информационно-технические средства анализируемой системы сетей радиосвязи.

Для синтеза тестовых способов компьютерных атак необходимо задать информацию о характеристиках системы, передаваемых сообщениях в данной системе связи и их информационных элементах, существующих в каждом сообщении.

На рис. 1 представлен интерфейс окна «Настройки проекта».

На следующем этапе необходимо постороить алгоритмическую модель конкретной системы связи. Пример построения представлен на рис. 2.

Построение алгоритмической модели происходит по следующим правилам:

- каждому входящему сообщению в любом локальном алгоритме должно соответствовать исходящее сообщение в другом локальном алгоритме;
- передача сообщений в рамках одного локального алгоритма недопустима;
- циклы формируются только с использованием логического блока;
- локальный алгоритм должен иметь начальный блок;

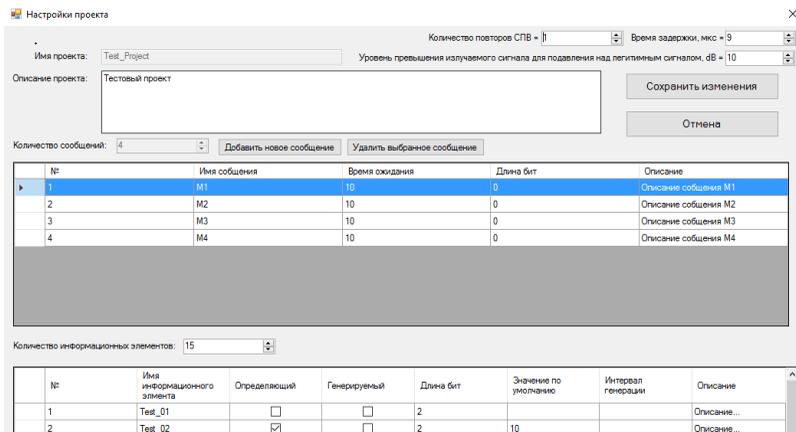


Рис. 1. Интерфес окна «Настройки проекта»

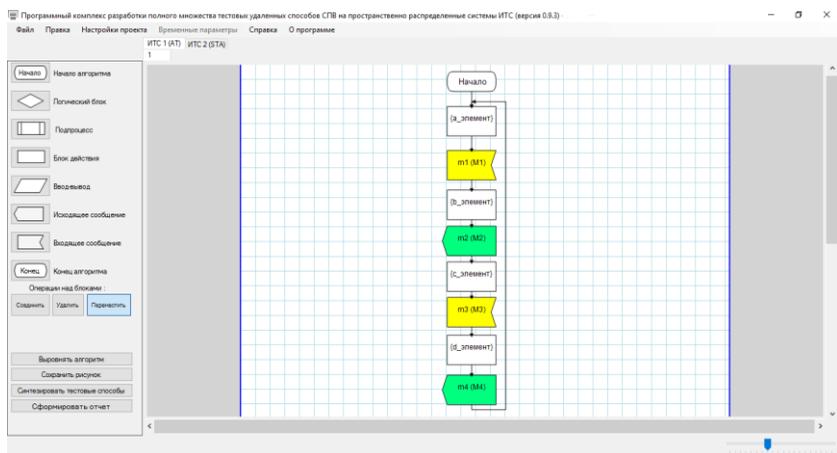


Рис. 2. Интерфес рабочего поля

- в случае отсутствия блока завершения алгоритма алгоритм должен быть зацикленным;
- на пути от возникновения информационного элемента сообщения до его использования может быть передача сообщений только между различными алгоритмами;
- возникновение и/или использование и/или уничтожение информационного элемента сообщения не должно осуществляться в процессе его передачи в сообщении;

– в каждом блоке может быть только один вход и один или несколько (условный блок) выходов; исключение составляют начальный блок, из которого существует только один выход, и конечный блок, в который может быть более одного входа.

В случае невыполнения вышеуказанных критериев программное средство отображает оператору сообщение о соответствующей ошибке.

Если построение локальных алгоритмов выполнено корректно, по нажатию клавиши «Сгенерировать отчет» запускается процесс анализа рассматриваемой системы, включающий построения локального и распределенного алгоритмов, после чего будет сгенерирован и открыт отчет в формате MS Word.

Важно отметить, в формируемом отчете содержится информация в виде обобщенного текстового описания компьютерных атак. Для блока реализации тестовых способов компьютерных атак генерируются специальные файлы бинарного типа, содержащие конкретные действия на машинном коде, производящие конкретную компьютерную атаку. Пример данного отчета изображен на рис 3.

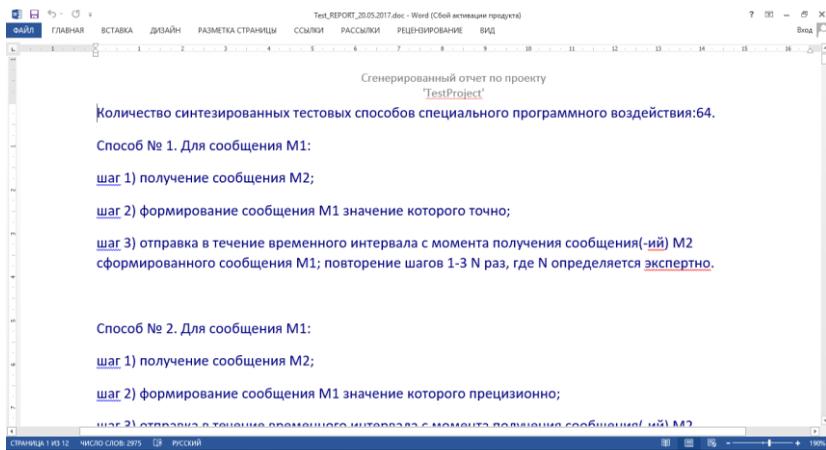


Рис. 3. Пример генерируемого отчета, содержащего описание компьютерных атак

3. Макет системы защиты сетей радиосвязи от компьютерных атак

С использованием разработанного программного обеспечения предложен макет системы защиты сетей радиосвязи от компьютерных атак, предназначенный для обнаружения, противодействия и предупреждения компьютерных атак на сети радиосвязи.

Возможности данной системы позволяют выявить уязвимости элементов сети на основе синтеза полного множества тестовых способов компьютерных атак для имеющейся информации об алгоритмах информационного воздействия элементов данной сети, а также позволяет восстанавливать работу сети радиосвязи в условиях компьютерных атак на основе анализа параметров функционирования сети.

Состав макета системы защиты сетей радиосвязи:

- Автоматизированное рабочее место (АРМ) синтеза тестовых способов компьютерных атак (включает ПО синтеза тестовых способов),
- АРМ управления предназначенный для обнаружения и противодействия компьютерным атакам,
- коммутатор,
- блок измерения и идентификации,
- блок реализации тестовых способов компьютерных атак.

В работе данного макета присутствуют ограничения по диапазону рабочих частот, по размеру защищаемой площади, скорости противодействия компьютерной атаки. Также отметим, вид требуемых генерируемых сигналов – радиосигнал со специальными сообщениями.

Заключение

В данной статье рассмотрено понятие компьютерных атак и их классификация, продемонстрирован функционал разработанного программного обеспечения синтеза тестовых способов компьютерных атак, а также предложен макет системы защиты сетей радиосвязи от компьютерных атак.

Исходя из этого можно сделать вывод, что решив задачу разработки полного множества тестовых удаленных компьютерных атак для потенциально доступной злоумышленнику информации об алгоритмах функционирования системы радиосвязи, безопасность систем радиосвязи возможно повысить, приняв меры по ее защите, описанные в выходных файлах, разработанного программного обеспечения.

Список литературы

1. Бойко А.А. Способ разработки тестовых удаленных информационно-технических воздействий на пространственно-распределенные системы информационно-технических средств. / А.А. Бойко, А.А. Дьякова. – Спб. : Защита информации, 2014 – 84 с.
2. Tel G. Introduction to Distributed Algorithms / G. Tel. – N. Y. : Cambridge University Press, 2001. – 612 p.

РАБОТЫ, ВЫПОЛНЕННЫЕ ИНОСТРАННЫМИ СТУДЕНТАМИ

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА НУМЕРОВА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ CUDA

А.И. Азиз

Студент

С.В. Борзунов

Доцент

Введение

В настоящее время решение актуальных научно-технических и прикладных задач, в их числе моделирование физических, химических, биологических и иных сложных процессов и объектов, не обходится без привлечения высокопроизводительных компьютерных систем, в которых применяются методы параллельного программирования. Такие компьютерные системы – суперкомпьютеры – используются во всех сферах, где применяется математическое моделирование; там, где требуется огромный объем вычислений, обработка большого количества данных в реальном времени, или решение задачи может быть найдено перебором множества значений исходных параметров. Для ускорения ресурсоемких расчетов в последние годы активно применяются возможности современных графических систем, реализующие массово-параллельные вычисления общего назначения на мощных графических процессорах. Графические процессоры (GPU) имеют свои особенности по сравнению с центральными процессорами (CPU), поскольку современные GPU представляют собой массивно-параллельные вычислительные устройства с большим количеством ядер и иерархически организованной собственной памятью. В связи с этим разработка и реализация вычислительных алгоритмов для систем с графическими процессорами является актуальной задачей.

1. Цель работы

Цель работы заключается в реализации метода Нумерова решения обыкновенных дифференциальных уравнений второго порядка с помощью технологии CUDA. Для достижения поставленной цели были поставлены и решены следующие задачи:

1. реализовать параллельную версию метода Нумерова, используя стандарт распараллеливания CUDA;

2. сравнить полученные результаты с известным точным решением;
3. определить достигнутое ускорение.

2. Архитектура графических процессоров и стандарт распараллеливания CUDA

Графические процессоры могут быть использованы для проведения вычислений общего назначения. По сравнению с центральным процессором, имеющим в большинстве современных неспециализированных компьютерных систем от 4 до 8 ядер, графический процессор может иметь вычислительный блок, состоящий из сотни вычислительных ядер и более (рис. 1) [1-3]. В связи с этим, GPU стали активно использоваться в качестве платформы для высокопроизводительных математических вычислений. Такая технология получила название GPGPU (General-purpose computing for graphics processing units) [1]. Поточковые процессоры имеют возможность выполнения независимых потоков или нитей (в иностранной литературе – threads). Такие процессоры объединяются в укрупненные множества – блоки (blocks) и сети (grids), которые имеют доступ к вышележащим уровням иерархической памяти.

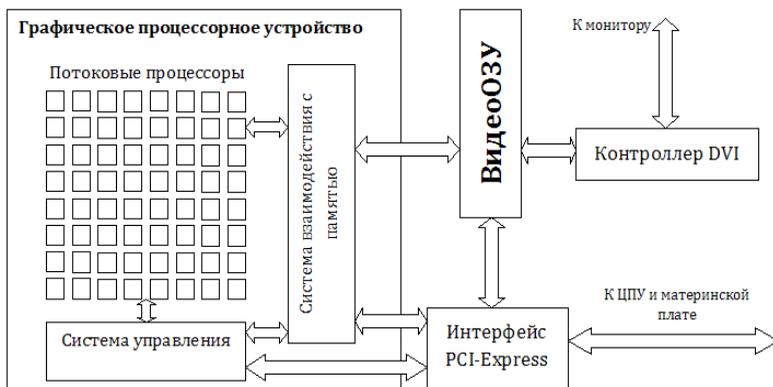


Рис. 1. Схематическое представление архитектуры GPU

Стандарт CUDA – это технология распараллеливания на программно-аппаратной вычислительной архитектуре NVIDIA, основанная на расширении языка Си. Она дает возможность организации доступа к определенному количеству инструкций графического ускорителя и управления его памятью при проведении параллельных вычислений [2]. В табл. 1 перечислены различные типы памяти, доступные нитям в стандарте CUDA для GPU [1].

Типы памяти в CUDA

Тип памяти	Доступ	Уровень выделения	Скорость работы
регистры (registers)	R/W	per-thread	высокая (on chip)
Local	R/W	per-thread	низкая (DRAM)
Shared	R/W	per-block	высокая (on-chip)
Global	R/W	per-grid	низкая (DRAM)
Constant	R/O	per-grid	высокая (on chip L1 cache)
Texture	R/O	per-grid	высокая (on chip L1 cache)

Расширения языка Си, вводимые в CUDA, состоят из:

1. спецификаторов функций, которые показывают, как выполняются функции и откуда они могут быть вызваны;
2. спецификаторов переменных, определяющих тип памяти, в котором сохраняются данные переменных;
3. директив, необходимых для запуска ядра, обозначающих как данные, так и иерархию потоков;
4. встроенных переменных, содержащих информацию о текущем потоке;
5. runtime, включающего в себя дополнительные типы данных.

В табл. 2 перечислены основные спецификаторы функций в стандарте CUDA, важные для организации вычислений с использованием графической системы.

Таблица 2

Спецификаторы функций в CUDA

Спецификатор	Выполняется на	Может вызываться из
<code>__device__</code>	device	device
<code>__global__</code>	device	host
<code>__host__</code>	host	host

3. Реализация метода Нумерова

Метод Нумерова является одним из важных методов численного решения дифференциальных уравнений второго порядка, активно исследуемых в физике:

$$y''(x) = -f(x)y(x).$$

К такому типу уравнений относится, в частности, уравнение Шредингера в квантовой механике.

Решение задачи Коши для уравнения указанного вида вычисляется на равномерной сетке. Обозначим шаг интегрирования через h . Число точек N на интервале изменения аргумента x представляет собой параметр сложности задачи. С увеличением N до величин порядка 10^5 – 10^6 время работы вычислительного алгоритма в последовательном случае составляет десятки секунд и более.

В соответствии с методом Нумерова очередное значение функции $y(x+h)$, являющейся решением задачи Коши, определяется на основании двух предыдущих значений:

$$y(x+h) = \frac{2 \left[1 - \frac{5}{12} h^2 f(x) \right] y(x) - \left[1 + \frac{h^2}{12} f(x-h) \right] y(x-h)}{1 + \frac{h^2}{12} f(x+h)}.$$

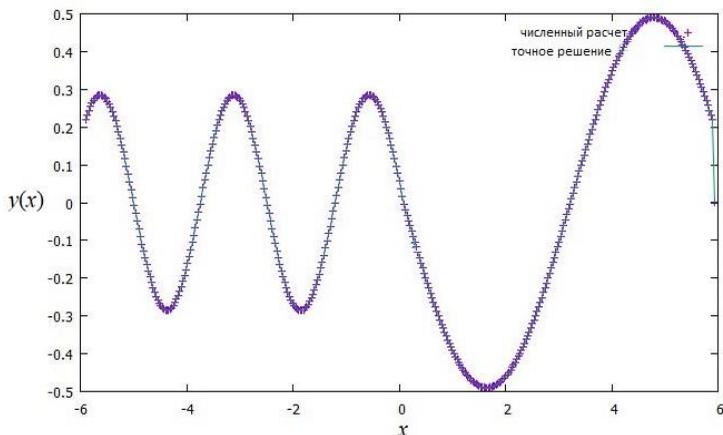
4. Результаты

В данной работе в качестве примера было получено решение задачи Коши для дифференциального уравнения вида $y''(x) - xy(x) = 0$. Для указанного уравнения решение можно записать в аналитическом виде через специальные функции, а именно через функции Эйри [4]. Результаты численного расчета в сравнении с точным решением представлены на рис. 2. Видно, что во всех точках рассматриваемого диапазона аргумента x имеется хорошее согласие с аналитическим решением. В табл. 2 приведены данные о времени выполнения вычислительной программы на CPU и с использованием ресурсов GPU, а также достигнутое на GPU ускорение для нескольких значений параметра сложности задачи.

Таблица 2

Время работы вычислительной программы и достигнутое ускорение в зависимости от параметра сложности задачи

Число точек N	Время расчета на CPU, с	Время расчета на GPU, с	Ускорение
10^4	1,048	1,233	0,850
$5 \cdot 10^4$	1,906	1,335	1,428
10^5	2,982	1,703	1,751



аналитическое решение – сплошная кривая, точное решение – крестики

Рис. 2. Сравнение аналитического и точного решения задачи Коши для дифференциального уравнения $y''(x) - xy(x) = 0$

Заключение

В работе реализован метод Нумерова численного решения обыкновенных дифференциальных уравнений с использованием графических процессоров. Показано, что применение стандарта распараллеливания CUDA позволило ускорить расчеты. Полученная реализация метода Нумерова может быть использована для физических и химических расчетов, в частности, для решения квантовомеханического уравнения Шредингера на суперкомпьютерах.

Список литературы

1. Сандерс Д. Технология CUDA в примерах. Введение в программирование графических процессоров / Д. Сандерс, Э. Кэндрот; пер. с англ. А.А. Слинкина. – М. : ДМК Пресс, 2011. – 232 с.
2. Gregg C. Where is the data? Why you cannot debate CPU vs. GPU performance without the answer / C. Gregg, K. Hazelwood // The IEEE International Symposium on Performance Analysis of Systems and Software, Austin, 10 April, 2011. – 2011. – P. 134-144.
3. Беседин К.Ю. Моделирование обработки запросов на гибридных вычислительных системах с многоядерными и приложения / К.Ю. Беседин, П.С. Костенецкий // Программные системы: теория и приложения. – 2014. – № 1. – С. 91-110.
4. NIST Handbook of Mathematical Functions / F. W. J. Olver [et al]. – Cambridge : Cambridge University Press, 2010. – 951 p.

ОСОБЕННОСТИ ФОРМИРОВАНИЯ УЧЕБНЫХ ГРУПП НА КУРСАХ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ МИНИСТЕРСТВА ОБРАЗОВАНИЯ РЕСПУБЛИКИ ИРАК

Я.Х.А. Алубади

Студент

И.В. Абрамов

Доцент

Введение

В республике Ирак много различных иностранных предприятий, как западных стран, Америки, в т. ч. России, так и восточных стран. В связи с крупными нефтяными природными запасами нефти присутствуют предприятия нефтяной промышленности. На рынке товаров и услуг много компаний торгующих лекарствами, мебелью, электронными устройствами.

В интересах повышения трудовой занятости населения Министерство образования Ирака активно занимается обучением и повышением знаний людей своей страны. Министерство образования организовало курсы повышения квалификации, которые обеспечивают возможность людям поступать на работу и справляться с производственными обязанностями. Для организации курсов повышения квалификации государство выделяет специальные средства. Но, кроме этого, Министерство образования заключает и коммерческие договора с разными иностранными компаниями, которые готовы оплачивать повышение квалификации своих специалистов.

1. Анализ требований Министерства Образования и Заказчиков

У Министерства Образования Ирака и у Заказчиков (организации, производственные предприятия), наряду с общими интересами, существуют свои требования/пожелания к участникам курсов повышения квалификации и к процессам их организации/проведения [1,2].

Заказчики в основном предъявляют следующие требования:

- развивать самоэффективность;
- перейти на новую ступень функционала (новые знания);
- освоить коллективные действия;
- получить Сертификат ICDL (International Computer Driving Licence);
- получить документ для повышения ранга (документ позволяет повышать оплату труда и подниматься по карьерной лестнице).

Министерство Образования предъявляет свои требования к слушателям:

- наличие стажа работы;
- хорошее зрение;
- к возрасту;
- знание компьютера;
- потребность в определенных учебных материалах.

2. Содержание и формы проведения учебных занятий на курсах повышения квалификации

Министерство Образования, основываясь на исследованиях текущих потребностей потенциальных Заказчиков, разработало и предлагает ряд учебных курсов, которые, в общем, отражают следующее:

- получение знаний о прикладных программах (*конкретизация*);
- получение знаний о компьютерной технике :
 - а) о компьютерах (серверы, рабочие станции, ноутбуки, планшеты, суперкомпьютеры и др.);
 - б) о компьютерной периферии (принтеры, сканеры, ИБП, плоттеры и др.);
 - в) о сетевом оборудовании (модемы, коммутаторы, маршрутизаторы, кабели, разъемы и др.).

Для установления взаимопонимания Министерство Образования предлагает, а Заказчики выбирают, не только содержание учебных курсов, но и формы обучения по различным критериям.

Формы обучения:

- курсы и семинары (по месту работы или за его пределами);
- наставничество;
- инструктаж и консультации (без отрыва от работы);
- самообучение (под контролем);
- заочное обучение.
- Критерии выбора форм методов обучения:
- время и место;
- необходимое оборудование;
- стоимость;
- цели обучения;
- требования к целевой группе;
- продолжительность и последовательность обучения;
- формы оценки и сертификации.

По приведенным критериям Заказчики удовлетворяют свои потребности и минимизируют свои затраты. По этим же критериям Министерство Образования повышает интеллектуальный уровень населения страны, обеспечивает специалистами рабочие места на рынке труда (как у себя в Министерстве, так и на предприятиях страны,

обеспечивая их эффективность за счет повышения компетентности сотрудников этих предприятий), а также имеет возможность пополнять свой бюджет.

3. Требования к слушателям курсов повышения квалификации

Министерство Образования Ирака предъявляет требования к своим сотрудникам, обучающим на курсах повышения квалификации специалистов Заказчиков, о формировании учебных групп. При этом сотрудникам Министерства приходится учитывать многие из перечисленных выше факторов (рис. 1).

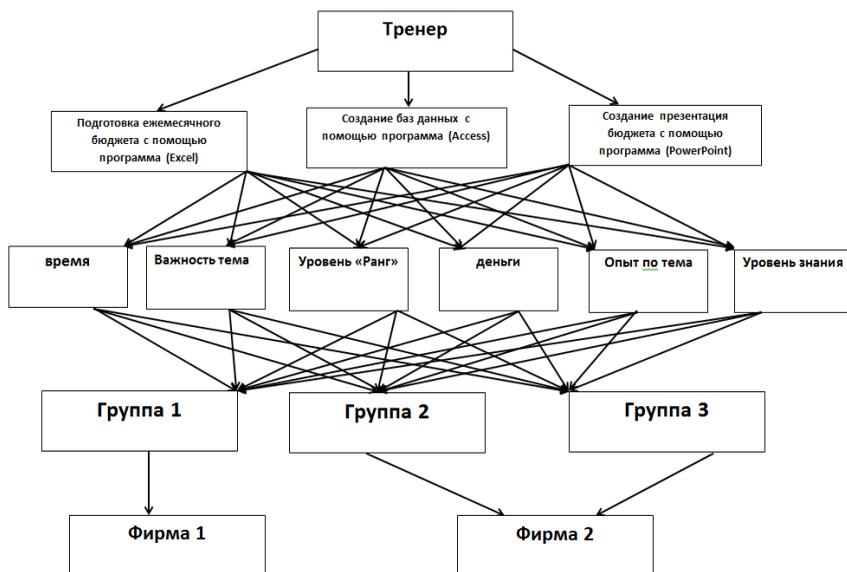


Рис. 1. Технологический процесс формирования сотрудниками Министерства Образования учебных групп Заказчиков

При организации курсов повышения квалификации [3,4] Министерство Образования учитывает мнение заинтересованных сторон. В качестве заинтересованных сторон могут выступать – Иракское правительство, Заказчики, чьи специалисты получают знания на курсах повышения квалификации, Спонсоры Заказчиков и Министерства Образования, и другие.

4. Разработка технологии организации курсов повышения квалификации

Исходная точка отсчета начала организации курсов – получение Заказов на обучение от Заказчиков. Эта ситуация отражена на рис. 2.

При получении Заказов на обучение Заказчики высказывают свои пожелания. Удовлетворить эти пожелания/требования позволяют темы, которые разработаны и могут корректироваться в блоке «Разработка ТЕМ курсов» (рис. 2). Далее при разработке конкретного курса повышения квалификации необходимо определить формы методов обучения. Среди организационных вопросов необходимо учесть выбор учебных тем по различным критериям, что отрабатывается в блоке «Учет специфики ТРЕБОВАНИЙ к уч. Курсам/Слушателям МИНИСТЕРСТВА» алгоритма. Все перечисленное реализовано в разделе «РЕАЛИЗАЦИЯ учебных ТЕМ» (рис. 2).

Разработанный курс повышения квалификации для конкретного Заказчика анализируется им и может подвергаться коррекции из-за дополнительных требований, которые могут возникнуть в результате анализа, например:

- обучение эффективным коммуникациям в конкретной предметной деятельности организации;
- учет специфики работы организации в разных направлениях деятельности;
- не просто групповая работа, а динамика групповой работы;
- обучение способам появления энтузиазма в коллективе;
- повышение уровня способности самооценки.

Такие возможные действия при организации курсов повышения квалификации должны быть обеспечены и отражены в алгоритме в новом разделе – «УСОВЕРШЕНСТВОВАНИЕ учебного материала».

Министерство Образования стремится предоставлять качественные услуги по повышению квалификации слушателей. Поэтому оно активно учитывает свой опыт предоставления таких услуг по разным направлениям деятельности. Но кроме этого необходимо постоянно вести мониторинг процессов предоставления услуг и корректировать их в динамике, не дожидаясь недовольства Заказчиков, которые обязательно скажутся на репутации Министерства Образования. Обеспечение необходимых действий по мониторингу предоставления услуг на курсах повышения квалификации приведены в алгоритме в блоке «МЕТОДИЧЕСКИЙ мониторинг».

При возникновении критической ситуации сразу должны возникать вопросы – ПОЧЕМУ она возникла? ЧЕМ вызвано недовольство Заказчика? КОГДА возникла критическая ситуация? негативные

УСЛОВИЯ проведения курсов ? Результатом последних действий должен быть раздел – «Повышение КАЧЕСТВА МЕТОДИКИ обучения» (рис. 2).

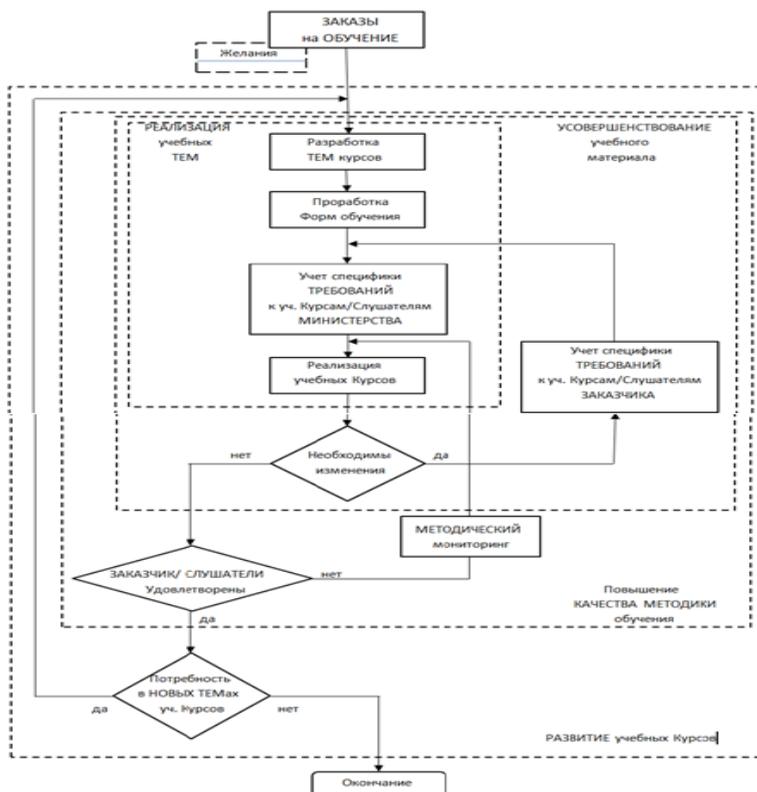


Рис. 2. Алгоритм повышения качества организации обучения

Заключение

Анализ потребностей и взаимоотношений участников рынка товаров и услуг в сегменте повышения знаний в области информационных технологий показывает общие потребности и их общее решение для достижения своих Целей (рис. 3):

Заказчики – минимум затрат + максимум знаний для своей организации;

Министерство Образования – максимум качества предоставляемых услуг.

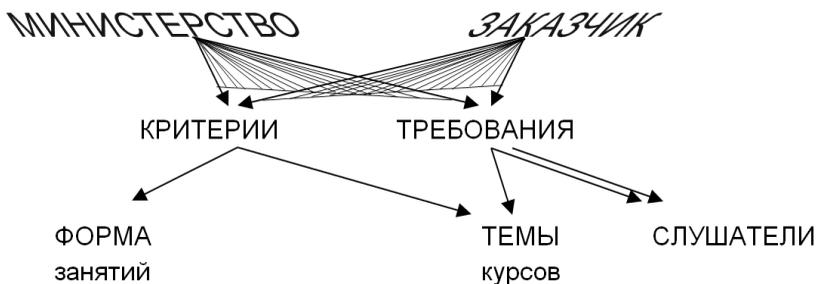


Рис. 3. Требования/желания Заказчиков и Министерства Образования

Многоаспектный процесс установления отношений «Министерство – Заказчик», а также разнообразие курсов повышения квалификации для сотрудников Заказчика, не только с точки зрения тематики, но и с точки зрения уровня знаний слушателей курсов, требуют наличия технологии процесса организации курсов, которая должна учитывать следующее:

- установление взаимопонимания и «товарно-денежных» отношений;
- учет текущих потребностей Заказчика во время обучения слушателей на курсах;
- оценку значимости каждого слушателя с учетом его знаний и целей его работодателя;
- формирование учебных групп в условиях многокритериальности и автоматизацию этого процесса для лиц, принимающих решение (ЛПР).

Список литературы

1. Жариков, О.Н. Системный подход в управлении / О.Н. Жариков, В.И. Королевская, С.Н. Хохлов. – М. : ЮНИТИ-ДАНА, 2001. – 62с.
2. Аксенова О.А. Организация корпоративной системы обучения. Проблемы повышения квалификации кадров и инвестиций в человеческий капитал / О.А.Аксенова; Санкт-Петербургский гос. политехнический ун-т. – СПб.: Изд-во Политехн. ин-та, 2005. – С. 54-55.
3. Арутюнов В. Инновации и система корпоративного образования / В. Арутюнов, Л. Стрекова, С. Цыганов // Высшее образование в России. – 2010. – № 1. – С. 29-39.
4. Веснин В.Р. Менеджмент/ В.Р. Веснин – М. : Проспект, 2004. – 504 с.

КОРРЕЛЯЦИОННЫЕ ЗАВИСИМОСТИ МЕЖДУ АЛЬТЕРАЦИЕЙ ЗУБЦА R ЭКГ и RR-ИНТЕРВАЛАМИ

Аль-Афаре Хазим

Студент

Я.А. Туровский

Доцент

Введение

Одним из современных методов оценки взаимосвязи работы сердца и нервной системы, во многом обеспечивающим регуляторные влияния является оценка variability сердечного ритма (BCP) [1]. Волны variability сердечного ритма несут информацию о контурах регуляции сердца со стороны нервной и гуморальной систем [2]. Таким образом, представляется актуальным изучение алгоритмов позволяющих оценить и расширить представление о тех или иных волновых процессах variability сердечного ритма.

1. Амплитудные характеристики зубца R и RR интервалов

Обычно variability сердечного ритма (BCP) рассчитывается путём анализа длительности интервала между последовательными сердечными сокращениями основываясь на ЭКГ (в аспекте интервалом между зубцами R (рис.1)) или кривых артериального давления, фотоплетизмография [3]. Частота сердцебиения может изменяться в широких пределах, в зависимости от функционального состояния организма.

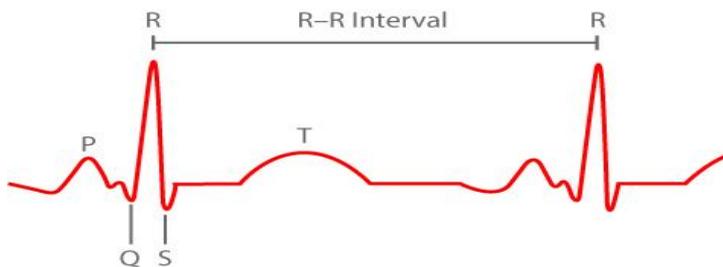


Рис. 1. Схема нормальной ЭКГ для RR- интервал

Известно, что одними из волн на кривой BCP являются т. н. дыхательные волны, вязанные с процессами вдоха и выхода. Так же известен феномен «альтерации зубца R», так же связанный с дыханием.

Актуальным представляется рассмотреть связь этих двух показателей в аспекте их взаимозаменяемости и, следовательно, сокращения числа датчиков на испытуемом.

В качестве базы данных мы использовали файлы ЭКГ взятые из (MIT-VIN) аритмичности [4]. Базы данных содержит 50 минут записи заметки включают в себя ЭКГ с частотой дискретизации 360 Гц (рис. 2).

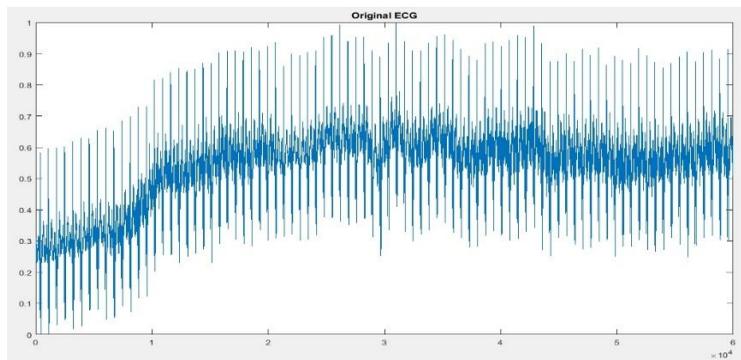


Рис. 2. Исходный сигнал ЭКГ

После этого, данные подвергались процедуре удаления артефактов фильтрацией по быстрому прямому и обратному преобразованию Фурье.

Затем анализировалась последовательность из не менее, чем 50 кардиоциклов (рис. 3 и 4).

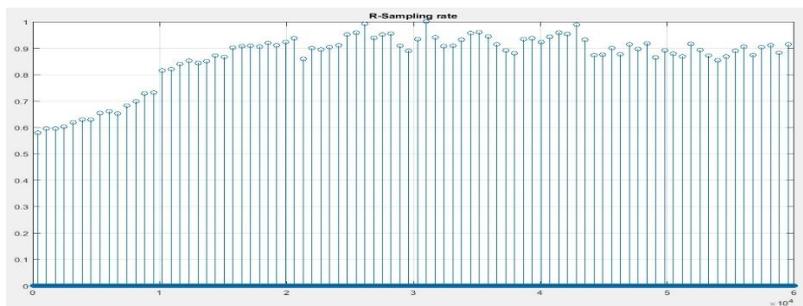


Рис. 3. Выборка R-зубца

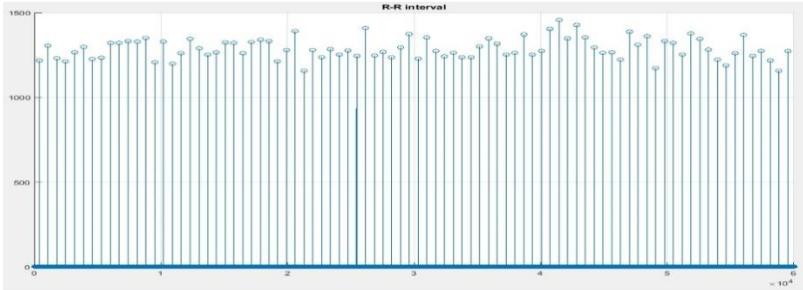


Рис. 4. Выборка RR- интервал

Соотносились результаты амплитуды зубца R , рассчитанные от изолинии и RR интервала. Таким образом, анализировалась связь дыхательной аритмии с феноменом альтерации зубца R. Для этой оценки использовался коэффициент корреляции Пирсона и коэффициент корреляции Спирмена для каждого (R-зубца, RR- интервал) в соответствии со следующими математическими формулами [5]:

Коэффициент Пирсона:

$$P_{x,y} = \frac{\text{cov}(x, y)}{\sigma_x \cdot \sigma_y}. \quad (1)$$

Коэффициент Спирмена:

$$r = 1 - \frac{6 \cdot \sum d^2}{n(n^2 - 1)}. \quad (2)$$

Некоторые результаты представлены в таблице.

Таблица

Корреляция Пирсона и Спирмена (R- зубец, RR-амплитуды)

№	name patient	name file	age	sex	ECG date	smoker	Pearson Correlation (R-RR)	Spearman Correlation (R-RR)
1	patient001	s0014lre	81	f	10/17/1990	no	0.238*	0.312*
2	patient002	s0015lre	58	f	10/17/1990	yes	0.251*	0.164
3	patient001	s0016lre	81	f	10/18/1990	no	0.470**	0.310**
4	patient003	s0017lre	63	m	10/18/1990	yes	-0.009	-0.007
.
48	patient022	s0066lrem	43	m	12/3/1990	yes	0.217	0.243*
49	patient020	s0069lrem	54	f	12/5/1990	no	0.508**	0.449**
50	patient021	s0073lrem	67	m	12/10/1990	yes	0.450**	0.494**

Заключение

В работе проанализирована корреляция между амплитудой зубца R и RR интервалов. Полученные значения позволяют утверждать, что коэффициент корреляции достаточно значим, однако не велик, что не позволяет использовать альтернацию зубца R, в качестве независимого показателя дыхательной ритмии.

Список литературы

1. Яблучанский Н.И. Вариабельность сердечного ритма в помощь практическому врачу. / Н.И. Яблучанский, А.В. Мартыненко. – Харьков : КНУ, 2010. – 131 с.
2. Kobayashi H., Park B.J, Miyazaki Y. Normative references of heart rate variability and salivary alpha amylase in healthy young male population / H. Kobayashi, B.J. Park, Y. Miyazaki // Journal of Physiological Anthropology. – 2012. – Vol. 31. – № 9. – P. 61-75.
3. Фрид М. Кардиология в таблицах и схемах / М. Фрид, С. Грайнс; пер. с англ. – М. : Практика, 1996. – 728 с.
4. Akselrod S. Power spectrum analysis of heart rate fluctuations control / S. Akselrod, D. Gordon. – New York : Wiley, 2010. – 672 p.
5. Eisenberg J. Heart Rate Variability during a Continuous Performance Test in Children with Problems of Attention / J. Eisenberg, R. Richman // The Israel Journal of Psychiatry and Related Science. – 2011. – Vol. 48. – № 1. – P. 19-24.

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ДЛЯ ОРТОГОНАЛЬНЫХ ПОЛИНОМОВ

Ф.А. Аль-Судани

Студент

А.В. Флегель

Доцент

Введение

Состояние современной вычислительной техники предполагает активное использование параллельных технологий при выполнении численных расчетов в различных научных и прикладных задачах. Часто наиболее эффективными архитектурами вычислительных систем являются реализации суперкомпьютеров как с распределенной памятью между различными вычислительными узлами системы, так и с доступом к общей памяти для вычислительных ядер в рамках одного узла. Как правило,

параллельное программирование на суперкомпьютерах использует универсальные подходы для систем с распределенной памятью, реализованные, например, с помощью функций библиотеки MPI (Message Passing Interface) [1, 2]. Однако, для задач, допускающих разделение на слабо взаимодействующие мелкие подзадачи, возможен более гибкий подход с совместным использованием MPI и технологии программирования OpenMP для систем с общей памятью [1-3]. Пример такого «гибридного» параллельного программирования рассмотрен в [2, 4]. Важность развития параллельных алгоритмов для систем с общей памятью также обусловлена повсеместным использованием многоядерных процессоров (в том числе и в персональных компьютерах).

В данной работе рассматривается параллельное вычисление классических ортогональных полиномов (на примере полинома Эрмита [5]) на многоядерном процессоре с использованием технологии OpenMP. Ортогональные полиномы широко используются в различных математических и физических задачах, поэтому их вычисление может являться одной из подзадач для реализации в рамках одного узла с общей памятью. Стоит отметить, что вычисление ортогонального полинома для фиксированного аргумента и порядка на одном ядре (без использования параллельности) занимает достаточно малое время (микросекунды), однако многократное вычисление функции для различных порядков и аргументов может привести к заметной потере времени вычислений общей задачи.

1. Полином Эрмита. Определение и методы вычисления

Ортогональным полиномом $P_n(x)$ порядка n называется полином степени $n = 0, 1, 2, \dots$, удовлетворяющий условию ортогональности на интервале (a, b) с неотрицательной весовой функцией $w(x)$:

$$\int_a^b P_n(x)P_m(x)w(x)dx = 0, \quad n \neq m. \quad (1)$$

Для полинома Эрмита $H_n(x)$ в (1): $w(x) = \exp(-x^2)$, $(a, b) = (-\infty, \infty)$.

Для малых n эффективным методом вычислений ортогональных полиномов является использование рекуррентных соотношений, которые для $H_n(x)$ имеют вид [5]:

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \quad H_0(x) = 1, \quad H_1(x) = 2x. \quad (2)$$

Для больших порядков n время вычисления последовательности полиномов $H_k(x)$ с индексами $k \leq n$ растет экспоненциально $\propto 2^n$.

Другой способ вычисления ортогональных полиномов – представление конечными суммами. Для полиномов Эрмита имеем [5]:

$$H_n(x) = n! \sum_{m=0}^{\lfloor n/2 \rfloor} (-1)^m \frac{(2x)^{n-2m}}{m!(n-2m)!}. \quad (3)$$

Формулу (3) удобно представить в виде:

$$H_n(x) = (2x)^n \sum_{m=0}^{\lfloor n/2 \rfloor} (-1)^m T_m^{(n)}(x), \quad (4)$$

где

$$T_m^{(n)}(x) = \prod_{k=1}^m A_k^{(n)}(x) = T_{m-1}^{(n)}(x) A_m^{(n)}(x), \quad T_0^{(n)}(x) = 1, \quad (5)$$

$$A_m^{(n)}(x) = (n-2m+1)(n-2m+2)/(2mx). \quad (6)$$

Нетрудно заметить, что сложность алгоритма вычисления $H_n(x)$ по схеме (4)-(6) с использованием перемножения элементов $A_k^{(n)}(x)$ в (5) растет как $\propto n^2$ при $n \gg 1$, и как $\propto n$ при использовании рекуррентной зависимости между слагаемыми $T_m^{(n)}(x)$.

Схема вычислений полиномов Эрмита в соответствии с уравнениями (4)-(6) будет использоваться в качестве базового (последовательного) алгоритма для дальнейшего распараллеливания.

2. Параллельные алгоритмы

Принципиальной сложностью параллельного вычисления ортогональных полиномов является рекурсивный характер слагаемых при представлении полиномов суммами. Так прямое использование рекурсии (5) для слагаемых в сумме (4) или соотношения (2) непосредственно для полиномов Эрмита не может быть реализовано параллельно ввиду зависимости вычислений на некотором шаге от результатов вычислений на предыдущем шаге. В связи с этим ниже предлагаются следующие алгоритмы распараллеливания.

1. Параллельно вычисляются слагаемые $T_m^{(n)}(x)$ в сумме (4), каждое из которых вычисляется последовательно прямым произведением множителей $A_m^{(n)}(x)$. Вычисление множителя $(2x)^n$ и массива элементов $A_m^{(n)}(x)$ также осуществляется параллельно.

2. Второй параллельный алгоритм заключается в следующем. После параллельного вычисления элементов массива $\{A_m^{(n)}(x)\}$, формируется массив $\{B_m^{(n)}(x)\}$ (также параллельно) из произведений $A_m^{(n)}(x)$ с числом сомножителей, равному числу P параллельных потоков:

$$B_m^{(n)}(x) = \prod_{p=m}^{m+P} A_p^{(n)}(x), \quad m = 1, \dots, \lfloor n/2 \rfloor - P. \quad (7)$$

С использованием элементов массива $\{B_m^{(n)}(x)\}$ на каждом параллельном потоке может быть реализовано рекуррентное вычисление слагаемых $T_m^{(n)}(x)$ с шагом P :

$$T_{m+P}^{(n)}(x) = T_m^{(n)}(x) B_m^{(n)}(x). \quad (8)$$

При этом для первых P слагаемых $T_m^{(n)}(x)$ с $m = 1, \dots, (P-1)$ по-прежнему вычисляется произведение элементов $A_m^{(n)}(x)$ в рамках одного потока.

3. В случае необходимости вычисления полиномов Эрмита на заданном интервале значений x имеет смысл отказаться от параллельного вычисления полинома в каждой фиксированной точке интервала. Вместо этого эффективно распараллелить по области определения функции введением сетки $x_i = i\Delta$ с вычислением $H_n(x_i)$ на отдельных ядрах процессора [2] (при этом выбираемая сетка для аргумента функции не обязана быть равномерной).

Описанные выше параллельные алгоритмы реализованы на языке C++ с использованием директив компилятора OpenMP. Реализация не имеет ограничений по отношению к используемой операционной системе. Ниже приводится анализ эффективности параллельных алгоритмов.

3. Анализ эффективности распараллеливания

На рис. 1 представлено ускорение первого описанного в предыдущем разделе параллельного алгоритма. Ускорение $S(P)$ для числа P вычислительных ядер определяется как отношение времени t_p вычислений на P потоках ко времени вычислений t_1 на одном потоке: $S(P) = t_p / t_1$. Идеальному случаю 100% эффективности распараллеливания соответствует значение $S(P) = P$. Как видно из рисунка, выигрыш во времени вычислений при использовании параллельности наблюдается только для больших значений порядка полинома (кривая для $n = 100$ на рис. 1). Данный факт объясняется ростом вычислительной сложности задачи с ростом n . Для небольших значений n программное время на организацию параллельных областей не перекрывается выигрышем от использования нескольких вычислительных ядер. Вычислительное время t_1 для описанного алгоритма составляет $t_1 = 22$ (мкс) для $H_{100}(0.7)$. С целью получения стабильных результатов для времени вычислений и 100% загрузки вычислительных ядер вычисления проводились последовательно 10^6 раз.

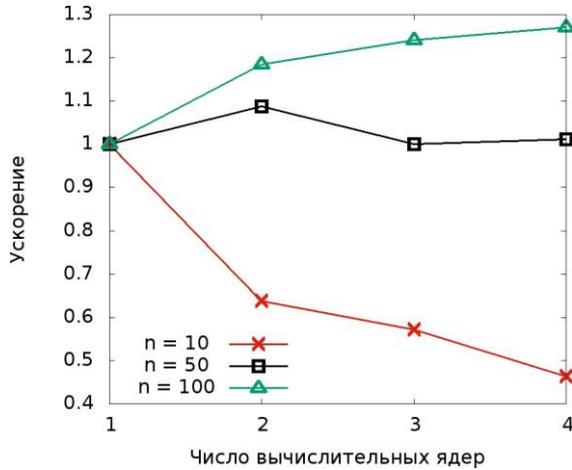


Рис. 1. Ускорение вычислительной части кода для расчета полиномов Эрмита $H_n(x)$ с $x=0.7$ и $n=10, 50, 100$ в соответствии с первым алгоритмом (см. текст)

На рис. 2 представлено ускорение второго параллельного алгоритма с использованием рекуррентного соотношения (8). При вычислении на одном ядре ($P=1$) этот алгоритм на порядок величины эффективнее первого: $t_1=2.4$ (мкс) для $H_{100}(0.7)$. Однако ввиду меньшей вычислительной сложности эффективность распараллеливания существенно хуже. Так для представленных на рис. 2 полиномов выигрыша во времени вычислений на нескольких ядрах получить не удалось.

Эффективность распараллеливания по интервалу значений аргумента полинома $H_n(x)$ для $x \in [0,2]$ показана на рис. 3, где представлены результаты для ускорения $S(P)$ для различных дискретизаций x . Из рисунка видно, что для любых размеров N сетки разбиения области $x \in [0,2]$ наблюдается рост $S(P)$ с увеличением P . При этом масштабируемость алгоритма тем лучше, чем больше размер сетки, что обусловлено ростом вычислительной сложности задачи. Высокая эффективность распараллеливания по области определения функции обусловлена независимостью вычислений отдельных значений $H_n(i\Delta)$.

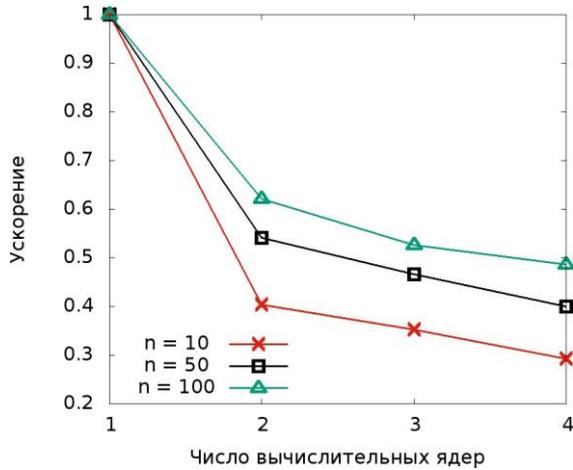


Рис. 2. То же, что на рис. 1, но для второго алгоритма

Заключение

В данной работе исследована эффективность параллельного вычисления классического ортогонального полинома Эрмита. В целом ввиду относительно малой алгоритмической сложности и принципиально непараллельной рекурсивной структуре слагаемых в представлении полиномов конечными суммами эффективность параллельных алгоритмов для вычисления полинома с фиксированным аргументом оказалась невысокой. Однако при необходимости вычисления функции на заданном интервале распараллеливание по области значений аргумента оказывается эффективным и хорошо масштабируемым, что связано с независимостью вычислений на параллельных ядрах. Описанные алгоритмы и их анализ справедливы для всех классических ортогональных полиномов.

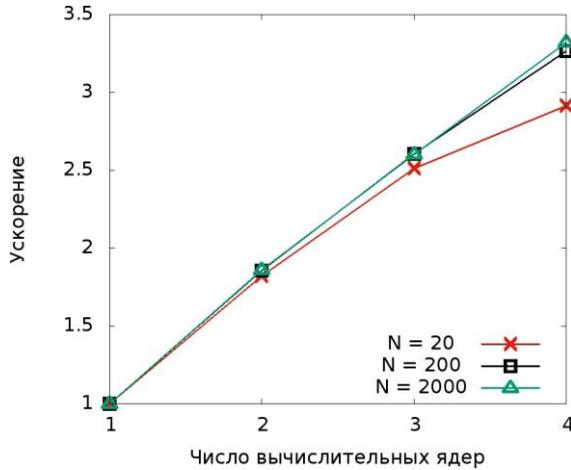


Рис. 3. Ускорение вычислительной части кода для расчета значений полинома Эрмита $H_{100}(i\Delta)$ на отрезке $[0,2]$ с шагом $\Delta = 2/N$ для указанных на рисунке значений N

Список литературы

1. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – Спб. : БХВ-Петербург, 2002. – 608 с.
2. Борзунов С.В. Практикум по параллельному программированию / С.В. Борзунов, С.Д. Кургалин, А.В. Флегель. – Спб. : БХВ, 2017. – 236 с.
3. The OpenMP API specification for parallel programming [Электронный ресурс]: сайт разработчиков OpenMP. – Режим доступа : <http://www.openmp.org>.
4. Карелин К.Н. Параллельные алгоритмы при моделировании электрон-атомного рассеяния в лазерном поле / К.Н. Карелин, А.В. Флегель // Вестник ВГУ. Сер. Системный анализ и информационные технологии. – Воронеж, 2008. – № 1. – С. 26-32.
5. Абрамовиц М. Справочник по специальным функциям / М. Абрамовиц, И.А. Стиган. – М. : Наука, 1979. – 832 с.

ПРОЦЕСС УПРАВЛЕНИЯ ТРУДОВЫМИ РЕСУРСАМИ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТОВ ORACLE

И. А. Мохаммед

Студент

И. В. Илларионов

Доцент

Введение

На сегодняшний день руководство любого развивающегося предприятия рано или поздно приходит к необходимости систематизации информации и автоматизации процессов управления персоналом. Процесс управления трудовыми ресурсами является непростой задачей, потому что он требует формирования единой кадровой политики, оказывающей влияние на конкурентоспособность и перспективы развития предприятия. Эффективное управление человеческим капиталом (HCM) в крупных компаниях не представляется возможным без использования информационных систем, так как в противном случае реализация процессов HCM является достаточно трудоемкой и занимает много времени. Таким образом, ни одна крупная компания не в состоянии оптимально организовать процесс трудовой деятельности без внедрения информационной системы управления персоналом.

1. Oracle HCM (Human Capital Management)

Среди современных систем управления персоналом необходимо выделить Oracle HCM (Human Capital Management). Oracle HCM – это компонент полнофункционального комплекса бизнес-приложений (ERP) Oracle E-Business Suit; система управления персоналом, в основе которой лежит функция оптимизации процессов кадровой службы [1]. Ее инструментарий позволяет осуществлять планирование процессов управления человеческими ресурсами, решать кадровые задачи разного характера и уровня сложности.

2. Цель эксперимента

Основной целью исследования является комплексный анализ Oracle HCM, его возможностей, направлений развития и сфер применения с учетом международной практики управления персоналом. Для достижения поставленной цели необходима реализация следующих задач:

охарактеризовать основные и специальные возможности Oracle HCM, проанализировать перспективы будущего развития системы, а также изучить особенности применения сервиса в Ираке. Объектом исследования является развитие и применение облачных сервисов в организации трудовой деятельности, в то время как предметом исследования выступает управление человеческим капиталом предприятий с помощью сервиса Oracle HCM.

3. Oracle Timetable

Интегрированная система Oracle Табельный учет (Oracle TimeTable) задействована в управлении данными по планированию и использованию рабочего времени в рамках всей компании. На базе Oracle Табельный учет руководство отдела кадров создает в системе единое хранилище данных учета времени сотрудников, обеспечивая контролируемый доступ к хранилищу. Преимущество системы состоит в том, что она занимается сбором и обработкой информации о количестве обработанных часов и заработной плате, устраняет необходимость многократного ручного ввода данных и позволяет настраивать нормативы использования рабочего времени в форме графиков, смен и календарей, куда включена информация об отпусках, больничных и других видах отсутствия. На основе интегрированной информации руководство получает возможность выявлять и контролировать возникающие в производственном цикле отклонения от принятых схем. Система дает пользователю доступ к профессиональному и веб интерфейсу. Быстрый ввод больших пакетов информации, проверка и передача информации в Oracle Зарплата, обеспечивает эффективный учет времени и своевременную обработку интегрированной информации. Таким образом, Oracle Табельный учет раскрывает перед руководством предприятия реальную исчерпывающую картину использования трудовых ресурсов и путей их оптимизации.

4. Oracle Payroll

В свою очередь интегрированная система Oracle Зарплата (Oracle Payroll) задействована в системе управления информацией по заработной плате [2]. Она поддерживает все виды начислений и удержаний и позволяет вести расчет и выплату любых видов оплаты, которые существуют в компании и утвержденных законодательством. Система позволяет выполнять следующие операции:

- управлять учетом затрат на персонал;
- вести планирование и расчет ФОТ;
- получать данные об отработанном времени из Oracle Табельный учет;

- передавать информацию в системы ведения бухгалтерского учета (например, Oracle Главная Книга);
- готовить статистическую и налоговую и отчетность согласно с утвержденными формами;
- поддерживать полную целостность и защищенность данных (включая архивные данные);
- предоставлять ряду уполномоченных сотрудников доступ к информации для составления отчетов, справок, и ответа на запросы [3].

5. Oracle Workforce Scheduling

Набор инструментов Oracle Планирование работы персонала (Oracle Workforce Scheduling) дает руководству компании возможности прогнозирования потребности в персонале и расстановке нужных сотрудников с соответствующими компетенциями на нужные позиции в соответствии с потребностями рынка. Инструмент формирует оптимальные графики работы, учитывая требования по обслуживанию клиентов и индивидуальные пожелания, и цели сотрудников. Эта система задействована в контроле расходов на оплату труда и рабочего времени. Интегрированные механизмы математической оптимизации создают 24/7 графики работы, учитывая изменения спроса, индивидуальные предпочтения и компетенции сотрудников, законодательные и бюджетные ограничения по рабочему времени, лучшие практики и внутренние правила. Подобное решение оптимизирует деятельность предприятий розничной торговли, медицинских центров и сервисных служб. Поиск близких к оптимальным решений ведет к учету самых разных правил и критериев, таких как: рабочие часы, компетенции работников, профессиональный уровень, допустимое время между переходами, распределение между сменами и т.д. Возможен учет нерегулярных событий – тренинг сотрудников и внеплановая встреча персонала.

6. Oracle Advanced Benefits

Система Oracle Компенсационный пакет (Oracle Advanced Benefits) предназначено для управления сложными компенсационными процессами, способствующими созданию благоприятных условий для привлечения и удержания наиболее талантливых и перспективных сотрудников. Рассматриваемая система служит инструментом мотивации персонала, учитывая интересы всей компании и отдельных сотрудников. Компенсационные схемы настраиваются в зависимости от особенностей организации – ее сферы деятельности, целей, бюджета и кадрового состава. Информирование о наличии льгот и их начисление, формирование персональных компенсационных предложений, распространение льгот на членов семьи усиливают мотивационную составляющую компенсационной

политики компании. Система может моделировать вариант наступления вероятного события, влияющего на общую сумму компенсации сотрудника, определять правила расчета элементов компенсаций, правила зачисления и обновления компенсаций, а также порядок выплат страховых премий. Обязательное оповещение потенциальных участников компенсаций обо всех имеющихся, существующих и планируемых схемах делает систему чрезвычайно удобной для всех сотрудников. Система поддерживает все виды зачисления на компенсации: автоматически, по требованию, на основе жизненного события и в запланированном режиме.

7. Oracle Incentive Compensation и Compensation Workbench

В тоже время система Oracle Комиссионные выплаты (Oracle Incentive Compensation) задействована в осуществлении строгого финансового контроля расходов. С ее помощью руководство компании создает прозрачную систему вознаграждения сотрудников, которая увязана с результатами их деятельности и с бизнес-целями компании. Интернет-архитектура, позволяющая управлять комиссионными выплатами в глобальном масштабе и интегрируемые финансовые процессы делают систему выплат гибкой и устойчивой по отношению к меняющейся компенсационной политике.

Следующий инструмент мотивации сотрудников – Платим за результат (Compensation Workbench) позволяет специалистам по расчету компенсаций и линейным руководителям оптимизировать процесс управления компенсациями в масштабах всего предприятия, путем увязывания всех видов компенсаций с реальными достижениями сотрудников.

В процессе управления человеческими ресурсами подбор персонала играет огромное значение. Инструмент интернет подбора персонала Oracle iRecruitment позволяет находить высококвалифицированных сотрудников и линейных руководителей, наиболее соответствующих потребностям компании и рынка. Автоматизация всего цикла операций по подбору наиболее подходящих соискателей и выявление степени их соответствия планируемой должности позволяет руководству сформировать коллектив при наименьшем расходовании финансовых средств и ресурсов времени. Система позволяет руководителям подразделений и рекрутерам создавать, копировать и обновлять описания вакансий и должностных инструкций для этих вакансий, размещая эти описания сайтах организации [4]. Благодаря инструментам Oracle iRecruitment, система автоматически определяет соответствие кандидатов объявленной вакансии и сравнивает их друг с другом по ряду признаков. В рамках системы рекрутеры имеют возможность управлять процессом интервьюирования кандидата, обновлять сведения по кандидату, осуществлять необходимые проверки,

отбирать и принимать на работу выбранных кандидатов. С помощью функции Рекомендаций менеджер по подбору персонала может порекомендовать кандидата другому линейному менеджеру.

Инвестиции в человеческий капитал очень важны, так как они повышают конкурентоспособность компании. Система управления персоналом Oracle HCM предоставляет пользователям возможности использования приложения Oracle Управление обучением для формирования кратко- и долгосрочных планов развития сотрудников, которая поддерживает все формы подготовки и переподготовки кадров. Применение комбинированного обучения, системы стандартного аудиторного и онлайн обучения, а также интерактивное взаимодействие участников процесса на базе форумов, чатов и т.д. активно способствует усвоению новых знаний и навыков. Приобщиться к процессу обучения могут не только сотрудники предприятия, но и ее партнеры и заказчики. Система обладает каталогом курсов и контрольным перечнем ресурсов. Администратор системы настраивает возможность многократного использования контента и оперативно обновляет его в соответствии с потребностями предприятия, предлагая для обучения учебный материал в стандартах IMS, AICC и SCORM. Каждый курс представляет вниманию пользователя набор компетенций, которыми он будет владеть после успешного завершения обучения. После получения заданной оценки по результатам тестирования, происходит процесс автоматического обновления профиля.

8. Oracle Self-service Human Resources

Oracle Управление персоналом (Oracle Self-service Human Resources) позволяет организации использовать свои трудовые ресурсы наиболее оптимальным образом. Она обладает Профилем Талантов, который представляет собой единое пространство для проведения кадровых процедур и получения данных о сотрудниках (компетенциях, плане карьерного роста, сертификатах, результатах аттестации, стаже работы, контактной информации и т. д.) [5]. Исчерпывающая информация о сотруднике является основой для принятия обоснованного решения. Для формирования цепочки согласования кадровых действий использует Oracle Workflow либо применяется интеграция с Oracle Approvals Management. Доступ к информации контролируется администратором системы, которая может в любое время присвоить или ограничить права пользователей.

Инструментом для постановки целей и контроля их исполнения служит Oracle Управление деятельностью (Oracle Performance Management). Он задействован в оценке сотрудников, в определении степени соответствия целей персонала с целями компании и в выявлении кандидатов для кадрового резерва. Работая со специализированными

интуитивным интерфейсами, линейные руководители предлагают способы своевременного просмотра и обновления информации о сотрудниках. Система поддерживает возможность создания индивидуальных планов и согласования целей персонала с целями организации.

Заключение

Среди основных возможностей автоматизированного решения Oracle Управление деятельностью необходимо выделить следующие:

- поддержка всех видов оценки и аттестации;
- настройка единой библиотеки целей;
- автоматическое распределение целей по заранее настроенным правилам;
- каскадирование целей путем дублирования целей руководителя;
- возможность формирования дополнительных целей сотрудников с изменением требований организации;
- контроль достижения целей;
- интеграция с системой Управления обучением Oracle;
- использование потока операций для поддержки следования документов на одобрение.

Результатом работы приложения является формирование консолидированной картины эффективности работы персонала. Планирование карьерного роста сотрудников и формирование кадрового резерва осуществляется на базе интегрированного решения Oracle Планирование Замещения (Oracle Succession Planning). Система открывает доступ к данным о позициях и ключевых сотрудниках и об их перемещении в компании для получения наилучшего результата. На сегодняшний день систему управления человеческими ресурсами Oracle HCM используют такие организации, как Ford, ING Bank Netherlands, медиа-холдинг Hearst, Макдоналдс, производитель кухонной техники Polaris, Toshiba Medical Systems и многие другие.

Таким образом, система кадрового менеджмента Oracle HCM открывает перед руководителем широкие возможности. Во-первых, она способствует оптимизации процесса принятия решений посредством предоставления пользователям необходимой информации. Во-вторых, наличие единой, унифицированной и эффективной информационной архитектуры обеспечивает низкую стоимость владения системой и окупаемость инвестиций, что позволяет сократить период выпуска продуктов на рынок и быстро приступить к работе с новыми источниками дохода. В-третьих, анализ данных, получаемых из различных HCM-процессов, дает возможность устранить «узкие места» менеджмента в режиме текущего времени. В-четвертых, встроенная во все компоненты

HR-системы предиктивная аналитика позволяет формировать планы и делать краткосрочные прогнозы развития внутренних процессов.

Список литературы

1. Oracle Human Capital Management Cloud [Электронный ресурс] : Oracle information portal. – Режим доступа : <http://www.oracle.com/ru/applications/human-capital-management/products/index.html>
2. Wolfe, A. HR Transformation Elevating Human Capital Management To C-Suite [Электронный ресурс] : Forbes information portal / A. Wolfe. – Режим доступа: <http://www.forbes.com/sites/oracle/2013/09/24/hr-transformation-elevating-human-capital-management-to-c-suite/#f09a49464a80>
3. Kanaracus, Ch. Oracle Goes After Workday, SuccessFactors With Fusion HCM [Электронный ресурс] : PCWorld information portal / Ch. Kanaracus. – Режим доступа : <http://www.pcworld.com/article/225753/article.html>
4. Горчинская, О. Решения Oracle в области бизнес-анализа [Электронный ресурс] : Аналитика завтрашнего дня / О. Горчинская. – Режим доступа : http://www.itoil.ru/common/oil_img/oracle
5. Дубинин, П. Обзор облака Oracle и возможностей Database Cloud Service [Электронный ресурс] : Аналитика завтрашнего дня / П. Дубин. – Режим доступа: http://www.pm-partner.ru/upload/medialibrary/d74/PD_cloud_concept1_ekb

МОДЕЛИРОВАНИЕ И АНАЛИЗ БАЛАНСИРОВКИ НАГРУЗКИ В ОБЛАЧНЫХ ВЫЧИСЛЕНИЯХ

А.А. Ходар,
Аспирант
У.Ж. Алькади,
Аспирант
Е.Н. Десятирикова
Профессор

Введение

Cloud computing («облачные вычисления») – концепция «вычислительного облака», согласно которой программы запускаются и выдают результаты работы в окно стандартного веб-браузера на локальном

ПК, при этом все приложения и их данные, необходимые для работы, находятся на удаленном сервере в Интернете.

Концепция «облачных вычислений» зародилась в 1960 г., когда Джон Маккарти высказал предположение, что когда-нибудь компьютерные вычисления будут производиться с помощью общенародных утилит.

Термин *cloud computing* стал широко употребляться в США с 2005 года после запуска компанией Amazon.com проекта Elastic Compute Cloud (Amazon EC2) и широко распространился в бизнесе, среди поставщиков информационных технологий и в научно-исследовательской среде.

К преимуществам облачных вычислений можно отнести следующие: снижаются требования к вычислительной мощности ПК, возрастают отказоустойчивость и безопасность, многократно увеличивается скорость обработки данных, снижаются затраты на аппаратное и программное обеспечение, на обслуживание, электроэнергию, экономится дисковое пространство.

Облачные вычисления – это технологии, которые помогают обмениваться данными и предоставлять много ресурсов пользователям. Пользователи платят только за те ресурсы, которые они использовали. Облачные вычисления хранят данные и распределенные ресурсы в открытой среде, и объем хранения данных очень быстро увеличивается. Таким образом, балансировка нагрузки является основной задачей в облачной среде. Балансировка нагрузки помогает распределить динамическую рабочую нагрузку между несколькими узлами, чтобы гарантировать, что ни один узел не будет перегружен. Данное исследование в основном сосредоточено на анализе производительности облачных вычислений и сравнении различных алгоритмов балансировки нагрузки с использованием сетевого симулятора Cloud Analyst.

1. Балансировка нагрузки

Балансировка нагрузки используется для распределения большой нагрузки на локальные узлы обработки для повышения общей производительности системы. В облачной вычислительной среде необходимо обеспечивать динамическую локальную рабочую нагрузку равномерно между всеми узлами [1].

Балансировка нагрузки помогает достичь высокого уровня удовлетворенности пользователей и надлежащего использования ресурсов. Балансировка нагрузки, таким образом, имеет целью обеспечить максимальную пропускную способность сети с минимальным временем отклика системы.

Балансировка нагрузки в облачных вычислениях выполняется на двух уровнях [2]:

– *Уровень виртуальной машины*, при этом балансировка проводится между приложениями, которые загружаются в облаке на виртуальную машину.

– *Уровень хоста*, когда распределяется нагрузка между виртуальной машиной и хост-ресурсами, что позволяет обрабатывать несколько поступающих запросов приложения.

2. Существующие политики балансировки нагрузки

Существуют различные алгоритмы балансировки нагрузки, используемые в облачных вычислениях. В данном исследовании изучены нижеследующие три алгоритма, которые могут быть реализованы в симуляторе Cloud Analyst [3].

2.1 Round Robin Algorithm (RR)

Это самый простой алгоритм, который использует понятие временного интервала. Здесь время делится на несколько секторов, и каждому узлу задается определенный квант времени, и в этом кванте узел будет выполнять свои операции.

Недостатком метода является то, что, хотя алгоритм очень простой, но для определения размера кванта он генерирует дополнительную нагрузку на планировщик. Кроме того, он имеет более высокие переключатели контекста, которые увеличивают время оборота, и низкую пропускную способность [3].

2.2. Активный Мониторинг Балансировки нагрузки (AMLB)

Этот алгоритм имеет динамический характер. Он хранит информацию о каждой виртуальной машине (VM) и количестве запросов, которые в настоящее время назначены для каждой VM. AMLB всегда находит наименее загруженную VM для назначения нового входящего запроса.

Основным недостатком метода являются дополнительные вычислительные расходы для сканирования очереди снова и снова [3].

2.3. Throttled Load Balancing Algorithm (TLB)

В этом алгоритме балансировщик нагрузки поддерживает таблицу индексов виртуальных машин, а также их состояния (Доступно или Занят). Балансировщик нагрузки сканирует индексную таблицу сверху вниз, чтобы найти подходящую виртуальную машину для выполнения рекомендуемого задания. При обработке запроса клиента, если соответствующая VM не найдена, балансировщик нагрузки возвращает -1 и инициирует запрос в ожидании очереди.

Недостаток метода в том, что нагрузка на первые машины больше, чем на остальные в том случае, когда количество запросов невелико [3].

3. Cloud Analyst Симулятор

Моделирование и анализ производительности трех указанных алгоритмов балансировки нагрузки проведены в работе с помощью инструмента «Cloud Analyst». Он позволяет пользователю выполнять многократные симуляции с небольшими изменениями параметров, а также позволяет настраивать расположение пользователей, которые создают приложение, и расположение центров обработки данных. Укажем терминологию эмулятора [4]:

- *регион*: в Cloud Analyst мир поделен на 6 регионов, которые совпадают с 6 основными континентами в мире;
- *база пользователей* : User Base рассматривается как единое целое, и используется для генерации трафика;
- *центр обработки данных*: брокерское обслуживание определяет, какой центр должен принять и обработать запрос, который приходит от каждой пользовательской базы;
- *VmLoadBalancer*: он отвечает за распределение нагрузки на доступный центр данных. VmLoadBalancer распределяет нагрузку в центре обработки данных на основе политики балансировки нагрузки.

4. Моделирование и эксперимент

Моделирование и виртуальный эксперимент являются лучшим способом проверки алгоритма в облачных вычислениях. Рассмотрим работу каждого из трех алгоритмов балансировки нагрузки на примере социальной сети Facebook, имеющей более 200 миллионов зарегистрированных пользователей по всему миру (табл. 1).

Таблица 1

Зарегистрированные пользователи Facebook по всему миру

Регион	ID регион	Количество пользователей
Северная Америка	0	80 миллионов
Южная Америка	1	20 миллионов
Европа	2	60 миллионов
Азия	3	27 миллионов
Африка	4	5 миллионов
Океания	5	8 миллионов

Для нашего моделирования предположим, что мы имеем аналогичную систему, но в масштабе 1/10 (табл. 2). Определим 6 пользовательских баз, представляющих указанные выше 6 регионов, со следующими параметрами.

Таблица 2

Параметры пользовательских баз

база	регион	часовой пояс	Часы пик (Местное время)	Часы пик (GMT)	пользователи онлайн в пиковых	пользователи онлайн в непикувых
UB1	0	GMT - 6.00	7.00–9.00 pm	13:00-15:00	400,000	40,000
UB2	1	GMT - 4.00	7.00–9.00 pm	15:00-17:00	100,000	10,000
UB3	2	GMT + 1.00	7.00–9.00 pm	20:00-22:00	300,000	30,000
UB4	3	GMT +6.00	7.00–9.00 pm	01:00-03:00	150,000	15,000
UB5	4	GMT + 2.00	7.00–9.00 pm	21:00-23:00	50,000	5,000
UB6	5	GMT +10.00	7.00–9.00 pm	09:00-11:00	80,000	8,000

Также определим центр обработки данных, который должен обрабатывать запрос, приходящий от каждой пользовательской базы со следующими параметрами (табл. 3).

Таблица 3

Параметры центра обработки данных

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/GB	Physical Hw Units
Dc1	0	X86	Linux	Xen	0.1	0.05	0.1	0.1	20

Ограничим модель тем, что каждая пользовательская база содержится в одном часовом поясе, и предположим, что большинство пользователей используют приложение по вечерам после работы около 2 часов.

Положим также, что 5% из зарегистрированных пользователей будут on-line в пиковое время одновременно и только одна десятая из этого числа будут в сети в течение непикувых часов. Положим, что каждый пользователь делает новый запрос каждые 5 минут, когда находится on-line.

5. Результаты моделирования

Нами три раза проведено моделирование в соответствии с предыдущими параметрами. Каждый раз, мы меняли алгоритм

балансировки нагрузки, который и анализировался. Сравнение результатов проводилось по критериям: общее время отклика (рис. 1) , время центра обработки данных (рис. 2) и стоимость обработки (табл. 4).

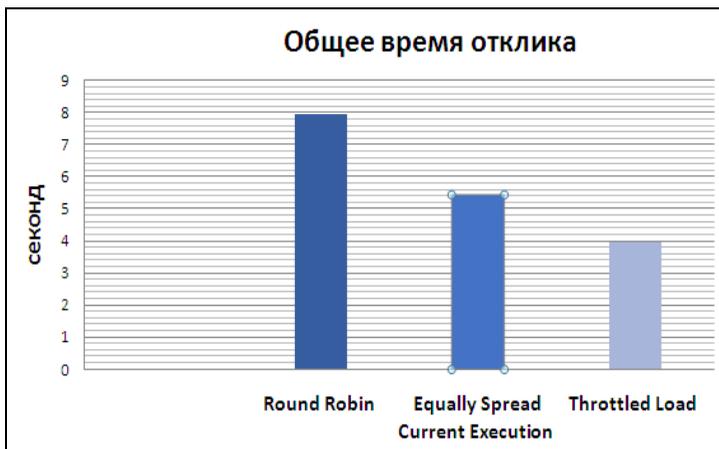


Рис. 1. Общее время отклика

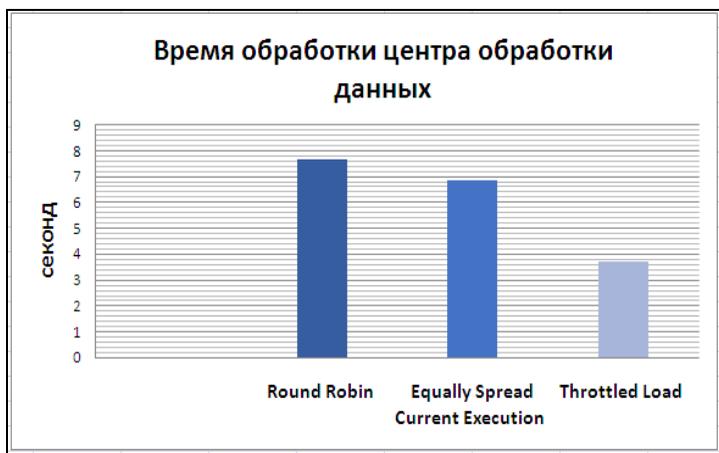


Рис. 2. Время обработки центра данных

Стоимость обработки

Алгоритм	VM Cost \$	Total \$
RR	120.5	632.34
AMLB	112.8	580.67
TLB	99.1	501.92

Заключение

Сравнивая результаты, полученные с использованием различных алгоритмов балансировки нагрузки, можно сделать вывод, что общее время отклика в алгоритме throttled лучше, чем в других алгоритмах, также лучше и время центра обработки данных.

Эта работа может быть продолжена с использованием различных алгоритмов оптимизации «роя частиц». Кроме того, мы планируем расширить исследование методов балансировки нагрузки, используя различные политики брокерских услуг.

Список литературы

1. Buyya, R. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities / R. Buyya, R. Ranjan, R.N. Calheiros // 2009 Int. Conf. High Perform. Comput. Simul. – USA: IEEE, 2009. – P. 1-11.
2. Calheiros, R.N. CloudSim : A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services / R.N. Calheiros. Eprint : Australia, 2009. – 9 p.
3. Maguluri, S.T. Stochastic models of load balancing and scheduling in cloud computing clusters / S.T. Maguluri, R. Srikant, L. Ying // 2012 Proc. IEEE INFOCOM. – 2012. – P. 702-710.
4. Корсуков, А.С. Инструментальные средства полунатурного моделирования распределенных вычислительных систем / А.С. Корсуков // Современные технологии. Системный анализ. Моделирование. – Иркутск : Издательство Иркутского государственного университета путей сообщения. – 2011. – Т. 3. – С. 105-110.

СОДЕРЖАНИЕ

1. Акимова Е.В., Митрофанова Е.Ю. СРАВНЕНИЕ МЕТОДОВ СУПЕРПИКСЕЛЬНОЙ СЕГМЕНТАЦИИ	3
2. Аксенов Р.П., Коваль А.С. СИСТЕМА УЧЕТА ПОСЕЩАЕМОСТИ НА ОСНОВЕ RFID	10
3. Антипов К.Г., Другалев А.И. РЕАЛИЗАЦИЯ МОДУЛЯ РАСЧЕТА СТИПЕНДИЙ ДЛЯ УЧЕБНЫХ ЗАВЕДЕНИЙ НА ПЛАТФОРМЕ 1С.....	16
4. Баранников И.С., Другалев А.И. МИКРОСЕРВИСНАЯ АРХИТЕКТУРА СО SPRING CLOUD.....	23
5. Безруков М.А., Другалев А.И. ГЕОПОЗИЦИОНИРОВАНИЕ СОТРУДНИКОВ ПОСРЕДСТВОМ МОБИЛЬНОГО ПРИЛОЖЕНИЯ 1С НА ПЛАТФОРМЕ ANDROID.....	27
6. Белевский Г.А., Борисов Д.Н. АНАЛИЗ МОДЕЛЕЙ КАНАЛОВ МНОГОЛУЧЕВОГО РАСПРОСТРАНЕНИЯ СИГНАЛА.....	33
7. Бобков Р.И., Зуев С.А. РАЗРАБОТКА ИНТЕРФЕЙСА ДЛЯ СЕМЕЙСТВА ОС LINUX АДАПТИРОВАННОГО ДЛЯ ЛЮДЕЙ ПОЖИЛОГО ВОЗРАСТА	37
8. Богомолов Д.Ю., Лысачев П.С. РЕАЛИЗАЦИЯ АУДИО- И ВИДЕОКОНФЕРЕНЦИЙ НА МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ	42
9. Васильев А.В., Дрюченко М.А. DLP СИСТЕМА НА ОСНОВЕ СТАТИСТИЧЕСКИХ И СТЕГАНОГРАФИЧЕСКИХ МЕТОДОВ	4
10. Гаршин Т.С., Соломатин Д.И. ПОСТРОЕНИЕ И АНАЛИЗ СОЦИАЛЬНЫХ ГРАФОВ ПОЛЬЗОВАТЕЛЕЙ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»	53
11. Гетманова К.Г., Павлов В.Е., Лобода А.В. РЕШЕНИЕ СИСТЕМ ПОЛИНОМИАЛЬНЫХ УРАВНЕНИЙ В ПАКЕТЕ MAPLE	59
12. Гончарова Д.А., Матвеев М.Г. МОДЕЛИРОВАНИЕ РАБОТЫ ФУНКЦИОНАЛЬНЫХ БЛОКОВ СОЦИАЛЬНОЙ СЕТИ С ЦЕЛЬЮ ИЗУЧЕНИЯ ИХ ЭФФЕКТИВНОСТИ	65
13. Горовой А.Б., Фертиков В.В. ИСПОЛЬЗОВАНИЕ КЛАСТЕРНОГО АНАЛИЗА В РАСПОЗНАВАНИИ НОТНЫХ ЛИСТОВ	7
14. Данилова М.В., Сычев А.В. АНАЛИЗ ВЛИЯНИЯ В СОЦИАЛЬНОЙ СЕТИ НА ОСНОВЕ ВЫДЕЛЕНИЯ СЕТЕВЫХ СООБЩЕСТВ.....	77

15. Дедов Д.С., Матвеев М.Г. ОПРЕДЕЛЕНИЕ РЕГЛАМЕНТНЫХ ПАРАМЕТРОВ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ НА ОСНОВЕ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ.....	82
16. Денисенко А.И., Лысачев П.С. ПЕРЕДАЧА ДАННЫХ МЕЖДУ АЛГОРИТМАМИ В ЭКСПЕРИМЕНТАХ С ВЛОЖЕННЫМИ ЦИКЛАМИ	90
17. Денищенко А.А., Нехаев А.Р. РАЗРАБОТКА ВЕБ ПРИЛОЖЕНИЯ ВИДЕОЧАТ	96
18. Железной А.С., Зуев А.С. УНИВЕРСАЛЬНЫЙ АВТОМАТИЧЕСКИЙ ТЕСТЕР РАДИОДЕТАЛЕЙ.....	103
19. Желтоухов М.О., Другалев А.И. ФРЕЙМВОРК ДЛЯ РАЗРАБОТКИ ЧАТБОТОВ НА ЯЗЫКЕ JAVA	107
20. Землянухин М.В., Соломатин Д.И. ОПТИМИЗАЦИЯ АЛГОРИТМА ПОИСКА БЛИЖАЙШИХ ТОЧЕК ДЛЯ ПРИМЕНЕНИЯ В ИТЕРАТИВНОМ АЛГОРИТМЕ БЛИЖАЙШИХ ТОЧЕК (ICP).....	112
21. Иванников Ю.Ю., Митрофанова Е.Ю. ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ БОТ-ТРАФИКА СРЕДИ ЗАПРОСОВ К ВЕБ-ПРИЛОЖЕНИЮ	119
22. Ильин М.С., Тарасов В.С. АНАЛИЗ СРЕДСТВ ПО УСКОРЕНИЮ ВЫЧИСЛЕНИЙ ДЛЯ УСТРОЙСТВ НА ПЛАТФОРМЕ ANDROID	123
23. Ильина А.А., Тарасов В.С. ТОНКАЯ НАСТРОЙКА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ	130
24. Исакова Д.В., Лысачев П.С. РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ СИСТЕМЫ ДЛЯ МОНИТОРИНГА ОКРУЖАЮЩЕЙ СРЕДЫ ВНУТРИ ПОМЕЩЕНИЯ.....	136
25. Казун Г.И., Сирота А.А. МОДИФИКАЦИЯ МЕТОДА ВИОЛЫ-ДЖОНСА НА ОСНОВЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ЦВЕТА КОЖИ	141
26. Калабухов К.С., Гаршина В.В. АНАЛИЗ ТОНАЛЬНОСТИ ТЕКСТА	149
27. Кашин Е.Д., Соломатин Д.И. КЛАССИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ ПО ТИПУ ПОЛУЧЕННОГО ВЫСШЕГО ОБРАЗОВАНИЯ НА ОСНОВЕ СООБЩЕНИЙ В СОЦИАЛЬНОЙ СЕТИ.....	154
28. Козориз Д.В., Лобода А.В. КОНФОРМНЫЕ ОТОБРАЖЕНИЯ ПЛОСКИХ ФИГУР	161

29. Козырев А.В., Самойлов Н.К. ОПТИМИЗАЦИЯ РАСПИСАНИЯ ДВИЖЕНИЯ ГОРОДСКОГО АВТОТРАНСПОРТА. АВТОМАТИЗАЦИЯ ИЗМЕНЕНИЯ И СОСТАВЛЕНИЯ МАРШРУТОВ	166
30. Колесов А.С., Соломатин Д.И. ПРОЕКТИРОВАНИЕ И СБОРКА 3D-ПРИНТЕРА, АВТОМАТИЗАЦИИ ЭТАПОВ 3D-ПЕЧАТИ	172
31. Копытина Е.А., Копытин А.В., Соломатин Д.И. ТЕОРЕТИКО-МНОЖЕСТВЕННЫЙ ПОДХОД К РАСПРЕДЕЛЕНИЮ ШТРАФОВ И ВОЗНАГРАЖДЕНИЙ В ПРОЕКТНОЙ ДЕЯТЕЛЬНОСТИ	177
32. Коротков В.В., Вахтин А.А. МОДЕЛИРОВАНИЕ ЭМОЦИЙ ВИРТУАЛЬНЫХ ПЕРСОНАЖЕЙ	186
33. Кулаков В.И., Борисов Д.Н. РАЗРАБОТКА УСТРОЙСТВА ДЛЯ МОНИТОРИНГА СОСТОЯНИЯ ОКРУЖАЮЩЕЙ СРЕДЫ ВНУТРИ ПОМЕЩЕНИЯ	191
34. Ласкин А.Г., Дрюченко М.А. АЛГОРИТМ ЗАЩИТЫ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ СТЕГАНОГРАФИЧЕСКОГО «САМОВЛОЖЕНИЯ» КОНТЕЙНЕРА.....	197
35. Левчук П.О., Зуев С.А. МАЛОГАБАРИТНЫЙ БРАСЛЕТ ДЛЯ СНЯТИЯ ЭКГ	204
36. Лоскутова А.В., Абрамов И.В. ОРГАНИЗАЦИЯ ПРОЕКТНОГО ОФИСА В КОМПАНИИ «WIZART ANIMATION»	207
37. Мелихов В.А., Толстобров А.П. ПРОЕКТИРОВАНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ВЫБОРА УЧЕБНЫХ ЭЛЕМЕНТОВ LMS MOODLE	211
38. Морозов Е.Ю., Лобода А.В. ИССЛЕДОВАНИЕ СИММЕТРИЙ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ	215
39. Мубаракшин Д.Э., Сирота А.А. РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ КЛАСТЕРНОГО АНАЛИЗА ДАННЫХ ДЛЯ ОБЪЕКТИВНОЙ ОЦЕНКИ ЧИСЛА ПОРОЖДАЮЩИХ КЛАССОВ.....	221
40. Назарьева В.А., Матвеев М.Г. МЕТОДИКА КОМПЬЮТЕРНОГО АНАЛИЗА ПРОИЗВОДСТВЕННЫХ БИЗНЕС-ПРОЦЕССОВ НА ПРИМЕРЕ СКЛАДА ПРОИЗВОДСТВЕННЫХ РЕСУРСОВ	227
41. Напрасникова М.А., Сычев А.В. СРАВНЕНИЕ МЕТОДОВ ОБНАРУЖЕНИЯ СКРЫТЫХ ТЕМ В ПОТОКЕ ТВИТОВ ИЗ СОЦИАЛЬНОЙ СЕТИ TWITTER	233

42. Нижневский В.В., Другалев А.И. МЕХАНИЗМ ОБМЕНА ДАННЫМИ МЕЖДУ ТИПОВЫМИ КОНФИГУРАЦИЯМИ 1С И МОБИЛЬНЫМ ПРИЛОЖЕНИЕМ 1С ПОД ANDROID.....	237
43. Новиков В.А., Соломатин Д.И. РАЗРАБОТКА УЧЕБНОГО КОМПИЛЯТОРА ДЛЯ ПРОЦЕДУРНОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ	242
44. Ногих А.А., Соломатин Д.И. АВТОМАТИЗАЦИЯ ПРОЦЕССА ОБНАРУЖЕНИЯ ВАНДАЛИЗМА В РУССКОЯЗЫЧНОМ РАЗДЕЛЕ ВИКИПЕДИИ.....	249
45. Обручников А.В., Соломатин Д.И. ПРИМЕНЕНИЕ ТЕХНОЛОГИИ БЛОКЧЕЙН НА ПРИМЕРЕ ПРОТОТИПА ПРИЛОЖЕНИЯ ДЛЯ ПРОВЕДЕНИЯ ЛОТЕРЕЙ	256
46. Паневина А.С., Абрамов И.В. РЕКОМЕНДАЦИИ ПРИ ВЫБОРЕ СИСТЕМЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА.....	263
47. Плотникова А.Г., Матвеев М.Г. СТАТИСТИЧЕСКИЙ АНАЛИЗ ПАРАМЕТРОВ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА.....	269
48. Полещук Х.А., Тюкачев Н.А. СВЯЗЬ ПЛАТЫ STK600 С ПК.....	275
49. Разуваев Д.В., Другалев А.И. КЛИЕНТ-СЕРВЕРНОЕ ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ЛИЧНЫМ ВРЕМЕНЕМ	282
50. Резниченко Е.В., Матвеев М.Г. РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ ПРЕДСКАЗАНИЯ ОТТОКА КЛИЕНТОВ С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ	287
51. Савченко И.Ю., Матвеев М.Г. ЭКСПЕРТНАЯ СИСТЕМА «АГРОКОНСУЛЬТАНТ» НА ОСНОВЕ АППАРАТА НЕЧЕТКОЙ ЛОГИКИ	292
52. Семенова А.Н., Абрамов И.В. СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ МЕТОДА АНАЛИЗА ИЕРАРХИЙ ДЛЯ ОПТИМИЗАЦИИ ВЫБОРА ПАРТНЕРОВ КОМПАНИИ	296
53. Соколова В.А., Гавшин А.Л. ОБРАБОТКА СИГНАЛОВ VLE-УСТРОЙСТВ НА БАЗЕ ПРОФИЛЯ FIND ME В ANDROID-ПРИЛОЖЕНИЯХ.....	303
54. Струков Р.И., Тюкачев Н.А. ПРИМЕНЕНИЕ SPI-ИНТЕРФЕЙСА ДЛЯ УПРАВЛЕНИЯ ЭКВАЛАЙЗЕРОМ ВХОДНОГО ВИДЕОСИГНАЛА.....	310
55. Танвель М.Ю., Вялых С.А. ОЦЕНКА ВОЗМОЖНОСТИ РЕАЛИЗАЦИИ УГРОЗ БЕЗОПАСНОСТИ ПЕРСОНАЛЬНЫХ ДАННЫХ.....	315

56. Тюленев С.И., Семенихин Н.А., Нечаев Ю.Б. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ОДНОГО КАНАЛА ПРИЕМА АНТЕННОЙ РЕШЕТКИ QPSK-СИГНАЛА	319
57. Хальзов С. А., Фертиков В. В. АЛГОРИТМ ПОСТРОЕНИЯ ДИСКРЕТНОЙ ДИАГРАММЫ ВОРОНОГО К-ГО ПОРЯДКА ПРОИЗВОЛЬНОЙ РАЗМЕРНОСТИ С ЗАДАННОЙ ТОЧНОСТЬЮ	323
58. Черепанова С.В., Тюкачев Н.А. СИСТЕМА АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ДОМАШНИМ ОБОРУДОВАНИЕМ С ОБРАТНОЙ СВЯЗЬЮ «УМНЫЙ» ДОМ	330
59. Швырева А.В., Лобода А.В. ИНТЕГРИРОВАНИЕ МАТРИЧНЫХ АЛГЕБР ЛИ.....	336
60. Щеглов А.В., Обущенко Е.Ю., Храмов В.Ю. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИНТЕЗА ТЕСТОВЫХ СПОСОБОВ ВОЗДЕЙСТВИЯ ДЛЯ СИСТЕМЫ ЗАЩИТЫ СЕТЕЙ РАДИОСВЯЗИ ОТ КОМПЬЮТЕРНЫХ АТАК.....	343
61. Азиз А.И., Борзунов С.В. ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА НУМЕРОВА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ CUDA	350
62. Алубади Я.Х.А., Абрамов И.В. ОСОБЕННОСТИ ФОРМИРОВАНИЯ УЧЕБНЫХ ГРУПП НА КУРСАХ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ МИНИСТЕРСТВА ОБРАЗОВАНИЯ РЕСПУБЛИКИ ИРАК.....	355
63. Хазим Аль-Афаре, Туровский Я.А. КОРРЕЛЯЦИОННЫЕ ЗАВИСИМОСТИ МЕЖДУ АЛЬТЕРАЦИЕЙ ЗУБЦА R ЭКГ и RR-ИНТЕРВАЛАМИ	361
64. Аль-Судани Ф.А., Флегель А.В. ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ДЛЯ ОРТОГОНАЛЬНЫХ ПОЛИНОМОВ	364
65. Мохаммед И.А., Илларионов И.В. ПРОЦЕСС УПРАВЛЕНИЯ ТРУДОВЫМИ РЕСУРСАМИ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТОВ ORACLE.....	371
66. Ходар А.А., Алькади У.Ж., Десятирикова Е.Н. МОДЕЛИРОВАНИЕ И АНАЛИЗ БАЛАНСИРОВКИ НАГРУЗКИ В ОБЛАЧНЫХ ВЫЧИСЛЕНИЯХ	377

Научное издание

**СБОРНИК СТУДЕНЧЕСКИХ НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ НАУК ВГУ**

Под редакцией *Д. Н. Борисова*

Выпуск 11

Издано в авторской редакции

Подписано в печать 15.06.2017. Формат 60×84/16.
Усл. печ. л. 22,7. Тираж 500 экз. Заказ 383

Издательский дом ВГУ
394018 Воронеж, пл. Ленина, 10
Отпечатано с готового оригинал-макета в типографии
Издательского дома ВГУ.
394018 Воронеж, ул. Пушкинская, 3